

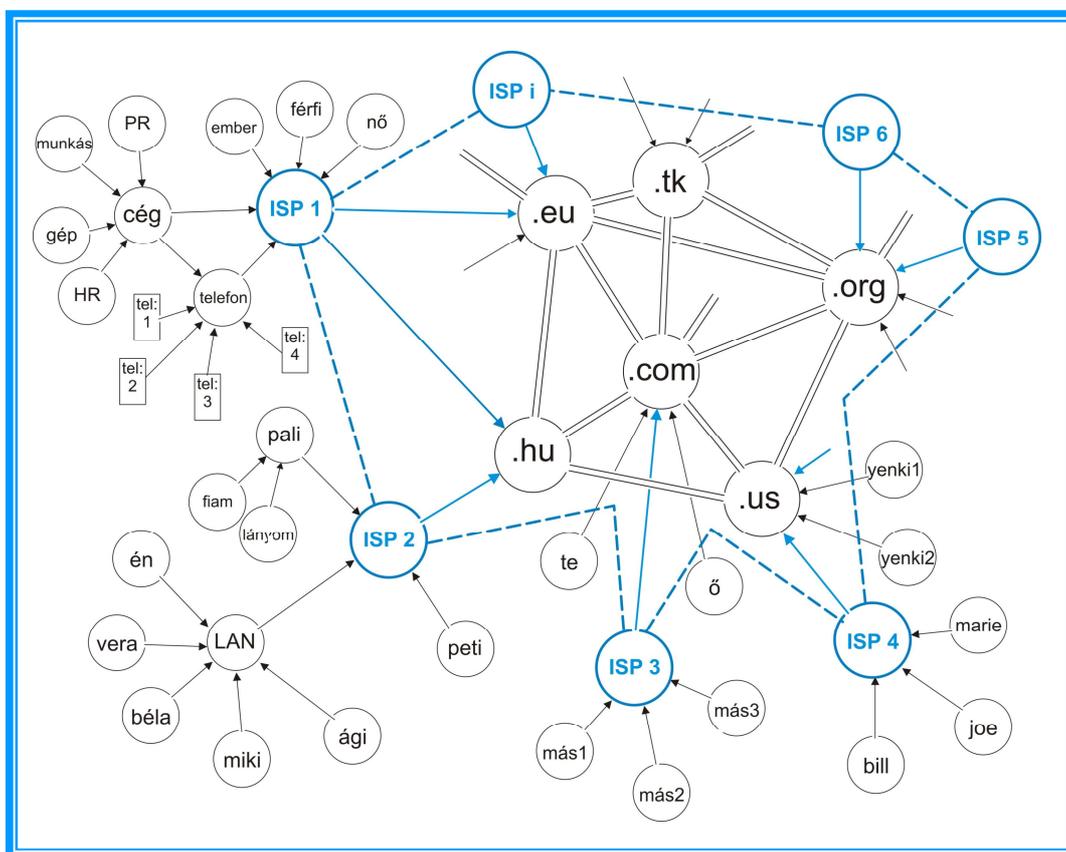
Some words about Networks.

I. part

Compiled by

Dr. Peter G. Gyarmati

for research purposes.



TCC COMPUTER STUDIO

2010.

Some words about Networks.

I. part

Compiled by Dr. Peter G. Gyarmati
for research purposes,
based on Wikipedia pages,
2010. 03. 6.

1.	Network science.....	5
2.	Network theory.....	6
3.	Graph theory	7
4.	Complex network.....	13
5.	Flow network	16
6.	Network diagram	18
7.	Network model (database).....	20
8.	Network analysis (electrical circuits)	21
9.	Social network.....	29
10.	Semantic network	37
11.	Radio and Television networks	39
12.	Business networking	40
13.	Dynamic network analysis.....	42
14.	Neural network	43
15.	Artificial neural network.....	49
16.	Perceptron	59

1. Network science

Network science is a new and emerging scientific discipline that examines the interconnections among diverse physical or engineered networks, information networks, biological networks, cognitive and semantic networks, and social networks. This field of science seeks to discover common principles, algorithms and tools that govern network behavior. The National Research Council defines Network Science as "the study of network representations of physical, biological, and social phenomena leading to predictive models of these phenomena."

The study of networks has emerged in diverse disciplines as a means of analyzing complex relational data. The earliest known paper in this field is the famous Seven Bridges of Königsberg written by Leonhard Euler in 1736. Euler's mathematical description of vertices and edges was the foundation of graph theory, a branch of mathematics that studies the properties of pairwise relations in a network structure. The field of graph theory continued to develop and found applications in chemistry (Sylvester, 1878).

In the 1930s Jacob Moreno, a psychologist in the Gestalt tradition, arrived in the United States. He developed the sociogram and presented it to the public in April 1933 at a convention of medical scholars. Moreno claimed that "before the advent of sociometry no one knew what the interpersonal structure of a group 'precisely' looked like (Moreno, 1953). The sociogram was a representation of the social structure of a group of elementary school students. The boys were friends of boys and the girls were friends of girls with the exception of one boy who said he liked a single girl. The feeling was not reciprocated. This network representation of social structure was found so intriguing that it was printed in the *The New York Times* (April 3, 1933, page 17). The sociogram has found many applications and has grown into the field of social network analysis.

Probabilistic theory in network science developed as an off-shoot of graph theory with Paul Erdős and Alfréd Rényi's eight famous papers on random graphs. For social networks the exponential random graph model or p^* graph is a notational framework used to represent the probability space of a tie occurring in a social network. An alternate approach to network probability structures is the network probability matrix, which models the probability of edges occurring in a network, based on the historic presence or absence of the edge in a sample of networks.

In the 1998, David Krackhardt and Kathleen Carley introduced the idea of a meta-network with the PCANS Model. They suggest that "all organizations are structured along these three domains, Individuals, Tasks, and Resources. Their paper introduced the concept that networks occur across multiple domains and that they are interrelated. This field has grown into another sub-discipline of network science called dynamic network analysis.

More recently other network science efforts have focused on mathematically describing different network topologies. Duncan Watts reconciled empirical data on networks with mathematical representation, describing the small-world network. Albert-László Barabási and Reka Albert developed the scale-free network which is a loosely defined network topology that contains hub vertices with many connections, that grow in a way to maintain a constant ratio in the number of the connections versus all other nodes. Although many networks, such as the internet, appear to maintain this aspect, other networks have long tailed distributions of nodes that only approximate scale free ratios.

Today, network science is an exciting and growing field. Scientists from many diverse fields are working together. Network science holds the promise of increasing collaboration across disciplines, by sharing data, algorithms, and software tools.

2. Network theory

Network theory is an area of computer science and network science and part of graph theory. It has application in many disciplines including particle physics, computer science, biology, economics, operations research, and sociology. Network theory concerns itself with the study of graphs as a representation of either symmetric relations or, more generally, of asymmetric relations between discrete objects. Applications of network theory include logistical networks, the World Wide Web, gene regulatory networks, metabolic networks, social networks, epistemological networks, etc. See list of network theory topics for more examples.

Network optimization

Network problems that involve finding an optimal way of doing something are studied under the name of combinatorial optimization. Examples include network flow, shortest path problem, transport problem, transshipment problem, location problem, matching problem, assignment problem, packing problem, routing problem, Critical Path Analysis and PERT (Program Evaluation & Review Technique).

Network analysis

Social network analysis

Social network analysis maps relationships between individuals in social networks.^[1] Such individuals are often persons, but may be groups (including cliques and cohesive blocks), organizations, nation states, web sites, or citations between scholarly publications (scientometrics).

Network analysis, and its close cousin traffic analysis, has significant use in intelligence. By monitoring the communication patterns between the network nodes, its structure can be established. This can be used for uncovering insurgent networks of both hierarchical and leaderless nature.

Biological network analysis

With the recent explosion of publicly available high throughput biological data, the analysis of molecular networks has gained significant interest. The type of analysis in this content are closely related to social network analysis, but often focusing on local patterns in the network. For example network motifs are small subgraphs that are over-represented in the network. Activity motifs are similar over-represented patterns in the attributes of nodes and edges in the network that are over represented given the network structure.

Link analysis

Link analysis is a subset of network analysis, exploring associations between objects. An example may be examining the addresses of suspects and victims, the telephone numbers they have dialed and financial transactions that they have partaken in during a given timeframe, and the familial relationships between these subjects as a part of police investigation. Link analysis here provides the crucial relationships and associations between very many objects of different types that are not apparent from isolated pieces of information. Computer-assisted or fully automatic computer-based link analysis is increasingly employed by banks and insurance agencies in fraud detection, by telecommunication operators in telecommunication network analysis, by medical sector in epidemiology and pharmacology, in law enforcement investigations, by search engines for relevance rating (and conversely by the spammers for spamdexing and by business owners for search engine optimization), and everywhere else where relationships between many objects have to be analyzed.

Web link analysis

Several Web search ranking algorithms use link-based centrality metrics, including (in order of appearance) Marchiori's Hyper Search, Google's PageRank, Kleinberg's HITS algorithm, and the TrustRank algorithm. Link analysis is also conducted in information science and communication science in order to understand and extract information from the structure of collections of web pages. For example the analysis might be of the interlinking between politicians' web sites or blogs.

Centrality measures

Information about the relative importance of nodes and edges in a graph can be obtained through centrality measures, widely used in disciplines like sociology. For example, eigenvector centrality uses the eigenvectors of the adjacency matrix to determine nodes that tend to be frequently visited.

Spread of content in networks

Content in a complex network can spread via two major methods: conserved spread and non-conserved spread.^[2] In conserved spread, the total amount of content that enters a complex network remains constant as it passes through. The model of conserved spread can best be represented by a pitcher containing a fixed amount of water being poured into a series of funnels connected by tubes. Here, the pitcher represents the original source and the water is the content being spread. The funnels and connecting tubing represent the nodes and the connections between nodes, respectively. As the water passes from one funnel into another, the water disappears instantly from the funnel that was previously exposed to the water. In non-conserved spread, the amount of content changes as it enters and passes through a complex network. The model of non-conserved spread can best be represented by a continuously running faucet running through a series of funnels connected by tubes. Here, the amount of water from the original source is infinite. Also, any funnels that have been exposed to the water continue to experience the water even as it passes into successive funnels. The non-conserved model is the most suitable for explaining the transmission of most infectious diseases.

See also

- | | |
|--|---|
| <ul style="list-style-type: none"> • Complex network • Network science • Network topology • Small-world networks | <ul style="list-style-type: none"> • Social circles • Scale-free networks • Sequential dynamical systems |
|--|---|

Implementations

- Orange, a free data mining software suite, module orngNetwork
- Pajek, program for (large) network analysis and visualization

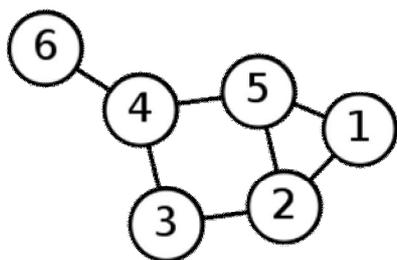
Notes

1. ^ Wasserman, Stanley and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press.
2. ^ Newman, M., Barabási, A.-L., Watts, D.J. [eds.] (2006) *The Structure and Dynamics of Networks*. Princeton, N.J.: Princeton University Press.

External links

- New Network Theory International Conference on 'New Network Theory'
- Network Workbench: A Large-Scale Network Analysis, Modeling and Visualization Toolkit
- Network analysis of computer networks
- Network analysis of organizational networks
- Network analysis of terrorist networks
- Network analysis of a disease outbreak
- Link Analysis: An Information Science Approach (book)
- Connected: The Power of Six Degrees (documentary)

3. Graph theory



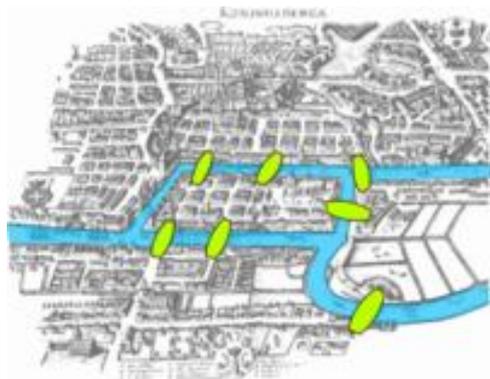
 A drawing of a graph

In mathematics and computer science, **graph theory** is the study of *graphs*: mathematical structures used to model pairwise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices or 'nodes' and a collection of *edges* that connect pairs of vertices. A graph may be *undirected*, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be *directed* from one vertex to another; see graph (mathematics) for more detailed definitions and for other variations in the types of graphs that are commonly

considered. The graphs studied in graph theory should not be confused with "graphs of functions" and other kinds of graphs.

Refer to Glossary of graph theory for basic definitions in graph theory.

History



The Königsberg Bridge problem

The paper written by Leonhard Euler on the *Seven Bridges of Königsberg* and published in 1736 is regarded as the first paper in the history of graph theory.^[1] This paper, as well as the one written by Vandermonde on the *knight problem*, carried on with the *analysis situs* initiated by Leibniz. Euler's formula relating the number of edges, vertices, and faces of a convex polyhedron was studied and generalized by Cauchy^[2] and L'Huilier,^[3] and is at the origin of topology.

More than one century after Euler's paper on the bridges of Königsberg and while Listing introduced topology, Cayley was led by the study of particular analytical forms arising from differential calculus to study a particular class of graphs, the *trees*. This study had many implications in theoretical chemistry. The involved techniques mainly concerned the enumeration of graphs having particular properties. Enumerative graph theory then rose from the results of Cayley and the fundamental results published by Pólya between 1935 and 1937 and the generalization of these by De Bruijn in 1959. Cayley linked his results on trees with the contemporary studies of chemical composition.^[4] The fusion of the ideas coming from mathematics with those coming from chemistry is at the origin of a part of the standard terminology of graph theory.

In particular, the term "graph" was introduced by Sylvester in a paper published in 1878 in *Nature*, where he draws an analogy between "quantic invariants" and "co-variants" of algebra and molecular diagrams:^[5]

"[...] Every invariant and co-variant thus becomes expressible by a *graph* precisely identical with a Kekuléan diagram or chemicograph. [...] I give a rule for the geometrical multiplication of graphs, *i.e.* for constructing a *graph* to the product of in- or co-variants whose separate graphs are given. [...]" (italics as in the original).

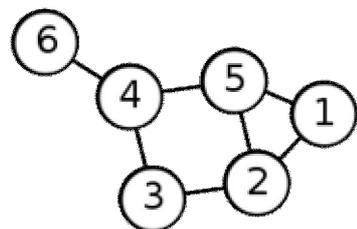
One of the most famous and productive problems of graph theory is the four color problem: "Is it true that any map drawn in the plane may have its regions colored with four colors, in such a way that any two regions having a common border have different colors?" This problem was first posed by Francis Guthrie in 1852 and its first written record is in a letter of De Morgan addressed to Hamilton the same year. Many incorrect proofs have been proposed, including those by Cayley, Kempe, and others. The study and the generalization of this problem by Tait, Heawood, Ramsey and Hadwiger led to the study of the colorings of the graphs embedded on surfaces with arbitrary genus. Tait's reformulation generated a new class of problems, the *factorization problems*, particularly studied by Petersen and König. The works of Ramsey on colorations and more specially the results obtained by Turán in 1941 was at the origin of another branch of graph theory, *extremal graph theory*.

The four color problem remained unsolved for more than a century. A proof produced in 1976 by Kenneth Appel and Wolfgang Haken,^{[6][7]} which involved checking the properties of 1,936 configurations by computer, was not fully accepted at the time due to its complexity. A simpler proof considering only 633 configurations was given twenty years later by Robertson, Seymour, Sanders and Thomas.^[8]

The autonomous development of topology from 1860 and 1930 fertilized graph theory back through the works of Jordan, Kuratowski and Whitney. Another important factor of common development of graph theory and topology came from the use of the techniques of modern algebra. The first example of such a use comes from the work of the physicist Gustav Kirchhoff, who published in 1845 his Kirchhoff's circuit laws for calculating the voltage and current in electric circuits.

The introduction of probabilistic methods in graph theory, especially in the study of Erdős and Rényi of the asymptotic probability of graph connectivity, gave rise to yet another branch, known as *random graph theory*, which has been a fruitful source of graph-theoretic results.

Vertex (graph theory)



A graph with 6 vertices and 7 edges

In graph theory, a **vertex** (plural **vertices**) or **node** is the fundamental unit out of which graphs are formed: an undirected graph consists of a set of vertices and a set of edges (unordered pairs of vertices), while a directed graph consists of a set of vertices and a set of arcs (ordered pairs of vertices). From the point of view of graph theory, vertices are treated as featureless and indivisible objects, although they may have additional structure depending on the application from which the graph arises; for instance, a semantic network is a graph in which the

vertices represent concepts or classes of objects.

The two vertices forming an edge are said to be its endpoints, and the edge is said to be incident to the vertices. A vertex w is said to be adjacent to another vertex v if the graph contains an edge (v,w) . The neighborhood of a vertex v is an induced subgraph of the graph, formed by all vertices adjacent to v .

The degree of a vertex in a graph is the number of edges incident to it. An **isolated vertex** is a vertex with degree zero; that is, a vertex that is not an endpoint of any edge. A **leaf vertex** (also **pendant vertex**) is a vertex with degree one. In a directed graph, one can distinguish the outdegree (number of outgoing edges) from the indegree (number of incoming edges); a **source vertex** is a vertex with indegree zero, while a **sink vertex** is a vertex with outdegree zero.

A cut vertex is a vertex the removal of which would disconnect the remaining graph; a vertex separator is a collection of vertices the removal of which would disconnect the remaining graph into small pieces. A k -vertex-connected graph is a graph in which removing fewer than k vertices always leaves the remaining graph connected. An independent set is a set of vertices no two of which are adjacent, and a vertex cover is a set of vertices that includes the endpoint of each edge in the graph. The vertex space of a graph is a vector space having a set of basis vectors corresponding with the graph's vertices.

A graph is vertex-transitive if it has symmetries that map any vertex to any other vertex. In the context of graph enumeration and graph isomorphism it is important to distinguish between **labeled vertices** and **unlabeled vertices**. A labeled vertex is a vertex that is associated with extra information that enables it to be distinguished from other labeled vertices; two graphs can be considered isomorphic only if the correspondence between their vertices pairs up vertices with equal labels. An unlabeled vertex is one that can be substituted for any other vertex based only on its adjacencies in the graph and not based on any additional information.

Vertices in graphs are analogous to, but not the same as, vertices of polyhedra: the skeleton of a polyhedron forms a graph, the vertices of which are the vertices of the polyhedron, but polyhedron vertices have additional structure (their geometric location) that is not assumed to be present in graph theory. The vertex figure of a vertex in a polyhedron is analogous to the neighborhood of a vertex in a graph.

In a directed graph, the forward star of a node u is defined as its outgoing edges. In a Graph G with the set of vertices V and the set of edges E , the forward star of u can be described as $\{(u, v) \in E\}$.

Drawing graphs

Graphs are represented graphically by drawing a dot for every vertex, and drawing an arc between two vertices if they are connected by an edge. If the graph is directed, the direction is indicated by drawing an arrow.

A graph drawing should not be confused with the graph itself (the abstract, non-visual structure) as there are several ways to structure the graph drawing. All that matters is which vertices are connected to which others by how many edges and not the exact layout. In practice it is often difficult to decide if two drawings represent the same graph. Depending on the problem domain some layouts may be better suited and easier to understand than others.

Graph-theoretic data structures

There are different ways to store graphs in a computer system. The data structure used depends on both the graph structure and the algorithm used for manipulating the graph. Theoretically one can distinguish between list and matrix structures but in concrete applications the best structure is often a combination of both. List structures are often preferred for sparse graphs as they have smaller memory requirements. Matrix structures on the other hand provide faster access for some applications but can consume huge amounts of memory.

List structures

Incidence list

The edges are represented by an array containing pairs (tuples if directed) of vertices (that the edge connects) and possibly weight and other data. Vertices connected by an edge are said to be *adjacent*.

Adjacency list

Much like the incidence list, each vertex has a list of which vertices it is adjacent to. This causes redundancy in an undirected graph: for example, if vertices A and B are adjacent, A's adjacency list contains B, while B's list contains A. Adjacency queries are faster, at the cost of extra storage space.

Matrix structures

Incidence matrix

The graph is represented by a matrix of size $|V|$ (number of vertices) by $|E|$ (number of edges) where the entry [vertex, edge] contains the edge's endpoint data (simplest case: 1 - connected, 0 - not connected).

Adjacency matrix

This is the n by n matrix A , where n is the number of vertices in the graph. If there is an edge from some vertex x to some vertex y , then the element $a_{x,y}$ is 1 (or in general the number of xy edges), otherwise it is 0. In computing, this matrix makes it easy to find subgraphs, and to reverse a directed graph.

Laplacian matrix or Kirchhoff matrix or Admittance matrix

This is defined as $D - A$, where D is the diagonal degree matrix. It explicitly contains both adjacency information and degree information.

Distance matrix

A symmetric n by n matrix D whose element $d_{x,y}$ is the length of a shortest path between x and y ; if there is no such path $d_{x,y} = \text{infinity}$. It can be derived from powers of A

$$d_{x,y} = \min\{n \mid A^n[x, y] \neq 0\}.$$

Problems in graph theory

Enumeration

There is a large literature on graphical enumeration: the problem of counting graphs meeting specified conditions. Some of this work is found in Harary and Palmer (1973).

Subgraphs, induced subgraphs, and minors

A common problem, called the subgraph isomorphism problem, is finding a fixed graph as a subgraph in a given graph. One reason to be interested in such a question is that many graph properties are *hereditary* for subgraphs, which means that a graph has the property if and only if all subgraphs have it too. Unfortunately, finding maximal subgraphs of a certain kind is often an NP-complete problem.

- Finding the largest complete graph is called the clique problem (NP-complete).

A similar problem is finding induced subgraphs in a given graph. Again, some important graph properties are hereditary with respect to induced subgraphs, which means that a graph has a property if and only if all induced subgraphs also have it. Finding maximal induced subgraphs of a certain kind is also often NP-complete. For example,

- Finding the largest edgeless induced subgraph, or independent set, called the independent set problem (NP-complete).

Still another such problem, the *minor containment problem*, is to find a fixed graph as a minor of a given graph. A minor or **subcontraction** of a graph is any graph obtained by taking a subgraph and contracting some (or no) edges. Many graph properties are hereditary for minors, which means that a graph has a property if and only if all minors have it too. A famous example:

- A graph is planar if it contains as a minor neither the complete bipartite graph $K_{3,3}$ (See the Three-cottage problem) nor the complete graph K_5 .

Another class of problems has to do with the extent to which various species and generalizations of graphs are determined by their *point-deleted subgraphs*, for example:

- The reconstruction conjecture

Graph coloring

- | | |
|--|--|
| <ul style="list-style-type: none"> • The four-color theorem • The strong perfect graph theorem • The Erdős–Faber–Lovász conjecture (unsolved) • The total coloring conjecture (unsolved) | <ul style="list-style-type: none"> • The list coloring conjecture (unsolved) • The Hadwiger conjecture (graph theory) (unsolved) |
|--|--|

Route problems

- | | |
|---|--|
| <ul style="list-style-type: none"> • Hamiltonian path and cycle problems • Minimum spanning tree • Route inspection problem (also called the "Chinese Postman Problem") • Seven Bridges of Königsberg | <ul style="list-style-type: none"> • Shortest path problem • Steiner tree • Three-cottage problem • Traveling salesman problem (NP-complete) |
|---|--|

Network flow

There are numerous problems arising especially from applications that have to do with various notions of flows in

networks, for example: Max flow min cut theorem

Visibility graph problems

- Museum guard problem

Covering problems

Covering problems are specific instances of subgraph-finding problems, and they tend to be closely related to the clique problem or the independent set problem.

- Set cover problem
- Vertex cover problem

Graph classes

Many problems involve characterizing the members of various classes of graphs. Overlapping significantly with other types in this list, this type of problem includes, for instance:

- Enumerating the members of a class
- Characterizing a class in terms of forbidden substructures
- Ascertaining relationships among classes (e.g., does one property of graphs imply another)
- Finding efficient algorithms to decide membership in a class
- Finding representations for members of a class

Applications

Applications of graph theory are primarily, but not exclusively, concerned with labeled graphs and various specializations of these.

Structures that can be represented as graphs are ubiquitous, and many problems of practical interest can be represented by graphs. The link structure of a website could be represented by a directed graph: the vertices are the web pages available at the website and a directed edge from page A to page B exists if and only if A contains a link to B. A similar approach can be taken to problems in travel, biology, computer chip design, and many other fields. The development of algorithms to handle graphs is therefore of major interest in computer science. There, the transformation of graphs is often formalized and represented by graph rewrite systems. They are either directly used or properties of the rewrite systems (e.g. confluence) are studied.

A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights, or weighted graphs, are used to represent structures in which pairwise connections have some numerical values. For example if a graph represents a road network, the weights could represent the length of each road. A digraph with weighted edges in the context of graph theory is called a network.

Networks have many uses in the practical side of graph theory, network analysis (for example, to model and analyze traffic networks). Within network analysis, the definition of the term "network" varies, and may often refer to a simple graph.

Many applications of graph theory exist in the form of network analysis. These split broadly into three categories:

1. First, analysis to determine structural properties of a network, such as the distribution of vertex degrees and the diameter of the graph. A vast number of graph measures exist, and the production of useful ones for various domains remains an active area of research.
2. Second, analysis to find a measurable quantity within the network, for example, for a transportation network, the level of vehicular flow within any portion of it.
3. Third, analysis of dynamical properties of networks.

Graph theory is also used to study molecules in chemistry and physics. In condensed matter physics, the three dimensional structure of complicated simulated atomic structures can be studied quantitatively by gathering statistics on graph-theoretic properties related to the topology of the atoms. For example, Franzblau's shortest-path (SP) rings. In chemistry a graph makes a natural model for a molecule, where vertices represent atoms and edges bonds. This approach is especially used in computer processing of molecular structures, ranging from chemical editors to database searching.

Graph theory is also widely used in sociology as a way, for example, to measure actors' prestige or to explore diffusion mechanisms, notably through the use of social network analysis software.

Likewise, graph theory is useful in biology and conservation efforts where a vertex can represent regions where certain species exist (or habitats) and the edges represent migration paths, or movement between the regions. This information is important when looking at breeding patterns or tracking the spread of disease, parasites or how changes to the movement can affect other species.

See also

- Gallery of named graphs
- Glossary of graph theory
- List of graph theory topics
- Publications in graph theory

Related topics

- Graph property
- Algebraic graph theory
- Conceptual graph
- Data structure
- Disjoint-set data structure
- Entitative graph
- Existential graph
- Graph data structure
- Graph algebras
- Graph automorphism
- Graph coloring
- Graph database
- Graph drawing
- Graph equation
- Graph rewriting
- Intersection graph
- Logical graph
- Loop
- Null graph
- Perfect graph
- Quantum graph
- Spectral graph theory
- Strongly regular graphs
- Symmetric graphs
- Tree data structure

Algorithms

- Bellman-Ford algorithm
- Dijkstra's algorithm
- Ford-Fulkerson algorithm
- Kruskal's algorithm
- Nearest neighbour algorithm
- Prim's algorithm
- Depth-first search
- Breadth-first search

Subareas

- Algebraic graph theory
- Geometric graph theory
- Extremal graph theory
- Probabilistic graph theory
- Topological graph theory

Related areas of mathematics

- Combinatorics
- Group theory
- Knot theory
- Ramsey theory

Generalizations

- Hypergraph
- Abstract simplicial complex

Prominent graph theorists

- Berge, Claude
- Bollobás, Béla
- Chung, Fan
- Dirac, Gabriel Andrew
- Erdős, Paul
- Euler, Leonhard
- Faudree, Ralph
- Golumbic, Martin
- Graham, Ronald
- Harary, Frank
- Heawood, Percy John
- König, Dénes
- Lovász, László
- Nešetřil, Jaroslav
- Rényi, Alfréd
- Ringel, Gerhard
- Robertson, Neil
- Seymour, Paul
- Szemerédi, Endre
- Thomas, Robin
- Thomassen, Carsten
- Turán, Pál
- Tutte, W. T.

Notes

1. [^] Biggs, N.; Lloyd, E. and Wilson, R. (1986). *Graph Theory, 1736-1936*. Oxford University Press.
2. [^] Cauchy, A.L. (1813). "Recherche sur les polyèdres - premier mémoire". *Journal de l'Ecole Polytechnique* **9 (Cahier 16)**: 66–86.
3. [^] L'Huilier, S.-A.-J. (1861). "Mémoire sur la polyèdrométrie". *Annales de Mathématiques* **3**: 169–189.
4. [^] Cayley, A. (1875). "Ueber die Analytischen Figuren, welche in der Mathematik Bäume genannt werden und

ihre Anwendung auf die Theorie chemischer Verbindungen". *Berichte der deutschen Chemischen Gesellschaft* **8**: 1056–1059. doi:10.1002/cber.18750080252.

5. ^ John Joseph Sylvester (1878), *Chemistry and Algebra*. Nature, volume 17, page 284. doi:10.1038/017284a0. Online version accessed on 2009-12-30.
6. ^ Appel, K. and Haken, W. (1977). "Every planar map is four colorable. Part I. Discharging". *Illinois J. Math.* **21**: 429–490.
7. ^ Appel, K. and Haken, W. (1977). "Every planar map is four colorable. Part II. Reducibility". *Illinois J. Math.* **21**: 491–567.
8. ^ Robertson, N.; Sanders, D.; Seymour, P. and Thomas, R. (1997). "The four color theorem". *Journal of Combinatorial Theory Series B* **70**: 2–44. doi:10.1006/jctb.1997.1750.

References

- Berge, Claude (1958), *Théorie des graphes et ses applications*, Collection Universitaire de Mathématiques, **II**, Paris: Dunod. English edition, Wiley 1961; Methuen & Co, New York 1962; Russian, Moscow 1961; Spanish, Mexico 1962; Roumanian, Bucharest 1969; Chinese, Shanghai 1963; Second printing of the 1962 first English edition, Dover, New York 2001.
- Biggs, N.; Lloyd, E.; Wilson, R. (1986), *Graph Theory, 1736–1936*, Oxford University Press.
- Bondy, J.A.; Murty, U.S.R. (2008), *Graph Theory*, Springer, ISBN 978-1-84628-969-9.
- Chartrand, Gary (1985), *Introductory Graph Theory*, Dover, ISBN 0-486-24775-9.
- Gibbons, Alan (1985), *Algorithmic Graph Theory*, Cambridge University Press.
- Golumbic, Martin (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press.
- Harary, Frank (1969), *Graph Theory*, Reading, MA: Addison-Wesley.
- Harary, Frank; Palmer, Edgar M. (1973), *Graphical Enumeration*, New York, NY: Academic Press.
- Mahadev, N.V.R.; Peled, Uri N. (1995), *Threshold Graphs and Related Topics*, North-Holland.
- Gallo, Giorgio; Pallotino, Stefano (December 1988). "Shortest Path Algorithms" (PDF). *Annals of Operations Research* (Netherlands: Springer) **13** (1): 1–79. doi:10.1007/BF02288320. <http://www.springerlink.com/content/awn535w405321948/>. Retrieved 2008-06-18.
- Chartrand, Gary (1985). *Introductory graph theory*. New York: Dover. ISBN 0-486-24775-9.
- Biggs, Norman; Lloyd, E. H.; Wilson, Robin J. (1986). *Graph theory, 1736-1936*. Oxford [Oxfordshire]: Clarendon Press. ISBN 0-19-853916-9.
- Harary, Frank (1969). *Graph theory*. Reading, Mass.: Addison-Wesley Publishing. ISBN 0-201-41033-8.
- Harary, Frank; Palmer, Edgar M. (1973). *Graphical enumeration*. New York, Academic Press. ISBN 0-12-324245-2.

Online textbooks

- Graph Theory with Applications (1976) by Bondy and Murty
- Phase Transitions in Combinatorial Optimization Problems, Section 3: Introduction to Graphs (2006) by Hartmann and Weigt
- Digraphs: Theory Algorithms and Applications 2007 by J. Bang-Jensen and G. Gutin
- Graph Theory, by Reinhard Diestel
- Weisstein, Eric W., "Graph Vertex" from MathWorld.

Other resources

- Image gallery: graphs
- Concise, annotated list of graph theory resources for researchers

4. Complex network

In the context of network theory, a **complex network** is a network (graph) with non-trivial topological features—features that do not occur in simple networks such as lattices or random graphs. The study of complex networks is a young and active area of scientific research inspired largely by the empirical study of real-world networks such as computer networks and social networks.

Definition

Most social, biological, and technological networks display substantial non-trivial topological features, with patterns of connection between their elements that are neither purely regular nor purely random. Such features include a heavy tail in the degree distribution, a high clustering coefficient, assortativity or disassortativity among

vertices, community structure, and hierarchical structure. In the case of directed networks these features also include reciprocity, triad significance profile and other features. In contrast, many of the mathematical models of networks that have been studied in the past, such as lattices and random graphs, do not show these features.

Two well-known and much studied classes of complex networks are scale-free networks and small-world networks, whose discovery and definition are canonical case-studies in the field. Both are characterized by specific structural features—power-law degree distributions for the former and short path lengths and high clustering for the latter. However, as the study of complex networks has continued to grow in importance and popularity, many other aspects of network structure have attracted attention as well.

The field continues to develop at a brisk pace, and has brought together researchers from many areas including mathematics, physics, biology, computer science, sociology, epidemiology, and others. Ideas from network science have been applied to the analysis of metabolic and genetic regulatory networks, the design of robust and scalable communication networks both wired and wireless, the development of vaccination strategies for the control of disease, and a broad range of other practical issues. Research on networks has seen regular publication in some of the most visible scientific journals and vigorous funding in many countries, has been the topic of conferences in a variety of different fields, and has been the subject of numerous books both for the lay person and for the expert.

Scale-free networks

A network is named scale-free if its degree distribution, i.e., the probability that a node selected uniformly at random has a certain number of links (degree), follows a particular mathematical function called a power law. The power law implies that the degree distribution of these networks has no characteristic scale. In contrast, network with a single well-defined scale are somewhat similar to a lattice in that every node has (roughly) the same degree. Examples of networks with a single scale include the Erdős–Rényi random graph and hypercubes. In a network with a scale-free degree distribution, some vertices have a degree that is orders of magnitude larger than the average - these vertices are often called "hubs", although this is a bit misleading as there is no inherent threshold above which a node can be viewed as a hub. If there were, then it wouldn't be a scale-free distribution!

Interest in scale-free networks began in the late 1990s with the apparent discovery of a power-law degree distribution in many real world networks such as the World Wide Web, the network of Autonomous systems (ASs), some network of Internet routers, protein interaction networks, email networks, etc. Although many of these distributions are not unambiguously power laws, their breadth, both in degree and in domain, shows that networks exhibiting such a distribution are clearly very different from what you would expect if edges existed independently and at random (a Poisson distribution). Indeed, there are many different ways to build a network with a power-law degree distribution. The Yule process is a canonical generative process for power laws, and has been known since 1925. However, it is known by many other names due to its frequent reinvention, e.g., The Gibrat principle by Herbert Simon, the Matthew effect, cumulative advantage and, most recently, preferential attachment by Barabási and Albert for power-law degree distributions.

Networks with a power-law degree distribution can be highly resistant to the random deletion of vertices, i.e., the vast majority of vertices remain connected together in a giant component. Such networks can also be quite sensitive to targeted attacks aimed at fracturing the network quickly. When the graph is uniformly random except for the degree distribution, these critical vertices are the ones with the highest degree, and have thus been implicated in the spread of disease (natural and artificial) in social and communication networks, and in the spread of fads (both of which are modeled by a percolation or branching process).

Small-world networks

A network is called a small-world network by analogy with the small-world phenomenon (popularly known as six degrees of separation). The small world hypothesis, which was first described by the Hungarian writer Frigyes Karinthy in 1929, and tested experimentally by Stanley Milgram (1967), is the idea that two arbitrary people are connected by only six degrees of separation, i.e. the diameter of the corresponding graph of social connections is not much larger than six. In 1998, Duncan J. Watts and Steven Strogatz published the first small-world network model, which through a single parameter smoothly interpolates between a random graph to a lattice. Their model demonstrated that with the addition of only a small number of long-range links, a regular graph, in which the diameter is proportional to the size of the network, can be transformed into a "small world" in which the average number of edges between any two vertices is very small (mathematically, it should grow as the logarithm of the size of the network), while the clustering coefficient stays large. It is known that a wide variety of abstract graphs exhibit the small-world property, e.g., random graphs and scale-free networks. Further, real world networks such as the World Wide Web and the metabolic network also exhibit this property.

In the scientific literature on networks, there is some ambiguity associated with the term "small world." In

addition to referring to the size of the diameter of the network, it can also refer to the co-occurrence of a small diameter and a high clustering coefficient. The clustering coefficient is a metric that represents the density of triangles in the network. For instance, sparse random graphs have a vanishingly small clustering coefficient while real world networks often have a coefficient significantly larger. Scientists point to this difference as suggesting that edges are correlated in real world networks.

Researchers and scientists

- | | |
|---|---|
| <ul style="list-style-type: none"> • Réka Albert • Luís Amaral • Alex Arenas • Albert-László Barabási • Alain Barrat • Marc Barthelemy • Stefano Boccaletti • Dirk Brockmann • Guido Caldarelli • Roger Guimerà | <ul style="list-style-type: none"> • Shlomo Havlin • Jon Kleinberg • José Mendes • Yamir Moreno • Adilson E. Motter • Mark Newman • Sidney Redner • Steven Strogatz • Alessandro Vespignani • Duncan J. Watts |
|---|---|

See also

- | | |
|---|--|
| <ul style="list-style-type: none"> • Network theory • Network science | <ul style="list-style-type: none"> • Dynamic Network Analysis • Complexity Science |
|---|--|

Books

- Albert-László Barabási, *Linked: How Everything is Connected to Everything Else*, 2004, ISBN 0-452-28439-2
- Alain Barrat, Marc Barthelemy, Alessandro Vespignani, *Dynamical processes in complex networks*, Cambridge University Press, 2008, ISBN 978-0-521-87950-7
- Stefan Bornholdt (Editor) and Heinz Georg Schuster (Editor), *Handbook of Graphs and Networks: From the Genome to the Internet*, 2003, ISBN 3-527-40336-1
- Guido Caldarelli, *Scale-Free Networks* Oxford University Press, 2007, ISBN 0-19-921151-7
- Matthias Dehmer and Frank Emmert-Streib (Eds.), "Analysis of Complex Networks: From Biology to Linguistics", Wiley-VCH, 2009, ISBN 3-527-32345-7
- S.N. Dorogovtsev and J.F.F. Mendes, *Evolution of Networks: From biological networks to the Internet and WWW*, Oxford University Press, 2003, ISBN 0-19-851590-1
- Mark Newman, Albert-László Barabási, and Duncan J. Watts, *The Structure and Dynamics of Networks*, Princeton University Press, Princeton, 2006, ISBN 978-0-691-11357-9
- R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A statistical physics approach*, Cambridge University Press, 2004, ISBN 0-521-82698-5
- Duncan J. Watts, *Six Degrees: The Science of a Connected Age*, Norton & Company, 2003, ISBN 0-393-04142-5
- Duncan J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness*, Princeton University Press, 2003, ISBN 0-691-11704-7

References

- D. J. Watts and S. H. Strogatz., *Collective dynamics of 'small-world' networks*, Nature Vol 393 (1998) 440-442
- S. H. Strogatz, *Exploring Complex Networks*, Nature Vol 410 (2001) 268-276
- R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks" *Reviews of Modern Physics* 74, (2002) 47
- S. N. Dorogovtsev and J.F.F. Mendes, *Evolution of Networks*, Adv. Phys. 51, 1079 (2002)
- M. E. J. Newman, The structure and function of complex networks, *SIAM Review* 45, 167-256 (2003)
- A. Barabasi and E. Bonabeau, *Scale-Free Networks*, *Scientific American*, (May 2003), 50-59
- S. Boccaletti et al., *Complex Networks: Structure and Dynamics*, *Phys. Rep.*, 424 (2006), 175-308.
- S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, *Critical phenomena in complex networks*, *Rev. Mod. Phys.* 80, 1275, (2008)
- R. Cohen, K. Erez, D. ben-Avraham, S. Havlin, "Resilience of the Internet to random breakdown" *Phys. Rev. Lett.* 85, 4626 (2000).

External links

- Network Science — United States Military Academy - Network Science Center

- Resources in Complex Networks — University of São Paulo - Institute of Physics at São Carlos
- Cx-Nets — Complex Networks Collaboratory
- GNET — Group of Complex Systems & Random Networks
- UCLA Human Complex Systems Program
- New England Complex Systems Institute
- Santa Fe Institute Networks Group
- Barabasi Networks Group
- Cosin Project Codes, Papers and Data on Complex Networks
- Complex network on arxiv.org
- [1] — Anna Nagurney's Virtual Center for Supernetworks
- BIOREL resource for quantitative estimation of the network bias in relation to external information
- Complexity Virtual Laboratory (VLAB)
- complexnetworks.fr — French computer science research group on networks

5. Flow network

In graph theory, a **flow network** is a directed graph where each edge has a **capacity** and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. Often in Operations Research, a directed graph is called a **network**, the vertices are called **nodes** and the edges are called **arcs**. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, except when it is a **source**, which has more outgoing flow, or **sink**, which has more incoming flow. A network can be used to model traffic in a road system, fluids in pipes, currents in an electrical circuit, or anything similar in which something travels through a network of nodes.

Definition

Suppose $G(V, E)$ is a finite directed graph in which every edge $(u, v) \in E$ has a non-negative, real-valued capacity $c(u, v)$. If $(u, v) \notin E$, we assume that $c(u, v) = 0$. We distinguish two vertices: a source s and a sink t . A flow network is a real function $f : V \times V \rightarrow \mathbb{R}$ with the following three properties for all nodes u and v :

Capacity constraints: $f(u, v) \leq c(u, v)$. The flow along an edge can not exceed its capacity.

Skew symmetry: $f(u, v) = -f(v, u)$. Flow from u to v must be the opposite of the from v to u .

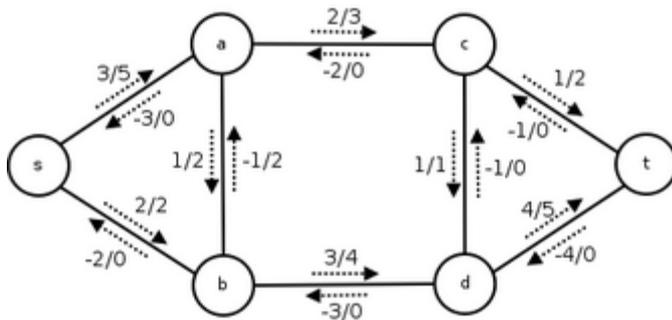
Flow conservation: $\sum_{w \in V} f(u, w) = 0$ unless $u = s$ or $w = t$. The net flow to a node is zero, except for the source, which "produces" flow, and the sink, which "consumes" flow.

Notice that $f(u, v)$ is the net flow from u to v . If the graph represents a physical network, and if there is a real capacity of, for example, 4 units from u to v , and a real flow of 3 units from v to u , we have $f(u, v) = 1$ and $f(v, u) = -1$.

The **residual capacity** of an edge is $c_f(u, v) = c(u, v) - f(u, v)$. This defines a **residual network** denoted $G_f(V, E_f)$, giving the amount of *available* capacity. See that there can be an edge from u to v in the residual network, even though there is no edge from u to v in the original network. Since flows in opposite directions cancel out, *decreasing* the flow from v to u is the same as *increasing* the flow from u to v . An **augmenting path** is a path (u_1, u_2, \dots, u_k) in the residual network, where $u_1 = s$, $u_k = t$, and $c_f(u_i, u_{i+1}) > 0$. A network is at maximum flow if and only if there is no augmenting path in the residual network.

Example

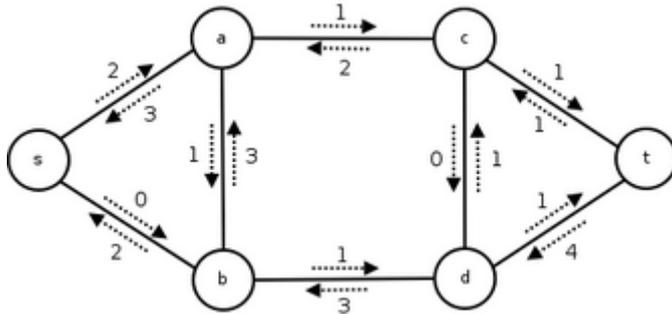
A flow network showing flow and capacity.



To the right you see a flow network with source labeled s , sink t , and four additional nodes. The flow and capacity is denoted f / c . Notice how the network upholds skew symmetry, capacity constraints and flow conservation. The total amount of flow from s to t is 5, which can be easily seen from the fact that the total outgoing flow from s is 5, which is also the incoming flow to t . We know that no flow appears or disappears in any of the other nodes.

Residual network for the above flow network, showing residual capacities.

Below you see the residual network for the given flow. Notice how there is positive residual capacity on some edges where the original capacity is zero, for example for the edge (d,c) . This flow is not a maximum flow. There is available capacity along the paths (s,a,c,t) , (s,a,b,d,t) and (s,a,b,d,c,t) , which are then the augmenting paths. The residual capacity of the first path is $\min(c(s,a) - f(s,a), c(a,c) - f(a,c), c(c,t) - f(c,t)) = \min(5 - 3, 3 - 2, 2 - 1) = \min(2, 1, 1) = 1$. Notice that augmenting path (s,a,b,d,c,t) does not exist in the original network, but you can send flow along it, and still get a legal flow.



If this is a real network, there might actually be a flow of 2 from a to b , and a flow of 1 from b to a , but we only maintain the **net** flow.

Applications

Picture a series of water pipes, fitting into a network. Each pipe is of a certain diameter, so it can only maintain a flow of a certain amount of water. Anywhere that pipes meet, the total amount of water coming into that junction must be equal to the amount going out, otherwise we would quickly run out of water, or we would have a build up of water. We have a water inlet, which is the source, and an outlet, the sink. A flow would then be one possible way for water to get from source to sink so that the total amount of water coming out of the outlet is consistent. Intuitively, the total flow of a network is the rate at which water comes out of the outlet.

Flows can pertain to people or material over transportation networks, or to electricity over electrical distribution systems. For any such physical network, the flow coming into any intermediate node needs to equal the flow going out of that node. Bollobás characterizes this constraint in terms of Kirchhoff's current law, while later authors (ie: Chartrand) mention its generalization to some conservation equation.

Flow networks also find applications in ecology: flow networks arise naturally when considering the flow of nutrients and energy between different organizations in a food web. The mathematical problems associated with such networks are quite different from those that arise in networks of fluid or traffic flow. The field of ecosystem network analysis, developed by Robert Ulanowicz and others, involves using concepts from information theory and thermodynamics to study the evolution of these networks over time..

Generalizations and specializations

The simplest and most common problem using flow networks is to find what is called the maximum flow, which provides the largest possible total flow from the source to the sink in a given graph. There are many other problems which can be solved using max flow algorithms, if they are appropriately modeled as flow networks, such as bipartite matching, the assignment problem and the transportation problem.

In a multi-commodity flow problem, you have multiple sources and sinks, and various "commodities" which are to flow from a given source to a given sink. This could be for example various goods that are produced at various factories, and are to be delivered to various given customers through the *same* transportation network.

In a minimum cost flow problem, each edge u,v has a given cost $k(u,v)$, and the cost of sending the flow $f(u,v)$ across the edge is $f(u,v) \cdot k(u,v)$. The objective is to send a given amount of flow from the source to the sink, at the lowest possible price.

In a circulation problem, you have a lower bound $l(u,v)$ on the edges, in addition to the upper bound $c(u,v)$. Each edge also has a cost. Often, flow conservation holds for *all* nodes in a circulation problem, and there is a connection from the sink back to the source. In this way, you can dictate the total flow with $l(t,s)$ and $c(t,s)$. The flow *circulates* through the network, hence the name of the problem.

In a **network with gains** or **generalized network** each edge has a **gain**, a real number (not zero) such that, if the edge has gain g , and an amount x flows into the edge at its tail, then an amount gx flows out at the head.

See also

- | | |
|--|---|
| <ul style="list-style-type: none"> • Constructal theory • Ford-Fulkerson algorithm • Flow (computer networking) | <ul style="list-style-type: none"> • Max-flow min-cut theorem • Oriented matroid • Shortest path problem |
|--|---|

Further reading

- Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin (1993). *Network Flows: Theory, Algorithms and Applications*. Prentice Hall. ISBN 0-13-617549-X.
- Bollobás, Béla (1979). *Graph Theory: An Introductory Course*. Heidelberg: Springer-Verlag. ISBN 3-540-90399-2.
- Chartrand, Gary & Oellermann, Ortrud R. (1993). *Applied and Algorithmic Graph Theory*. New York: McGraw-Hill. ISBN 0-07-557101-3.
- Even, Shimon (1979). *Graph Algorithms*. Rockville, Maryland: Computer Science Press. ISBN 0-914894-21-8.
- Gibbons, Alan (1985). *Algorithmic Graph Theory*. Cambridge: Cambridge University Press. ISBN 0-521-28881-9 ISBN 0-521-24659-8.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001) [1990]. "26". *Introduction to Algorithms* (2nd edition ed.). MIT Press and McGraw-Hill. pp. 696–697. ISBN 0-262-03293-7.

External links

- Maximum Flow Problem
- Real graph instances
- Software, papers, test graphs, etc.
- Solutions for network flow problems
- Software and papers for network flow problems
- Lemon C++ library with several maximum flow and minimum cost circulation algorithms

6. Network diagram

A **network diagram** is a general type of diagram, which represents some kind of network. A network in general is an interconnected group or system, or a fabric or structure of fibrous elements attached to each other at regular intervals, or formally: a graph.

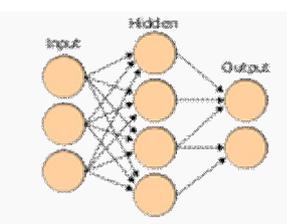
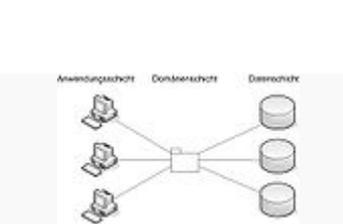
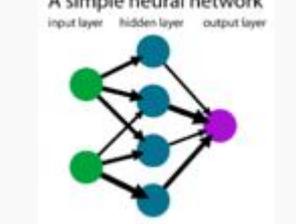
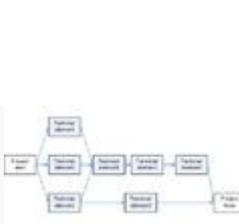
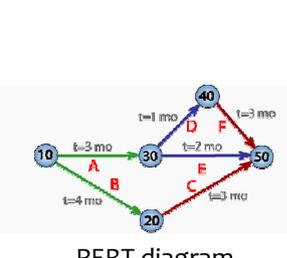
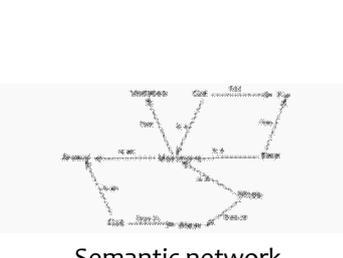
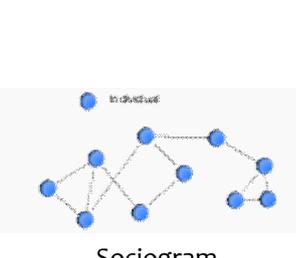
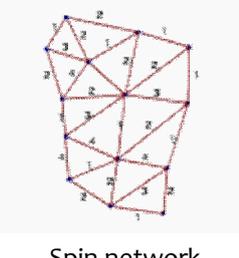
A network diagram is a special kind of cluster diagram, which even more general represents any cluster or small group or bunch of something, structured or not. Both the flow diagram and the tree diagram can be seen as a specific type of network diagram.

There are different types network diagrams:

- Artificial neural network or "neural network" (NN), is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation.
- Computer network diagram is a schematic depicting the nodes and connections amongst nodes in a computer network or, more generally, any telecommunications network.
- In project management according to Baker et al. (2003), a "network diagram is the logical representation of activities, that defines the sequence or the work of a project. It shows the path of a project, lists starting and completion dates, and names the responsibilities for each task. At a glance it explains how the work of the project goes together... A network for a simple project might consist one or two pages, and on a larger project several network diagrams may exist".^[1] Specific diagrams here are
 - Project network: a general flow chart depicting the sequence in which a project's terminal elements are to be completed by showing terminal elements and their dependencies.
 - PERT network

- Neural network diagram: ia a network or circuit of biological neurons or artificial neural networks, which are composed of artificial neurons or nodes.
- A semantic network is a network or circuit of biological neurons. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes]].^[2]
- A sociogram is a graphic representation of social links that a person has. It is a sociometric chart that plots the structure of interpersonal relations in a group situation.

Gallery

 <p>Artificial neural network</p>	 <p>Computer network diagram</p>	 <p>Neural network diagram</p>	 <p>Project network</p>
 <p>PERT diagram</p>	 <p>Semantic network</p>	 <p>Sociogram</p>	 <p>Spin network</p>

Network topologies

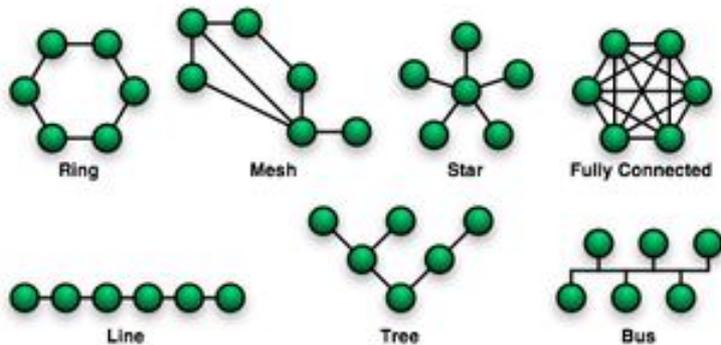


Diagram of different network topologies.

In computer science the elements of a network are arranged in certain basic shapes (see figure):

- Ring: The ring network connects each node to exactly two other nodes, forming a circular pathway for activity or signals - a ring. The interaction or data travels from node to node, with each node handling every packet.
- Mesh is a way to route data, voice and instructions between nodes. It allows for continuous connections and reconfiguration around broken or blocked paths by “hopping” from node to node until the destination is reached.
- Star: The star network consists of one central element, switch, hub or computer, which acts as a conduit to coordinate activity or transmit messages.
- Fully connected: Every node is connected to every other node.
- Line - Everything connected in a single line.
- Tree: This consists of tree-configured nodes connected to switches/concentrators, each connected to a linear bus backbone. Each hub rebroadcasts all transmissions received from any peripheral node to all peripheral nodes on the network, sometimes including the originating node. All peripheral nodes may thus communicate with all others by transmitting to, and receiving from, the central node only.
- Bus: In this network architecture a set of clients are connected via a shared communications line, called a bus.

Related topics

Network theory

Network theory is an area of applied mathematics and part of graph theory. It has application in many disciplines including particle physics, computer science, biology, economics, operations research, and sociology. Network theory concerns itself with the study of graphs as a representation of either symmetric relations or, more generally, of asymmetric relations between discrete objects. Examples of which include logistical networks, the World Wide Web, gene regulatory networks, metabolic networks, social networks, epistemological networks, etc. See list of network theory topics for more examples.

Network topology

Network topology is the study of the arrangement or mapping of the elements (links, nodes, etc.) of a network, especially the physical (real) and logical (virtual) interconnections between nodes.^[4]

Any particular network topology is determined only by the graphical mapping of the configuration of physical and/or logical connections between nodes. LAN Network Topology is, therefore, technically a part of graph theory. Distances between nodes, physical interconnections, transmission rates, and/or signal types may differ in two networks and yet their topologies may be identical.

See also

- Computer network
- Neural network
- Radio network
- Television network
- Telecommunications network
- Semantic network
- Social network
- System context diagram

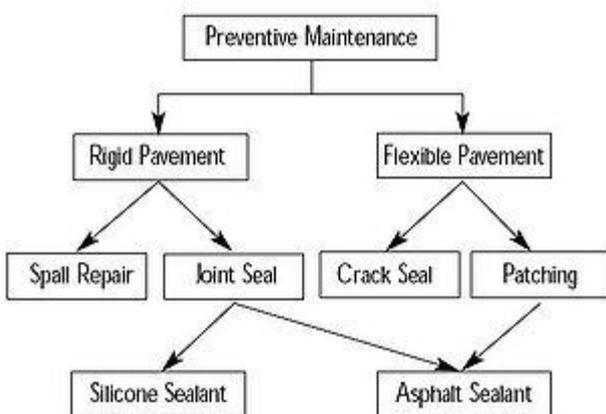
References

1. ^ Sunny Baker, G. Michael Campbell, Kim Baker (2003). *The Complete Idiot's Guide to Project Management*. pp. 104. ISBN 0028639200.
2. ^ Jonh F.Sowa (1987). "Semantic Networks". in Stuart C Shapiro. *Encyclopedia of Artificial Intelligence*. <http://www.jfsowa.com/pubs/semnet.htm>. Retrieved 2008-04-29.
3. ^ Committee on Network Science for Future Army Applications, National Research Council (2005). *Network Science*. National Academies Press. ISBN 0309100267. http://www.nap.edu/catalog.php?record_id=11516.
4. ^ Groth, David; Toby Skandier (2005). *Network+ Study Guide, Fourth Edition*. Sybex, Inc.. ISBN 0-7821-4406-3.

7. Network model (database)

The **network model** is a database model conceived as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice.

Network Model



The network model is a database model conceived as a flexible way of representing objects and their relationships. Its original inventor was Charles Bachman, and it was developed into a standard specification published in 1969 by the CODASYL Consortium. Where the hierarchical model structures data as a tree of records, with each record having one parent record and many children, the network model allows each record to have multiple parent and child records, forming a lattice structure.

Example of a Network Model.

The network model's original inventor was Charles Bachman, and it was developed into a standard specification published in 1969 by the CODASYL Consortium.

Overview

Where the hierarchical model structures data as a tree of records, with each record having one parent record and many children, the network model allows each record to have multiple parent and child records, forming a

generalized graph structure. This property applies at two levels: the schema is a generalized graph of record types connected by relationship types (called "set types" in CODASYL), and the database itself is a generalized graph of record occurrences connected by relationships (CODASYL "sets"). Cycles are permitted at both levels.

The chief argument in favour of the network model, in comparison to the hierarchic model, was that it allowed a more natural modeling of relationships between entities. Although the model was widely implemented and used, it failed to become dominant for two main reasons. Firstly, IBM chose to stick to the hierarchical model with semi-network extensions in their established products such as IMS and DL/I. Secondly, it was eventually displaced by the relational model, which offered a higher-level, more declarative interface. Until the early 1980s the performance benefits of the low-level navigational interfaces offered by hierarchical and network databases were persuasive for many large-scale applications, but as hardware became faster, the extra productivity and flexibility of the relational model led to the gradual obsolescence of the network model in corporate enterprise usage.

Some Well-known Network Databases

- | | |
|---|--|
| <ul style="list-style-type: none"> • Digital Equipment Corporation DBMS-10 • Digital Equipment Corporation DBMS-20 • Digital Equipment Corporation VAX DBMS • Honeywell IDS (Integrated Data Store) • IDMS (Integrated Database Management System) | <ul style="list-style-type: none"> • Raima Data Manager (RDM) Embedded • RDM Server • TurboIMAGE • Univac DMS-1100 |
|---|--|

History

In 1969, the Conference on Data Systems Languages (CODASYL) established the first specification of the network database model. This was followed by a second publication in 1971, which became the basis for most implementations. Subsequent work continued into the early 1980s, culminating in an ISO specification, but this had little influence on products.

See also

- | | |
|--|--|
| <ul style="list-style-type: none"> • CODASYL • Navigational database | <ul style="list-style-type: none"> • Semantic Web |
|--|--|

References

Further reading

- Charles W. Bachman, *The Programmer as Navigator*. ACM Turing Award lecture, Communications of the ACM, Volume 16, Issue 11, 1973, pp. 653-658, ISSN 0001-0782, doi:10.1145/355611.362534

External links

- CODASYL Systems Committee "Survey of Data Base Systems", 1968
- Network (CODASYL) Data Model

8. Network analysis (electrical circuits)

A network, in the context of electronics, is a collection of interconnected components. **Network analysis** is the process of finding the voltages across, and the currents through, every component in the network. There are a number of different techniques for achieving this. However, for the most part, they assume that the components of the network are all linear. The methods described in this article are only applicable to *linear* network analysis except where explicitly stated.

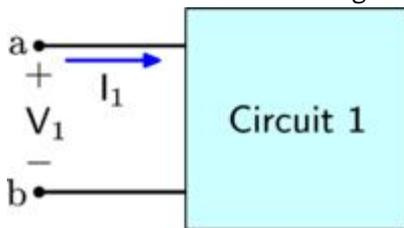
Definitions

- Component** A device with two or more terminals into which, or out of which, charge may flow.
- Node** A point at which terminals of more than two components are joined. A conductor with a substantially zero resistance is considered to be a node for the purpose of analysis.
- Branch** The component(s) joining two nodes.

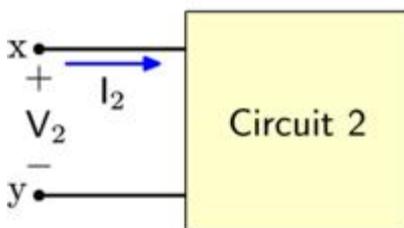
Mesh	A group of branches within a network joined so as to form a complete loop.
Port	Two terminals where the current into one is identical to the current out of the other.
Circuit	A current from one terminal of a generator, through load component(s) and back into the other terminal. A circuit is, in this sense, a one-port network and is a trivial case to analyse. If there is any connection to any other circuits then a non-trivial network has been formed and at least two ports must exist. Often, "circuit" and "network" are used interchangeably, but many analysts reserve "network" to mean an idealised model consisting of ideal components. ^[1]
Transfer function	The relationship of the currents and/or voltages between two ports. Most often, an input port and an output port are discussed and the transfer function is described as gain or attenuation.
Component transfer function	For a two-terminal component (i.e. one-port component), the current and voltage are taken as the input and output and the transfer function will have units of impedance or admittance (it is usually a matter of arbitrary convenience whether voltage or current is considered the input). A three (or more) terminal component effectively has two (or more) ports and the transfer function cannot be expressed as a single impedance. The usual approach is to express the transfer function as a matrix of parameters. These parameters can be impedances, but there is a large number of other approaches, see two-port network.

Equivalent circuits

A useful procedure in network analysis is to simplify the network by reducing the number of components. This can be done by replacing the actual components with other notional components that have the same effect. A particular technique might directly reduce the number of components, for instance by combining impedances in series. On the other hand it might merely change the form in to one in which the components can be reduced in a later operation. For instance, one might transform a voltage generator into a current generator using Norton's theorem in order to be able to later combine the internal resistance of the generator with a parallel impedance load.



A resistive circuit is a circuit containing only resistors, ideal current sources, and ideal voltage sources. If the sources are constant (DC) sources, the result is a DC circuit. The analysis of a circuit refers to the process of solving for the voltages and currents present in the circuit. The solution principles outlined here also apply to phasor analysis of AC circuits.



Two circuits are said to be **equivalent** with respect to a pair of terminals if the voltage across the terminals and current through the terminals for one network have the same relationship as the voltage and current at the terminals of the other network.

If $V_2 = V_1$ implies $I_2 = I_1$ for all (real) values of V_1 , then with respect to terminals ab and xy, circuit 1 and circuit 2 are equivalent.

If $V_2 = V_1$ implies $I_2 = I_1$ for all (real) values of V_1 , then with respect to terminals ab and xy, circuit 1 and circuit 2 are equivalent.

The above is a sufficient definition for a one-port network. For more than one port, then it must be defined that the currents and voltages between all pairs of corresponding ports must bear the same relationship. For instance, star and delta networks are effectively three port networks and hence require three simultaneous equations to fully specify their equivalence.

Impedances in series and in parallel

Any two terminal network of impedances can eventually be reduced to a single impedance by successive applications of impedances in series or impedances in parallel.

Impedances in series: $Z_{eq} = Z_1 + Z_2 + \dots + Z_n.$

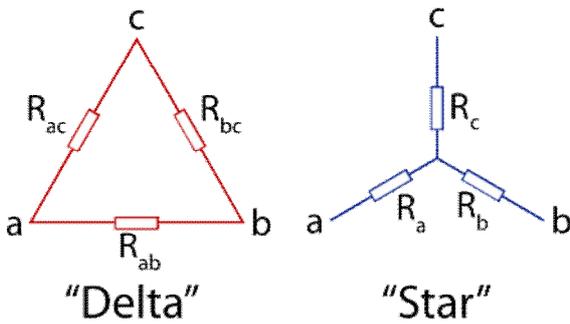
$$\frac{1}{Z_{eq}} = \frac{1}{Z_1} + \frac{1}{Z_2} + \dots + \frac{1}{Z_n}.$$

Impedances in parallel:

$$Z_{eq} = \frac{Z_1 Z_2}{Z_1 + Z_2}.$$

The above simplified for only two impedances in parallel:

Delta-wye transformation



A network of impedances with more than two terminals cannot be reduced to a single impedance equivalent circuit. An n-terminal network can, at best, be reduced to n impedances. For a three terminal network, the three impedances can be expressed as a three node delta (Δ) network or a four node star (Y) network. These two networks are equivalent and the transformations between them are given below. A general network with an arbitrary number of terminals cannot be reduced to the minimum number of impedances using only series and parallel combinations. In general, Y- Δ and Δ -Y transformations must also be used. It can

be shown that this is sufficient to find the minimal network for any arbitrary network with successive applications of series, parallel, Y- Δ and Δ -Y; no more complex transformations are required.

For equivalence, the impedances between any pair of terminals must be the same for both networks, resulting in a set of three simultaneous equations. The equations below are expressed as resistances but apply equally to the general case with impedances.

Delta-to-star transformation equations

$$R_a = \frac{R_{ac}R_{ab}}{R_{ac} + R_{ab} + R_{bc}}$$

$$R_b = \frac{R_{ab}R_{bc}}{R_{ac} + R_{ab} + R_{bc}}$$

$$R_c = \frac{R_{bc}R_{ac}}{R_{ac} + R_{ab} + R_{bc}}$$

Star-to-delta transformation equations

$$R_{ac} = \frac{R_a R_b + R_b R_c + R_c R_a}{R_b}$$

$$R_{ab} = \frac{R_a R_b + R_b R_c + R_c R_a}{R_c}$$

$$R_{bc} = \frac{R_a R_b + R_b R_c + R_c R_a}{R_a}$$

General form of network node elimination

The star-to-delta and series-resistor transformations are special cases of the general resistor network node elimination algorithm. Any node connected by N resistors ($R_1 \dots R_N$) to nodes 1 .. N can be replaced by $\binom{N}{2}$ resistors interconnecting the remaining N nodes. The resistance between any two nodes x and y is given by:

$$R_{xy} = R_x R_y \sum_{i=1}^N \frac{1}{R_i}$$

For a star-to-delta ($N = 3$) this reduces to:

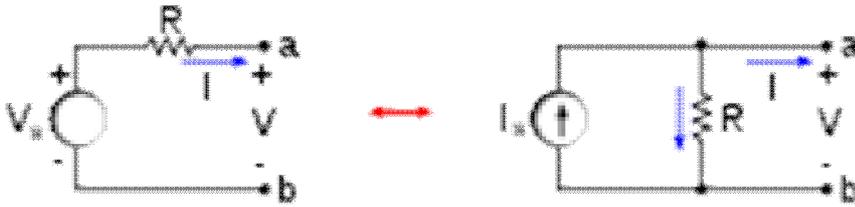
$$R_{ab} = R_a R_b \left(\frac{1}{R_a} + \frac{1}{R_b} + \frac{1}{R_c} \right) = \frac{R_a R_b (R_a R_b + R_a R_c + R_b R_c)}{R_a R_b R_c} = \frac{R_a R_b + R_b R_c + R_c R_a}{R_c}$$

For a series reduction ($N = 2$) this reduces to:

$$R_{ab} = R_a R_b \left(\frac{1}{R_a} + \frac{1}{R_b} \right) = \frac{R_a R_b (R_a + R_b)}{R_a R_b} = R_a + R_b$$

For a dangling resistor ($N = 1$) it results in the elimination of the resistor because $\binom{1}{2} = 0$.

Source transformation



☐ A generator with an internal impedance (ie non-ideal generator) can be represented as either an ideal voltage generator or an ideal current generator plus the impedance. These two forms are equivalent and the transformations are given below. If the two networks are equivalent with respect to terminals ab, then V and I must be identical for both networks. Thus,

$$V_s = RI_{\text{sor}} \quad I_s = \frac{V_s}{R}$$

- Norton's theorem states that any two-terminal network can be reduced to an ideal current generator and a parallel impedance.
- Thévenin's theorem states that any two-terminal network can be reduced to an ideal voltage generator plus a series impedance.

Simple networks

Some very simple networks can be analysed without the need to apply the more systematic approaches.

Voltage division of series components

Consider n impedances that are connected in **series**. The voltage V_i across any impedance Z_i is

$$V_i = Z_i I = \left(\frac{Z_i}{Z_1 + Z_2 + \dots + Z_n} \right) V$$

Current division of parallel components

Consider n impedances that are connected in **parallel**. The current I_i through any impedance Z_i is

$$I_i = \left(\frac{\left(\frac{1}{Z_i}\right)}{\left(\frac{1}{Z_1}\right) + \left(\frac{1}{Z_2}\right) + \dots + \left(\frac{1}{Z_n}\right)} \right) I$$

for $i = 1, 2, \dots, n$.

Special case: Current division of two parallel components

$$I_1 = \left(\frac{Z_2}{Z_1 + Z_2} \right) I$$

$$I_2 = \left(\frac{Z_1}{Z_1 + Z_2} \right) I$$

Nodal analysis

1. Label all **nodes** in the circuit. Arbitrarily select any node as reference.
2. Define a voltage variable from every remaining node to the reference. These voltage variables must be defined as voltage rises with respect to the reference node.
3. Write a KCL equation for every node except the reference.
4. Solve the resulting system of equations.

Mesh analysis

Mesh — a loop that does not contain an inner loop.

1. Count the number of “window panes” in the circuit. Assign a mesh current to each window pane.
2. Write a KVL equation for every mesh whose current is unknown.
3. Solve the resulting equations

Superposition

In this method, the effect of each generator in turn is calculated. All the generators other than the one being considered are removed; either short-circuited in the case of voltage generators, or open circuited in the case of current generators. The total current through, or the total voltage across, a particular branch is then calculated by summing all the individual currents or voltages.

There is an underlying assumption to this method that the total current or voltage is a linear superposition of its parts. The method cannot, therefore, be used if non-linear components are present. Note that mesh analysis and node analysis also implicitly use superposition so these too, are only applicable to linear circuits.

Choice of method

Choice of method^[2] is to some extent a matter of taste. If the network is particularly simple or only a specific current or voltage is required then ad-hoc application of some simple equivalent circuits may yield the answer without recourse to the more systematic methods.

- Superposition is possibly the most conceptually simple method but rapidly leads to a large number of equations and messy impedance combinations as the network becomes larger.
- Nodal analysis: The number of voltage variables, and hence simultaneous equations to solve, equals the number of nodes minus one. Every voltage source connected to the reference node reduces the number of unknowns (and equations) by one. Nodal analysis is thus best for voltage sources.
- Mesh analysis: The number of current variables, and hence simultaneous equations to solve, equals the number of meshes. Every current source in a mesh reduces the number of unknowns by one. Mesh analysis is thus best for current sources. Mesh analysis, however, cannot be used with networks which cannot be drawn as a planar network, that is, with no crossing components.^[3]

Transfer function

A transfer function expresses the relationship between an input and an output of a network. For resistive networks, this will always be a simple real number or an expression which boils down to a real number. Resistive networks are represented by a system of simultaneous algebraic equations. However in the general case of linear networks, the network is represented by a system of simultaneous linear differential equations. In network analysis, rather than use the differential equations directly, it is usual practice to carry out a Laplace transform on them first and then express the result in terms of the Laplace parameter s , which in general is complex. This is described as working in the s -domain. Working with the equations directly would be described as working in the time (or t) domain because the results would be expressed as time varying quantities. The Laplace transform is the mathematical method of transforming between the s -domain and the t -domain.

This approach is standard in control theory and is useful for determining stability of a system, for instance, in an amplifier with feedback.

Two terminal component transfer functions

For two terminal components the transfer function is the relationship between the current input to the device and the resulting voltage across it. The transfer function, $Z(s)$, will thus have units of impedance - ohms. For the three passive components found in electrical networks, the transfer functions are;

$$\text{Resistor } Z(s) = R$$

$$\text{Inductor } Z(s) = sL$$

$$\text{Capacitor } Z(s) = \frac{1}{sC}$$

For a network to which only steady ac signals are applied, s is replaced with $j\omega$ and the more familiar values from ac network theory result.

$$\text{Resistor } Z(j\omega) = R$$

$$\text{Inductor } Z(j\omega) = j\omega L$$

$$\text{Capacitor } Z(j\omega) = \frac{1}{j\omega C}$$

Finally, for a network to which only steady dc is applied, s is replaced with zero and dc network theory applies.

$$\text{Resistor } Z = R$$

$$\text{Inductor } Z = 0$$

$$\text{Capacitor } Z = \infty$$

Two port network transfer function

Transfer functions, in general, in control theory are given the symbol $H(s)$. Most commonly in electronics, transfer function is defined as the ratio of output voltage to input voltage and given the symbol $A(s)$, or more commonly (because analysis is invariably done in terms of sine wave response), $A(j\omega)$, so that;

$$A(j\omega) = \frac{V_o}{V_i}$$

The A standing for attenuation, or amplification, depending on context. In general, this will be a complex function of $j\omega$, which can be derived from an analysis of the impedances in the network and their individual transfer functions. Sometimes the analyst is only interested in the magnitude of the gain and not the phase angle. In this case the complex numbers can be eliminated from the transfer function and it might then be written as;

$$A(\omega) = \left| \frac{V_o}{V_i} \right|$$

Two port parameters

The concept of a two-port network can be useful in network analysis as a black box approach to analysis. The behaviour of the two-port network in a larger network can be entirely characterised without necessarily stating anything about the internal structure. However, to do this it is necessary to have more information than just the $A(j\omega)$ described above. It can be shown that four such parameters are required to fully characterise the two-port network. These could be the forward transfer function, the input impedance, the reverse transfer function (ie, the voltage appearing at the input when a voltage is applied to the output) and the output impedance. There are many others (see the main article for a full listing), one of these expresses all four parameters as impedances. It is usual to express the four parameters as a matrix;

$$\begin{bmatrix} V_1 \\ V_0 \end{bmatrix} = \begin{bmatrix} z(j\omega)_{11} & z(j\omega)_{12} \\ z(j\omega)_{21} & z(j\omega)_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_0 \end{bmatrix}$$

The matrix may be abbreviated to a representative element;

$$[z(j\omega)]_{\text{or just } [z]}$$

These concepts are capable of being extended to networks of more than two ports. However, this is rarely done in reality as in many practical cases ports are considered either purely input or purely output. If reverse direction transfer functions are ignored, a multi-port network can always be decomposed into a number of two-port networks.

Distributed components

Where a network is composed of discrete components, analysis using two-port networks is a matter of choice, not essential. The network can always alternatively be analysed in terms of its individual component transfer functions. However, if a network contains distributed components, such as in the case of a transmission line, then it is not possible to analyse in terms of individual components since they do not exist. The most common approach to this is to model the line as a two-port network and characterise it using two-port parameters (or something equivalent to them). Another example of this technique is modelling the carriers crossing the base region in a high frequency transistor. The base region has to be modelled as distributed resistance and capacitance rather than lumped components.

Image analysis

Transmission lines and certain types of filter design use the image method to determine their transfer parameters. In this method, the behaviour of an infinitely long cascade connected chain of identical networks is considered. The input and output impedances and the forward and reverse transmission functions are then calculated for this infinitely long chain. Although, the theoretical values so obtained can never be exactly realised in practice, in many cases they serve as a very good approximation for the behaviour of a finite chain as long as it is not too short.

Non-linear networks

Most electronic designs are, in reality, non-linear. There is very little that does not include some semiconductor devices. These are invariably non-linear, the transfer function of an ideal semiconductor pn junction is given by the very non-linear relationship;

$$i = I_o(e^{\frac{v}{V_T}} - 1)$$

where;

- i and v are the instantaneous current and voltage.
- I_o is an arbitrary parameter called the reverse leakage current whose value depends on the construction of the device.
- V_T is a parameter proportional to temperature called the thermal voltage and equal to about 25mV at room temperature.

There are many other ways that non-linearity can appear in a network. All methods utilising linear superposition will fail when non-linear components are present. There are several options for dealing with non-linearity depending on the type of circuit and the information the analyst wishes to obtain.

Boolean analysis of switching networks

A switching device is one where the non-linearity is utilised to produce two opposite states. CMOS devices in digital circuits, for instance, have their output connected to either the positive or the negative supply rail and are never found at anything in between except during a transient period when the device is actually switching. Here the non-linearity is designed to be extreme, and the analyst can actually take advantage of that fact. These kinds of networks can be analysed using Boolean algebra by assigning the two states ("on"/"off", "positive"/"negative" or whatever states are being used) to the boolean constants "0" and "1".

The transients are ignored in this analysis, along with any slight discrepancy between the actual state of the device and the nominal state assigned to a boolean value. For instance, boolean "1" may be assigned to the state of +5V. The output of the device may actually be +4.5V but the analyst still considers this to be boolean "1". Device manufacturers will usually specify a range of values in their data sheets that are to be considered undefined (ie the result will be unpredictable).

The transients are not entirely uninteresting to the analyst. The maximum rate of switching is determined by the speed of transition from one state to the other. Happily for the analyst, for many devices most of the transition occurs in the linear portion of the devices transfer function and linear analysis can be applied to obtain at least an approximate answer.

It is mathematically possible to derive boolean algebras which have more than two states. There is not too much use found for these in electronics, although three-state devices are passingly common.

Separation of bias and signal analyses

This technique is used where the operation of the circuit is to be essentially linear, but the devices used to implement it are non-linear. A transistor amplifier is an example of this kind of network. The essence of this technique is to separate the analysis in to two parts. Firstly, the dc biases are analysed using some non-linear method. This establishes the quiescent operating point of the circuit. Secondly, the small signal characteristics of the circuit are analysed using linear network analysis. Examples of methods that can be used for both these stages are given below.

Graphical method of dc analysis

In a great many circuit designs, the dc bias is fed to a non-linear component via a resistor (or possibly a network of resistors). Since resistors are linear components, it is particularly easy to determine the quiescent operating point of the non-linear device from a graph of its transfer function. The method is as follows: from linear network analysis the output transfer function (that is output voltage against output current) is calculated for the network of resistor(s) and the generator driving them. This will be a straight line and can readily be superimposed on the

transfer function plot of the non-linear device. The point where the lines cross is the quiescent operating point.

Perhaps the easiest practical method is to calculate the (linear) network open circuit voltage and short circuit current and plot these on the transfer function of the non-linear device. The straight line joining these two point is the transfer function of the network.

In reality, the designer of the circuit would proceed in the reverse direction to that described. Starting from a plot provided in the manufacturers data sheet for the non-linear device, the designer would choose the desired operating point and then calculate the linear component values required to achieve it.

It is still possible to use this method if the device being biased has its bias fed through another device which is itself non-linear - a diode for instance. In this case however, the plot of the network transfer function onto the device being biased would no longer be a straight line and is consequently more tedious to do.

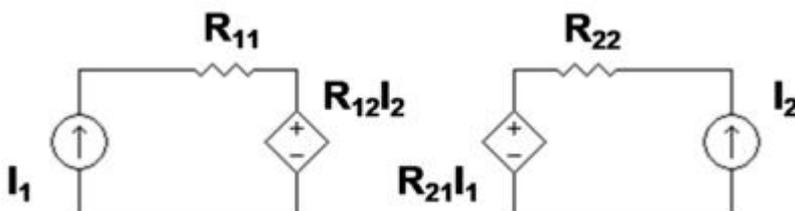
Small signal equivalent circuit

This method can be used where the deviation of the input and output signals in a network stay within a substantially linear portion of the non-linear devices transfer function, or else are so small that the curve of the transfer function can be considered linear. Under a set of these specific conditions, the non-linear device can be represented by an equivalent linear network. It must be remembered that this equivalent circuit is entirely notional and only valid for the small signal deviations. It is entirely inapplicable to the dc biasing of the device.

For a simple two-terminal device, the small signal equivalent circuit may be no more than two components. A resistance equal to the slope of the v/i curve at the operating point (called the dynamic resistance), and tangent to the curve. A generator, because this tangent will not, in general, pass through the origin. With more terminals, more complicated equivalent circuits are required.

A popular form of specifying the small signal equivalent circuit amongst transistor manufacturers is to use the two-port network parameters known as $[h]$ parameters. These are a matrix of four parameters as with the $[z]$ parameters but in the case of the $[h]$ parameters they are a hybrid mixture of impedances, admittances, current gains and voltage gains. In this model the three terminal transistor is considered to be a two port network, one of its terminals being common to both ports. The $[h]$ parameters are quite different depending on which terminal is chosen as the common one. The most important parameter for transistors is usually the forward current gain, h_{21} , in the common emitter configuration. This is designated h_{fe} on data sheets.

The small signal equivalent circuit in terms of two-port parameters leads to the concept of dependent generators. That is, the value of a voltage or current generator depends linearly on a voltage or current elsewhere in the circuit. For instance the $[z]$ parameter model leads to dependent voltage generators as shown in this diagram;



 $[z]$ parameter equivalent circuit showing dependent voltage generators

There will always be dependent generators in a two-port parameter equivalent circuit. This applies to the $[h]$ parameters as well as to the $[z]$ and any other kind. These dependencies must be preserved when developing the equations in a larger linear network analysis.

Piecewise linear method

In this method, the transfer function of the non-linear device is broken up into regions. Each of these regions is approximated by a straight line. Thus, the transfer function will be linear up to a particular point where there will be a discontinuity. Past this point the transfer function will again be linear but with a different slope.

A well known application of this method is the approximation of the transfer function of a pn junction diode. The actual transfer function of an ideal diode has been given at the top of this (non-linear) section. However, this formula is rarely used in network analysis, a piecewise approximation being used instead. It can be seen that the diode current rapidly diminishes to $-I_o$ as the voltage falls. This current, for most purposes, is so small it can be ignored. With increasing voltage, the current increases exponentially. The diode is modelled as an open circuit up to the knee of the exponential curve, then past this point as a resistor equal to the bulk resistance of the semiconducting material.

The commonly accepted values for the transition point voltage are 0.7V for silicon devices and 0.3V for germanium

devices. An even simpler model of the diode, sometimes used in switching applications, is short circuit for forward voltages and open circuit for reverse voltages.

The model of a forward biased pn junction having an approximately constant 0.7V is also a much used approximation for transistor base-emitter junction voltage in amplifier design.

The piecewise method is similar to the small signal method in that linear network analysis techniques can only be applied if the signal stays within certain bounds. If the signal crosses a discontinuity point then the model is no longer valid for linear analysis purposes. The model does have the advantage over small signal however, in that it is equally applicable to signal and dc bias. These can therefore both be analysed in the same operations and will be linearly superimposable.

Time-varying components

In linear analysis, the components of the network are assumed to be unchanging, but in some circuits this does not apply, such as sweep oscillators, voltage controlled amplifiers, and variable equalisers. In many circumstances the change in component value is periodic. A non-linear component excited with a periodic signal, for instance, can be represented as periodically varying *linear* component. Sidney Darlington disclosed a method of analysing such periodic time varying circuits. He developed canonical circuit forms which are analogous to the canonical forms of Ronald Foster and Wilhelm Cauer used for analysing linear circuits.^[4]

See also

- | | |
|--|---|
| <ul style="list-style-type: none"> • Bartlett's bisection theorem • Circuit theory • Equivalent impedance transforms • Kirchhoff's circuit laws • Mesh analysis • Millman's Theorem • Ohm's law | <ul style="list-style-type: none"> • Reciprocity theorem • Resistive circuit • Series and parallel circuits • Tellegen's theorem • Two-port network • Wye-delta transform |
|--|---|

References

1. ^ Belevitch V (May 1962). "Summary of the history of circuit theory". *Proceedings of the IRE* **50** (5): 849. doi:10.1109/JRPROC.1962.288301. cites "IRE Standards on Circuits: Definitions of Terms for Linear Passive Reciprocal Time Invariant Networks, 1960". *Proceedings of the IRE* **48** (9): 1609. September 1960. doi:10.1109/JRPROC.1960.287676.to justify this definition. Sidney Darlington Darlington S (1984). "A history of network synthesis and filter theory for circuits composed of resistors, inductors, and capacitors". *IEEE Trans. Circuits and Systems* **31** (1): 4. follows Belevitch but notes there are now also many colloquial uses of "network".
2. ^ Nilsson, J W, Riedel, S A (2007). *Electric Circuits* (8th ed). Pearson Prentice Hall. p. 112–113. ISBN 0-13-198925-1. http://books.google.com/books?id=sxmM8RFL99wC&lpg=PA200&dq=isbn%3A0131989251&lr=&as_drrb_is=q&as_minm_is=0&as_miny_is=&as_maxm_is=0&as_maxy_is=&as_brr=0&pg=PA112#v=onepage&q=112&f=false.
3. ^ Nilsson, J W, Riedel, S A (2007). *Electric Circuits* (8th ed.). Pearson Prentice Hall. p. 94. ISBN 0-13-198925-1. http://books.google.com/books?id=sxmM8RFL99wC&lpg=PA200&dq=isbn%3A0131989251&lr=&as_drrb_is=q&as_minm_is=0&as_miny_is=&as_maxm_is=0&as_maxy_is=&as_brr=0&pg=PA94#v=onepage&q=&f=false.
4. ^ US patent 3265973, Sidney Darlington, Irwin W. Sandberg, "Synthesis of two-port networks having periodically time-varying elements", granted 1966-08-09

External links

- | | |
|--|---|
| <ul style="list-style-type: none"> • Circuit Analysis Techniques • Nodal Analysis of Op Amp Circuits • Analysis of Resistive Circuits | <ul style="list-style-type: none"> • Circuit Analysis Related Laws, Examples and Solutions |
|--|---|

9. Social network

A **social network** is a social structure made of individuals (or organizations) called "nodes," which are tied (connected) by one or more specific types of interdependency, such as friendship, kinship, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige.

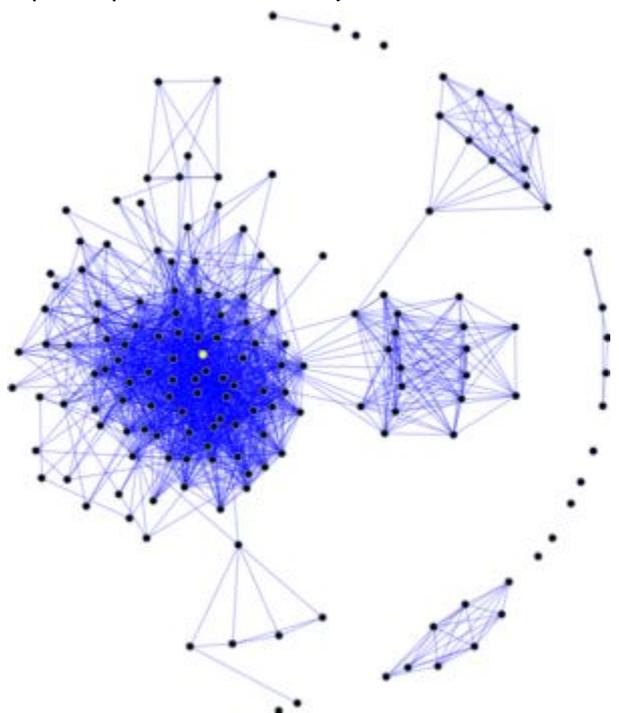
Social network analysis views social relationships in terms of network theory consisting of *nodes* and *ties*. Nodes are the individual actors within the networks, and ties are the relationships between the actors. The resulting

graph-based structures are often very complex. There can be many kinds of ties between the nodes. Research in a number of academic fields has shown that social networks operate on many levels, from families up to the level of nations, and play a critical role in determining the way problems are solved, organizations are run, and the degree to which individuals succeed in achieving their goals.

In its simplest form, a social network is a map of all of the relevant ties between all the nodes being studied. The network can also be used to measure social capital – the value that an individual gets from the social network. These concepts are often displayed in a social network diagram, where nodes are the points and ties are the lines.

Social network analysis

Social network analysis (related to *network theory*) has emerged as a key technique in modern sociology. It has also gained a significant following in anthropology, biology, communication studies, economics, geography, information science, organizational studies, social psychology, and sociolinguistics, and has become a popular topic of speculation and study.



☞ An example of a social network diagram. The node with the highest betweenness centrality is marked in yellow.

People have used the idea of "**social network**" loosely for over a century to connote complex sets of relationships between members of social systems at all scales, from interpersonal to international. In 1954, J. A. Barnes started using the term systematically to denote patterns of ties, encompassing concepts traditionally used by the public and those used by social scientists: bounded groups (e.g., tribes, families) and social categories (e.g., gender, ethnicity). Scholars such as S.D. Berkowitz, Stephen Borgatti, Ronald Burt, Kathleen Carley, Martin Everett, Katherine Faust, Linton Freeman, Mark Granovetter, David Knoke, David Krackhardt, Peter Marsden, Nicholas Mullins, Anatol Rapoport, Stanley Wasserman, Barry Wellman, Douglas R. White, and Harrison White expanded the use of systematic social network analysis.^[1]

Social network analysis has now moved from being a suggestive metaphor to an analytic approach to a paradigm, with its own theoretical statements, methods, social network analysis software, and researchers.

Analysts reason from whole to part; from structure to relation to individual; from behavior to attitude. They typically either study *whole networks* (also known as *complete networks*), all of the ties containing specified relations in a defined population, or *personal networks* (also known as *egocentric networks*), the ties that specified people have, such as their "personal communities".^[2] The distinction between whole/complete networks and personal/egocentric networks has depended largely on how analysts were able to gather data. That is, for groups such as companies, schools, or membership societies, the analyst was expected to have complete information about who was in the network, all participants being both potential egos and alters. Personal/egocentric studies were typically conducted when identities of egos were known, but not their alters. These studies rely on the egos to provide information about the identities of alters and there is no expectation that the various egos or sets of alters will be tied to each other. A *snowball network* refers to the idea that the alters identified in an egocentric survey then become egos themselves and are able in turn to nominate additional alters. While there are severe logistic limits to conducting snowball network studies, a method for examining *hybrid networks* has recently been developed in which egos in complete networks can nominate alters otherwise not listed who are then available for all subsequent egos to see.^[3] The hybrid network may be valuable for examining whole/complete networks that are expected to include important players beyond those who are formally identified. For example, employees of a company often work with non-company consultants who may be part of a network that cannot fully be defined prior to data collection.

Several analytic tendencies distinguish social network analysis:^[4]

There is no assumption that groups are the building blocks of society: the approach is open to studying less-bounded social systems, from nonlocal communities to links among websites.

Rather than treating individuals (persons, organizations, states) as discrete units of analysis, it focuses on how the structure of ties affects individuals and their relationships.

In contrast to analyses that assume that socialization into norms determines behavior, network analysis looks to see the extent to which the structure and composition of ties affect norms.

The shape of a social network helps determine a network's usefulness to its individuals. Smaller, tighter networks can be less useful to their members than networks with lots of loose connections (weak ties) to individuals outside the main network. More open networks, with many weak ties and social connections, are more likely to introduce new ideas and opportunities to their members than closed networks with many redundant ties. In other words, a group of friends who only do things with each other already share the same knowledge and opportunities. A group of individuals with connections to other social worlds is likely to have access to a wider range of information. It is better for individual success to have connections to a variety of networks rather than many connections within a single network. Similarly, individuals can exercise influence or act as brokers within their social networks by bridging two networks that are not directly linked (called filling structural holes).^[5]

The power of social network analysis stems from its difference from traditional social scientific studies, which assume that it is the attributes of individual actors—whether they are friendly or unfriendly, smart or dumb, etc.—that matter. Social network analysis produces an alternate view, where the attributes of individuals are less important than their relationships and ties with other actors within the network. This approach has turned out to be useful for explaining many real-world phenomena, but leaves less room for individual agency, the ability for individuals to influence their success, because so much of it rests within the structure of their network.

Social networks have also been used to examine how organizations interact with each other, characterizing the many informal connections that link executives together, as well as associations and connections between individual employees at different organizations. For example, power within organizations often comes more from the degree to which an individual within a network is at the center of many relationships than actual job title. Social networks also play a key role in hiring, in business success, and in job performance. Networks provide ways for companies to gather information, deter competition, and collude in setting prices or policies.^[6]

History of social network analysis

A summary of the progress of social networks and social network analysis has been written by Linton Freeman.^[7]

Precursors of social networks in the late 1800s include Émile Durkheim and Ferdinand Tönnies. Tönnies argued that social groups can exist as personal and direct social ties that either link individuals who share values and belief (*gemeinschaft*) or impersonal, formal, and instrumental social links (*gesellschaft*). Durkheim gave a non-individualistic explanation of social facts arguing that social phenomena arise when interacting individuals constitute a reality that can no longer be accounted for in terms of the properties of individual actors. He distinguished between a traditional society – "mechanical solidarity" – which prevails if individual differences are minimized, and the modern society – "organic solidarity" – that develops out of cooperation between differentiated individuals with independent roles.

Georg Simmel, writing at the turn of the twentieth century, was the first scholar to think directly in social network terms. His essays pointed to the nature of network size on interaction and to the likelihood of interaction in ramified, loosely-knit networks rather than groups (Simmel, 1908/1971 1908/1971).

After a hiatus in the first decades of the twentieth century, three main traditions in social networks appeared. In the 1930s, J.L. Moreno pioneered the systematic recording and analysis of social interaction in small groups, especially classrooms and work groups (sociometry), while a Harvard group led by W. Lloyd Warner and Elton Mayo explored interpersonal relations at work. In 1940, A.R. Radcliffe-Brown's presidential address to British anthropologists urged the systematic study of networks.^[8] However, it took about 15 years before this call was followed-up systematically.

Social network analysis developed with the kinship studies of Elizabeth Bott in England in the 1950s and the 1950s-1960s urbanization studies of the University of Manchester group of anthropologists (centered around Max Gluckman and later J. Clyde Mitchell) investigating community networks in southern Africa, India and the United Kingdom. Concomitantly, British anthropologist S.F. Nadel codified a theory of social structure that was influential in later network analysis.^[9]

In the 1960s-1970s, a growing number of scholars worked to combine the different tracks and traditions. One large group was centered around Harrison White and his students at Harvard University: Ivan Chase, Bonnie Erickson, Harriet Friedmann, Mark Granovetter, Nancy Howell, Joel Levine, Nicholas Mullins, John Padgett, Michael Schwartz and Barry Wellman. Also important in this early group were Charles Tilly, who focused on networks in political sociology and social movements, and Stanley Milgram, who developed the "six degrees of separation" thesis.^[10] White's group thought of themselves as rebelling against the reigning structural-functionalist orthodoxy of then-dominant Harvard sociologist Talcott Parsons, leading them to devalue concerns with symbols, values, norms and culture. They also were opposed to the methodological individualism espoused by another Harvard

sociologist, George Homans, which was endemic among the dominant survey researchers and positivists of the time. Mark Granovetter and Barry Wellman are among the former students of White who have elaborated and popularized social network analysis.^[11]

White's was not the only group. Significant independent work was done by scholars elsewhere: University of California Irvine social scientists interested in mathematical applications, centered around Linton Freeman, including John Boyd, Susan Freeman, Kathryn Faust, A. Kimball Romney and Douglas White; quantitative analysts at the University of Chicago, including Joseph Galaskiewicz, Wendy Griswold, Edward Laumann, Peter Marsden, Martina Morris, and John Padgett; and communication scholars at Michigan State University, including Nan Lin and Everett Rogers. A substantively-oriented University of Toronto sociology group developed in the 1970s, centered on former students of Harrison White: S.D. Berkowitz, Harriet Friedmann, Nancy Leslie Howard, Nancy Howell, Lorne Tepperman and Barry Wellman, and also including noted modeler and game theorist Anatol Rapoport.^[12]

Research

Social network analysis has been used in epidemiology to help understand how patterns of human contact aid or inhibit the spread of diseases such as HIV in a population. The evolution of social networks can sometimes be modeled by the use of agent based models, providing insight into the interplay between communication rules, rumor spreading and social structure.

SNA may also be an effective tool for mass surveillance -- for example the Total Information Awareness program was doing in-depth research on strategies to analyze social networks to determine whether or not U.S. citizens were political threats.

Diffusion of innovations theory explores social networks and their role in influencing the spread of new ideas and practices. Change agents and opinion leaders often play major roles in spurring the adoption of innovations, although factors inherent to the innovations also play a role.

Robin Dunbar has suggested that the typical size of a egocentric network is constrained to about 150 members due to possible limits in the capacity of the human communication channel. The rule arises from cross-cultural studies in sociology and especially anthropology of the maximum size of a village (in modern parlance most reasonably understood as an *ecovillage*). It is theorized in evolutionary psychology that the number may be some kind of limit of average human ability to recognize members and track emotional facts about all members of a group. However, it may be due to economics and the need to track "free riders", as it may be easier in larger groups to take advantage of the benefits of living in a community without contributing to those benefits.

Mark Granovetter found in one study that more numerous weak ties can be important in seeking information and innovation. Cliques have a tendency to have more homogeneous opinions as well as share many common traits. This homophilic tendency was the reason for the members of the cliques to be attracted together in the first place. However, being similar, each member of the clique would also know more or less what the other members knew. To find new information or insights, members of the clique will have to look beyond the clique to its other friends and acquaintances. This is what Granovetter called the "the strength of weak ties".

Guanxi is a central concept in Chinese society (and other East Asian cultures) that can be summarized as the use of personal influence. Guanxi can be studied from a social network approach.^[13]

The small world phenomenon is the hypothesis that the chain of social acquaintances required to connect one arbitrary person to another arbitrary person anywhere in the world is generally short. The concept gave rise to the famous phrase six degrees of separation after a 1967 *small world experiment* by psychologist Stanley Milgram. In Milgram's experiment, a sample of US individuals were asked to reach a particular target person by passing a message along a chain of acquaintances. The average length of successful chains turned out to be about five intermediaries or six separation steps (the majority of chains in that study actually failed to complete). The methods (and ethics as well) of Milgram's experiment was later questioned by an American scholar, and some further research to replicate Milgram's findings had found that the degrees of connection needed could be higher.^[14] Academic researchers continue to explore this phenomenon as Internet-based communication technology has supplemented the phone and postal systems available during the times of Milgram. A recent electronic small world experiment at Columbia University found that about five to seven degrees of separation are sufficient for connecting any two people through e-mail.^[15]

Collaboration graphs can be used to illustrate good and bad relationships between humans. A positive edge between two nodes denotes a positive relationship (friendship, alliance, dating) and a negative edge between two nodes denotes a negative relationship (hatred, anger). Signed social network graphs can be used to predict the future evolution of the graph. In signed social networks, there is the concept of "balanced" and "unbalanced" cycles. A balanced cycle is defined as a cycle where the product of all the signs are positive. Balanced graphs

represent a group of people who are unlikely to change their opinions of the other people in the group. Unbalanced graphs represent a group of people who are very likely to change their opinions of the people in their group. For example, a group of 3 people (A, B, and C) where A and B have a positive relationship, B and C have a positive relationship, but C and A have a negative relationship is an unbalanced cycle. This group is very likely to morph into a balanced cycle, such as one where B only has a good relationship with A, and both A and B have a negative relationship with C. By using the concept of balances and unbalanced cycles, the evolution of signed social network graphs can be predicted.

One study has found that happiness tends to be correlated in social networks. When a person is happy, nearby friends have a 25 percent higher chance of being happy themselves. Furthermore, people at the center of a social network tend to become happier in the future than those at the periphery. Clusters of happy and unhappy people were discerned within the studied networks, with a reach of three degrees of separation: a person's happiness was associated with the level of happiness of their friends' friends' friends.^[16]

Some researchers have suggested that human social networks may have a genetic basis.^[17] Using a sample of twins from the National Longitudinal Study of Adolescent Health, they found that in-degree (the number of times a person is named as a friend), transitivity (the probability that two friends are friends with one another), and betweenness centrality (the number of paths in the network that pass through a given person) are all significantly heritable. Existing models of network formation cannot account for this intrinsic node variation, so the researchers propose an alternative "Attract and Introduce" model that can explain heritability and many other features of human social networks.^[18]

Application to Environmental Issues

The 1984 book *The IRG Solution* argued that central media and government-type hierarchical organizations could not adequately understand the environmental crisis we were manufacturing, or how to initiate adequate solutions. It argued that the widespread introduction of Information Routing Groups was required to create a social network whose overall intelligence could collectively understand the issues and devise and implement correct workable solutions and policies.

Metrics (Measures) in social network analysis

Betweenness

The extent to which a node lies between other nodes in the network. This measure takes into account the connectivity of the node's neighbors, giving a higher value for nodes which bridge clusters. The measure reflects the number of people who a person is connecting indirectly through their direct links.^[19]

Bridge

An edge is said to be a bridge if deleting it would cause its endpoints to lie in different components of a graph.

Centrality

This measure gives a rough indication of the social power of a node based on how well they "connect" the network. "Betweenness", "Closeness", and "Degree" are all measures of centrality.

Centralization

The difference between the number of links for each node divided by maximum possible sum of differences. A centralized network will have many of its links dispersed around one or a few nodes, while a decentralized network is one in which there is little variation between the number of links each node possesses.

Closeness

The degree an individual is near all other individuals in a network (directly or indirectly). It reflects the ability to access information through the "grapevine" of network members. Thus, closeness is the inverse of the sum of the shortest distances between each individual and every other person in the network. (See also: Proxemics) The shortest path may also be known as the "geodesic distance".

Clustering coefficient

A measure of the likelihood that two associates of a node are associates themselves. A higher clustering coefficient indicates a greater 'cliquishness'.

Cohesion

The degree to which actors are connected directly to each other by cohesive bonds. Groups are identified as 'cliques' if every individual is directly tied to every other individual, 'social circles' if there is less stringency of direct contact, which is imprecise, or as structurally cohesive blocks if precision is wanted.^[20]

Degree

The count of the number of ties to other actors in the network. See also degree (graph theory).

(Individual-level) Density

The degree a respondent's ties know one another/ proportion of ties among an individual's nominees. Network or global-level density is the proportion of ties in a network relative to the total number possible (sparse versus

dense networks).

Flow betweenness centrality

The degree that a node contributes to sum of maximum flow between all pairs of nodes (not that node).

Eigenvector centrality

A measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to nodes having a high score contribute more to the score of the node in question.

Local Bridge

An edge is a local bridge if its endpoints share no common neighbors. Unlike a bridge, a local bridge is contained in a cycle.

Path Length

The distances between pairs of nodes in the network. Average path-length is the average of these distances between all pairs of nodes.

Prestige

In a directed graph prestige is the term used to describe a node's centrality. "Degree Prestige", "Proximity Prestige", and "Status Prestige" are all measures of Prestige. See also degree (graph theory).

Radiality

Degree an individual's network reaches out into the network and provides novel information and influence.

Reach

The degree any member of a network can reach other members of the network.

Structural cohesion

The minimum number of members who, if removed from a group, would disconnect the group.^[21]

Structural equivalence

Refers to the extent to which nodes have a common set of linkages to other nodes in the system. The nodes don't need to have any ties to each other to be structurally equivalent.

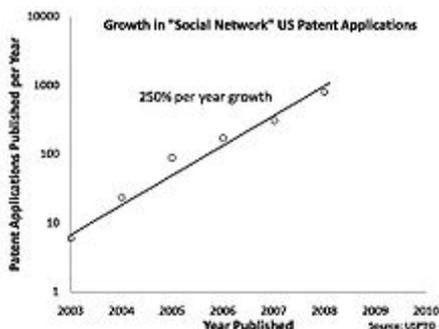
Structural hole

Static holes that can be strategically filled by connecting one or more links to link together other points. Linked to ideas of social capital: if you link to two people who are not linked you can control their communication.

Network analytic software

Network analytic tools are used to represent the nodes (agents) and edges (relationships) in a network, and to analyze the network data. Like other software tools, the data can be saved in external files. Additional information comparing the various data input formats used by network analysis software packages is available at NetWiki. Network analysis tools allow researchers to investigate large networks like the Internet, disease transmission, etc. These tools provide mathematical functions that can be applied to the network model.

Visual representation of social networks is important to understand the network data and convey the result of the analysis [2]. Network analysis tools are used to change the layout, colors, size and advanced properties of the network representation.



Patents

There has been rapid growth in the number of US patent applications that cover new technologies related to social networking. The number of published applications has been growing at about 250% per year over the past five years. There are now over 2000 published applications.^[22] Only about 100 of these applications have issued as patents, however, largely due to the multi-year backlog in examination of business method patents and ethical issues connected with this patent category^[23]

See also

- Clique
- Community of practice
- Dynamic network analysis
- Digital footprint
- FOAF (software) (Friend of a friend)
- Friendship paradox
- Knowledge management
- Social networking service
- Social network aggregation
- Social network analysis software
- Social software
- Social unit
- Social web
- SocialRank

- List of social networking websites
- Mathematical sociology
- Metcalfe's Law
- Network analysis
- Network of practice
- Network science
- Organizational patterns
- Small world phenomenon
- Social-circles network model
- Socio-technical systems
- Surveillance
- Triadic closure
- Value network
- Virtual community
- Virtual organization
- Weighted network

References

1. ^ Linton Freeman, *The Development of Social Network Analysis*. Vancouver: Empirical Press, 2006.
2. ^ Wellman, Barry and S.D. Berkowitz, eds., 1988. *Social Structures: A Network Approach*. Cambridge: Cambridge University Press.
3. ^ Hansen, William B. and Reese, Eric L. 2009. *Network Genie User Manual*. Greensboro, NC: Tanglewood Research.
4. ^ Freeman, Linton. 2006. *The Development of Social Network Analysis*. Vancouver: Empirical Pres, 2006; Wellman, Barry and S.D. Berkowitz, eds., 1988. *Social Structures: A Network Approach*. Cambridge: Cambridge University Press.
5. ^ Scott, John. 1991. *Social Network Analysis*. London: Sage.
6. ^ Wasserman, Stanley, and Faust, Katherine. 1994. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press.
7. ^ *The Development of Social Network Analysis* Vancouver: Empirical Press.
8. ^ A.R. Radcliffe-Brown, "On Social Structure," *Journal of the Royal Anthropological Institute*: 70 (1940): 1-12.
9. ^ [Nadel, SF. 1957. *The Theory of Social Structure*. London: Cohen and West.
10. ^ The Networked Individual: A Profile of Barry Wellman. [1]
11. ^ Mark Granovetter, "Introduction for the French Reader," *Sociologica* 2 (2007): 1-8; Wellman, Barry. 1988. "Structural Analysis: From Method and Metaphor to Theory and Substance." Pp. 19-61 in *Social Structures: A Network Approach*, edited by Barry Wellman and S.D. Berkowitz. Cambridge: Cambridge University Press.
12. ^ Mark Granovetter, "Introduction for the French Reader," *Sociologica* 2 (2007): 1-8; Wellman, Barry. 1988. "Structural Analysis: From Method and Metaphor to Theory and Substance." Pp. 19-61 in *Social Structures: A Network Approach*, edited by Barry Wellman and S.D. Berkowitz. Cambridge: Cambridge University Press. (see also Scott, 2000 and Freeman, 2004).
13. ^ Barry Wellman, Wenhong Chen and Dong Weizhen. "Networking Guanxi." Pp. 221-41 in *Social Connections in China: Institutions, Culture and the Changing Nature of Guanxi*, edited by Thomas Gold, Douglas Guthrie and David Wank. Cambridge University Press, 2002.
14. ^ *Could It Be A Big World After All?:* Judith Kleinfeld article.
15. ^ Six Degrees: The Science of a Connected Age, Duncan Watts.
16. ^ James H. Fowler and Nicholas A. Christakis. 2008. "Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study." *British Medical Journal*. December 4, 2008: doi:10.1136/bmj.a2338. Media account for those who cannot retrieve the original: Happiness: It Really is Contagious Retrieved December 5, 2008.
17. ^ "Genes and the Friends You Make". *Wall Street Journal*. January 27, 2009. <http://online.wsj.com/article/SB123302040874118079.html>.
18. ^ Fowler, J. H. (10 February 2009). "Model of Genetic Variation in Human Social Networks" (PDF). *Proceedings of the National Academy of Sciences* **106** (6): 1720–1724. doi:10.1073/pnas.0806746106. http://jhfwler.ucsd.edu/genes_and_social_networks.pdf.
19. ^ The most comprehensive reference is: Wasserman, Stanley, & Faust, Katherine. (1994). *Social Networks Analysis: Methods and Applications*. Cambridge: Cambridge University Press. A short, clear basic summary is in Krebs, Valdis. (2000). "The Social Life of Routers." *Internet Protocol Journal*, 3 (December): 14-25.
20. ^ Cohesive.blocking is the R program for computing structural cohesion according to the Moody-White (2003) algorithm. This wiki site provides numerous examples and a tutorial for use with R.
21. ^ Moody, James, and Douglas R. White (2003). "Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups." *American Sociological Review* 68(1):103-127. Online: (PDF file.
22. ^ USPTO search on published patent applications mentioning "social network"

23. ^ USPTO search on issued patents mentioning “social network”

Further reading

- Barnes, J. A. "Class and Committees in a Norwegian Island Parish", *Human Relations* 7:39-58
- Berkowitz, Stephen D. 1982. *An Introduction to Structural Analysis: The Network Approach to Social Research*. Toronto: Butterworth. ISBN 0409813621
- Brandes, Ulrik, and Thomas Erlebach (Eds.). 2005. *Network Analysis: Methodological Foundations* Berlin, Heidelberg: Springer-Verlag.
- Breiger, Ronald L. 2004. "The Analysis of Social Networks." Pp. 505–526 in *Handbook of Data Analysis*, edited by Melissa Hardy and Alan Bryman. London: Sage Publications. ISBN 0761966528 *Excerpts in pdf format*
- Burt, Ronald S. (1992). *Structural Holes: The Structure of Competition*. Cambridge, MA: Harvard University Press. ISBN 067484372X
- **(Italian)** Casaleggio, Davide (2008). *TU SEI RETE. La Rivoluzione del business, del marketing e della politica attraverso le reti sociali*. ISBN 8890182652
- Carrington, Peter J., John Scott and Stanley Wasserman (Eds.). 2005. *Models and Methods in Social Network Analysis*. New York: Cambridge University Press. ISBN 9780521809597
- Christakis, Nicholas and James H. Fowler "The Spread of Obesity in a Large Social Network Over 32 Years," *New England Journal of Medicine* 357 (4): 370-379 (26 July 2007)
- Doreian, Patrick, Vladimir Batagelj, and Anuska Ferligoj. (2005). *Generalized Blockmodeling*. Cambridge: Cambridge University Press. ISBN 0521840856
- Freeman, Linton C. (2004) *The Development of Social Network Analysis: A Study in the Sociology of Science*. Vancouver: Empirical Press. ISBN 1594577145
- Hill, R. and Dunbar, R. 2002. "Social Network Size in Humans." *Human Nature*, Vol. 14, No. 1, pp. 53–72.
- Jackson, Matthew O. (2003). "A Strategic Model of Social and Economic Networks". *Journal of Economic Theory* 71: 44–74. doi:10.1006/jeth.1996.0108. pdf
- Huisman, M. and Van Duijn, M. A. J. (2005). Software for Social Network Analysis. In P. J. Carrington, J. Scott, & S. Wasserman (Editors), *Models and Methods in Social Network Analysis* (pp. 270–316). New York: Cambridge University Press. ISBN 9780521809597
- Krebs, Valdis (2006) *Social Network Analysis, A Brief Introduction*. (Includes a list of recent SNA applications Web Reference.)
- Lin, Nan, Ronald S. Burt and Karen Cook, eds. (2001). *Social Capital: Theory and Research*. New York: Aldine de Gruyter. ISBN 0202306437
- Mullins, Nicholas. 1973. *Theories and Theory Groups in Contemporary American Sociology*. New York: Harper and Row. ISBN 0060446498
- Müller-Prothmann, Tobias (2006): *Leveraging Knowledge Communication for Innovation. Framework, Methods and Applications of Social Network Analysis in Research and Development*, Frankfurt a. M. et al.: Peter Lang, ISBN 0-8204-9889-0.
- Manski, Charles F. (2000). "Economic Analysis of Social Interactions". *Journal of Economic Perspectives* 14: 115–36. [3] via JSTOR
- Moody, James, and Douglas R. White (2003). "Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups." *American Sociological Review* 68(1):103-127. [4]
- Newman, Mark (2003). "The Structure and Function of Complex Networks". *SIAM Review* 56: 167–256. doi:10.1137/S003614450342480. pdf
- Nohria, Nitin and Robert Eccles (1992). *Networks in Organizations*. second ed. Boston: Harvard Business Press. ISBN 0875843247
- Nooy, Wouter d., A. Mrvar and Vladimir Batagelj. (2005). *Exploratory Social Network Analysis with Pajek*. Cambridge: Cambridge University Press. ISBN 0521841739
- Scott, John. (2000). *Social Network Analysis: A Handbook*. 2nd Ed. Newberry Park, CA: Sage. ISBN 0761963383
- Sethi, Arjun. (2008). *Valuation of Social Networking* [5]
- Tilly, Charles. (2005). *Identities, Boundaries, and Social Ties*. Boulder, CO: Paradigm press. ISBN 1594511314
- Valente, Thomas W. (1995). *Network Models of the Diffusion of Innovations*. Cresskill, NJ: Hampton Press. ISBN 1881303217
- Wasserman, Stanley, & Faust, Katherine. (1994). *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press. ISBN 0521382696
- Watkins, Susan Cott. (2003). "Social Networks." Pp. 909–910 in *Encyclopedia of Population*. rev. ed. Edited by Paul George Demeny and Geoffrey McNicoll. New York: Macmillan Reference. ISBN 0028656776
- Watts, Duncan J. (2003). *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton:

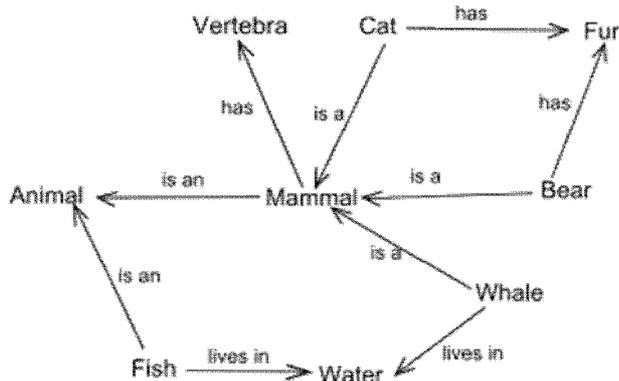
Princeton University Press. ISBN 0691117047

- Watts, Duncan J. (2004). *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company. ISBN 0393325423
- Wellman, Barry (1998). *Networks in the Global Village: Life in Contemporary Communities*. Boulder, CO: Westview Press. ISBN 0813311500
- Wellman, Barry. 2001. "Physical Place and Cyber-Place: Changing Portals and the Rise of Networked Individualism." *International Journal for Urban and Regional Research* 25 (2): 227-52.
- Wellman, Barry and Berkowitz, Stephen D. (1988). *Social Structures: A Network Approach*. Cambridge: Cambridge University Press. ISBN 0521244412
- Weng, M. (2007). *A Multimedia Social-Networking Community for Mobile Devices* Interactive Telecommunications Program, Tisch School of the Arts/ New York University
- White, Harrison, Scott Boorman and Ronald Breiger. 1976. "Social Structure from Multiple Networks: I Blockmodels of Roles and Positions." *American Journal of Sociology* 81: 730-80.

External links

- The International Network for Social Network Analysis (INSNA) - professional society of social network analysts, with more than 1,000 members
- Annual International Workshop on Social Network Mining and Analysis SNAKDD - Annual computer science workshop for interdisciplinary studies on social network mining and analysis
- VisualComplexity.com - a visual exploration on mapping complicated and complex networks
- Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon
- NetLab at the University of Toronto, studies the intersection of social, communication, information and computing networks
- Netwiki (wiki page devoted to social networks; maintained at University of North Carolina at Chapel Hill)
- Building networks for learning- A guide to on-line resources on strengthening social networking.
- Program on Networked Governance - Program on Networked Governance, Harvard University

10. Semantic network



Example of a semantic network

A **semantic network** is a network which represents semantic relations among concepts. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges.^[1]

History

"Semantic Nets" were first invented for computers by Richard H. Richens of the Cambridge Language Research Unit in 1956 as an "interlingua" for machine translation of natural languages.

They were developed by Robert F. Simmons at System Development Corporation in the early 1960s and later featured prominently in the work of Allan M. Collins and colleagues (e.g., Collins and Quillian,^{[2][3]} Collins and Loftus).^[4]

In the 1960s to 1980s the idea of a semantic link was developed within hypertext systems as the most basic unit, or edge, in a semantic network. These ideas were extremely influential, and there have been many attempts to add typed link semantics to HTML and XML.

Semantic network construction

WordNet

An example of a semantic network is WordNet, a lexical database of English. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Some of the most common semantic relations defined are meronymy (A is part of B, i.e. B has A as a part of itself), holonymy (B is part of A, i.e. A has B as a part of itself), hyponymy (or troponymy) (A is subordinate of B; A is kind of B), hypernymy (A is superordinate of B), synonymy (A denotes the same as B) and antonymy (A denotes the opposite of B).

WordNet properties have been studied from a network theory perspective and compared to other semantic networks created from Roget's Thesaurus and word association tasks. From this perspective the three of them are a small world structure.^[5]

It is also possible to represent logical descriptions using semantic networks such as the existential Graphs of Charles Sanders Peirce or the related Conceptual Graphs of John F. Sowa.^[1] These have expressive power equal to or exceeding standard first-order predicate logic. Unlike WordNet or other lexical or browsing networks, semantic networks using these representations can be used for reliable automated logical deduction. Some automated reasoners exploit the graph-theoretic features of the networks during processing.

Other examples

Other examples of semantic networks are Gellish models. Gellish English with its Gellish English dictionary, is a formal language that is defined as a network of relations between concepts and names of concepts. Gellish English is a formal subset of natural English, just as Gellish Dutch is a formal subset of Dutch, whereas multiple languages share the same concepts. Other Gellish networks consist of knowledge models and information models that are expressed in the Gellish language. A Gellish network is a network of (binary) relations between things. Each relation in the network is an expression of a fact that is classified by a relation type. Each relation type itself is a concept that is defined in the Gellish language dictionary. Each related thing is either a concept or an individual thing that is classified by a concept. The definitions of concepts are created in the form of definition models (definition networks) that together form a Gellish Dictionary. A Gellish network can be documented in a Gellish database and is computer interpretable.

Software tools

There are also elaborate types of semantic networks connected with corresponding sets of software tools used for lexical knowledge engineering, like the Semantic Network Processing System (SNePS) of Stuart C. Shapiro^[6] or the MultiNet paradigm of Hermann Helbig,^[7] especially suited for the semantic representation of natural language expressions and used in several NLP applications.

See also

- | | |
|---|---|
| <ul style="list-style-type: none"> • Conceptual graph • Knowledge representation • Mind map • Network diagram | <ul style="list-style-type: none"> • Semantic lexicon • Semantic neural network • Unified Medical Language System • Word Sense Disambiguation |
|---|---|

Examples

- | | |
|--|---|
| <ul style="list-style-type: none"> • Lexipedia • WordNet | <ul style="list-style-type: none"> • SNOMED CT |
|--|---|

References

1. ^{a b} John F. Sowa (1987). "Semantic Networks". in Stuart C Shapiro. *Encyclopedia of Artificial Intelligence*. <http://www.jfsowa.com/pubs/semnet.htm>. Retrieved 2008-04-29.
2. ^a Allan M. Collins; M.R. Quillian (1969). "Retrieval time from semantic memory". *Journal of verbal learning and verbal behavior* **8** (2): 240–248. doi:10.1016/S0022-5371(69)80069-1.
3. ^a Allan M. Collins; M. Ross Quillian (1970). "Does category size affect categorization time?". *Journal of verbal learning and verbal behavior* **9** (4): 432–438. doi:10.1016/S0022-5371(70)80084-6.
4. ^a Allan M. Collins; Elizabeth F. Loftus (1975). "A spreading-activation theory of semantic processing". *Psychological Review* **82** (6): 407–428. doi:10.1037/0033-295X.82.6.407.
5. ^a Steyvers, M.; Tenenbaum, J.B. (2005). "The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth". *Cognitive Science* **29** (1): 41–78. doi:10.1207/s15516709cog2901_3.
6. ^a Stuart C. Shapiro

7. ^ Hermann Helbig

Further reading

- Allen, J. and A. Frisch (1982). "What's in a Semantic Network". In: *Proceedings of the 20th. annual meeting of ACL*, Toronto, pp. 19-27.
- John F. Sowa, Alexander Borgida (1991). *Principles of Semantic Networks: Explorations in the Representation of Knowledge*.

External links

- "Semantic Networks" by John F. Sowa
- "Semantic Link Network" by Hai Zhuge

11. Radio and Television networks

There are two types of radio networks currently in use around the world: the one-to-many broadcast type commonly used for public information and entertainment; and the two-way type used more commonly for public safety and public services such as police, fire, taxis, and delivery services. Following is a description of the former type of radio network although many of the same components and much of the same basic technology applies to both.

The **Broadcast** type of **radio network** is a network system which distributes programming to multiple stations simultaneously, or slightly delayed, for the purpose of extending total coverage beyond the limits of a single broadcast signal. The resulting expanded audience for programming or information essentially applies the benefits of mass-production to the broadcasting enterprise. A radio network has two sales departments, one to package and sell programs to radio stations, and one to sell the audience of those programs to advertisers.

Most radio networks also produce much of their programming. Originally, radio networks owned some or all of the radio stations that broadcast the network's programming. Presently however, there are many networks that do not own any stations and only produce and/or distribute programming. Similarly station ownership does not always indicate network affiliation. A company might own stations in several different markets and purchase programming from a variety of networks.

Radio networks rose rapidly with the growth of regular broadcasting of radio to home listeners in the 1920s. This growth took various paths in different places. In Britain the BBC was developed with public funding, in the form of a broadcast receiving license, and a broadcasting monopoly in its early decades. In contrast, in the United States of America various competing commercial networks arose funded by advertising revenue. In that instance, the same corporation that owned or operated the network often manufactured and marketed the listener's radio.

Major technical challenges to be overcome when distributing programs over long distances are maintaining signal quality and managing the number of switching/relay points in the signal chain. Early on, programs were sent to remote stations (either owned or affiliated) by various methods, including leased telephone lines, pre-recorded gramophone records and audio tape. The world's first all-radio, non-wireline network was claimed to be the Rural Radio Network, a group of six upstate New York FM stations that began operation in June 1948. Terrestrial microwave relay, a technology later introduced to link stations, has been largely supplanted by coaxial cable, fiber, and satellite, which usually offer superior cost-benefit ratios.

Many early radio networks evolved into Television networks.

Radio network

The **Two-way** type of **radio network** shares many of the same technologies and components as the **Broadcast type radio network** but is generally set up with fixed broadcast points (transmitters) with co-located receivers and mobile receivers/transmitters or **Tran-ceiver**s. In this way both the fixed and mobile radio units can communicate with each other over broad geographic regions ranging in size from small single cities to entire states/provinces or countries. There are many ways in which multiple fixed transmit/receive sites can be interconnected to achieve the range of coverage required by the jurisdiction or authority implementing the system: conventional wireless links in numerous frequency bands, fibre-optic links, or micro-wave links. In all of these cases the signals are typically backhauled to a central switch of some type where the radio message is processed and resent (repeated) to all transmitter sites where it is required to be heard.

In contemporary two-way radio systems a concept called **trunking** is commonly used to achieve better efficiency of radio spectrum use and provide very wide ranging coverage with no switching of channels required by the

mobile radio user as it roams throughout the system coverage. Trunking of two-way radio is identical to the concept used for cellular phone systems where each fixed and mobile radio is specifically identified to the system **Controller** and its operation is switched by the controller. See also the entries Two-way radio and Trunked radio system to see more detail on how various types of radios and radio systems work.

Television network

A **television network** is a distribution network for television content whereby a central operation provides programming for many television stations. Until the mid-1980s, television programming in most countries of the world was dominated by a small number of broadcast networks. Many early television networks (e.g. the BBC, NBC or CBS) evolved from earlier radio networks.

In countries where most networks broadcast identical, centrally originated content to all their stations and where most individual transmitters therefore operate only as large "repeater stations", the terms *television network*, *television channel* and *television station* have become interchangeable in everyday language, with only professionals in TV-related occupations continuing to make a difference between them. Within the industry, a tiering is sometimes created among groups of networks based on whether their programming is simultaneously originated from a central point, and whether the network master control has the technical and administrative capability to take over the programming of their affiliates in real-time when it deems this necessary—the most common example being breaking national news events.

In North America in particular, many television channels available via cable and satellite television are branded as "networks" but are not truly networks in the sense defined above, as they are singular operations – they have no affiliates or component stations. Such channels are more precisely referred to by terms such as "specialty channels" (Canada) or "cable networks" (U.S.), although the latter term is somewhat of a misnomer, even though these channels are networked across the country by various cable and satellite systems.

A network may or may not produce all of its own programming. If not, production houses such as Warner Bros. and Sony Pictures can distribute their content to the different networks and it is common that a certain production house may have programmes on two or more rival networks. Similarly, some networks may import television programmes from other countries or use archival programming to help complement their schedules.

Regulation

FCC regulations in the United States restricted the number of television stations that could be owned by any one network, company or individual. This led to a system where most local television stations were independently owned, but received programming from the network through a franchising contract, except in a few big cities that had network owned-and-operated stations and independent stations. In the early days of television, when there were often only one or two stations broadcasting in an area, the stations were usually affiliated with several networks and were able to choose which programs to air. Eventually, as more stations were licensed, it became common for each station to be affiliated with only one network and carry all of the "prime time" network programs. Local stations however occasionally break from regularly scheduled network programming, especially when there is breaking local news (e.g. severe weather). Moreover, when stations return to network programming from commercial breaks, the station's logo is displayed in the first few seconds before switching to the network's logo.

Another FCC regulation, the Prime Time Access Rule, restricted the number of hours of network programming that could be broadcast on the local affiliate stations. This was done to encourage the development of locally produced programs and to give local residents access to broadcast time. More often, the result included a substantial amount of syndicated programming, usually consisting of old movies, independently produced and syndicated shows, and reruns of network programs. Occasionally, these shows were presented by a local host, especially in programs that showed cartoons and short comedies intended for children. See List of local children's television series (United States).

12. Business networking

Business networking is a marketing method by which business opportunities are created through networks of like-minded business people. There are several prominent business networking organizations that create models of networking activity that, when followed, allow the business person to build new business relationship and generate business opportunities at the same time.

Many business people contend business networking is a more cost-effective method of generating new business

than advertising or public relations efforts. This is because business networking is a low-cost activity that involves more personal commitment than company money.

As an example, a business network may agree to meet weekly or monthly with the purpose of exchanging business leads and referrals with fellow members. To complement this activity, members often meet outside this circle, on their own time, and build their own "one-to-one" relationship with the fellow member.

Business networking can be conducted in a local business community, or on a more larger scale via the Internet. Business networking websites have grown over recent years due to the Internet's ability to connect people from all over the world.

Business networking can have a meaning also in the ICT domain, i.e. the provision of operating support to companies / organizations, and related value chains / value networks.

It refers to an activity coordination with a wider scope and a simpler implementation than pre-organized workflows or web-based impromptu searches for transaction counterparts (workflow is useful to coordinate activities, but it is complicated by the use of s.c. "patterns" to deviate the flow of work from a pure sequence, in order to compensate its intrinsic "linearity"; impromptu searches for transaction counterparts on the web are useful as well, but only for non strategic supplies; both are complicated by a plethora of interfaces -- SOA / XML / web services -- needed among different organizations and even between different IT applications within the same organization).

Online business networking

Businesses are increasingly using business social networks like **Business Book**[[1]]or professional business networking tools like Boardex as a means of growing their circle of business contacts and promoting themselves online. Since businesses are expanding globally, social networks make it easier to keep in touch with other contacts around the world. Specific cross-border e-commerce platforms and business partnering networks now make globalization accessible also for small and medium sized companies.

Face-to-face business networking

Professionals who wish to leverage their presentation skills with the urgency of physically being present attend general and exclusive events. Many professionals tend to prefer face-to-face networking over online based networking because the potential for higher quality relationships are possible. Many individuals also prefer face-to-face because people tend to prefer actually knowing and meeting who they intend to do business with.

General business networking

Before online networking, there was and has always been, networking face-to-face. "Schmoozing" or "rubbing elbows" are expressions used among business professionals for introducing and meeting one another, and establishing rapport.

Networked Businesses

With networking developing many businesses now have this as a core part of their strategy, those that have developed a strong network of connections suppliers and companies can be seen as "Networked Businesses" and will tend to source the business and their suppliers through the network of relationships that they have in place. Networked businesses tend to be Open, Random and Supportive - ORS whereas those relying on hierarchical, traditional managed approaches are Closed Selective and Controlling - CSC.

Business networking in the ICT domain

Companies / organizations - and related value chains / value networks - need some sort of IT support. Traditionally, it is provided by software applications, software packages /suites, ERPs and/or workflows; presently, also by different types of web-based innovations.

A truly "ICT" business networking approach rethinks - and rebuilds - the operating support from scratch, around two key business features: information contributions, to be provided by the activities involved (whether they are performed by human beings, automated tools or jointly by the two, in a coordinated way); (automated) information exchanges, to be provided by the TLC network.

Information contributions and exchanges, in turn, need to be supported by data storage (plain or redundant, with or without automated recovery to grant service continuity) and access security (signature, encryption, authentication, decryption), which both can be provided either as add-ons or as built-in features.

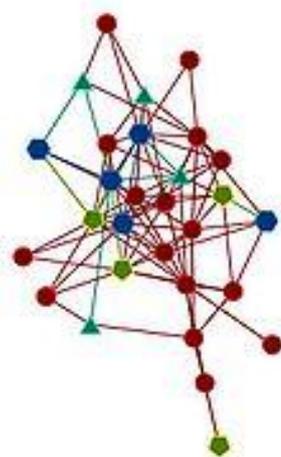
13. Dynamic network analysis

Dynamic network analysis (DNA) is an emergent scientific field that brings together traditional social network analysis (SNA), link analysis (LA) and multi-agent systems (MAS) within network science and network theory. There are two aspects of this field. The first is the statistical analysis of DNA data. The second is the utilization of simulation to address issues of network dynamics. DNA networks vary from traditional social networks in that they are larger, dynamic, multi-mode, multi-plex networks, and may contain varying levels of uncertainty.

DNA statistical tools are generally optimized for large-scale networks and admit the analysis of multiple networks simultaneously in which, there are multiple types of nodes (multi-node) and multiple types of links (multi-plex). In contrast, SNA statistical tools focus on single or at most two mode data and facilitate the analysis of only one type of link at a time.

DNA statistical tools tend to provide more measures to the user, because they have measures that use data drawn from multiple networks simultaneously. From a computer simulation perspective, nodes in DNA are like atoms in quantum theory, nodes can be, though need not be, treated as probabilistic. Whereas nodes in a traditional SNA model are static, nodes in a DNA model have the ability to learn.

Properties change over time; nodes can adapt: A company's employees can learn new skills and increase their value to the network; Or, capture one terrorist and three more are forced to improvise. Change propagates from one node to the next and so on. DNA adds the element of a network's evolution and considers the circumstances under which change is likely to occur.



An example of a multi-entity, multi-network, dynamic network diagram

Illustrative problems that people in the DNA area work on

- Developing metrics and statistics to assess and identify change within and across networks.
- Developing and validating simulations to study network change, evolution, adaptation, decay... See Computer simulation and organizational studies
- Developing and validating formal models of network generation and evolution
- Developing and testing theory of network change, evolution, adaptation, decay...
- Developing techniques to visualize network change overall or at the node or group level
- Developing statistical techniques to see whether differences observed over time in networks are due to simply different samples from a distribution of links and nodes or changes over time in the underlying distribution of links and nodes
- Developing control processes for networks over time
- Developing algorithms to change distributions of links in networks over time
- Developing algorithms to track groups in networks over time.
- Developing tools to extract or locate networks from various data sources such as texts.
- Developing statistically valid measurements on networks over time.
- Examining the robustness of network metrics under various types of missing data
- Empirical studies of multi-mode multi-link multi-time period networks
- Examining networks as probabilistic time-variant phenomena
- Forecasting change in existing networks
- Identifying trails through time given a sequence of networks.
- Identifying changes in node criticality given a sequence of networks anything else related to multi-mode multi-link multi-time period networks.

Further reading

- Kathleen M. Carley, 2003, “Dynamic Network Analysis” in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Ronald Breiger, Kathleen Carley, and Philippa Pattison, (Eds.) Committee on Human Factors, National Research Council, National Research Council. Pp. 133–145, Washington, DC.
- Kathleen M. Carley, 2002, “Smart Agents and Organizations of the Future” The Handbook of New Media. Edited by Leah Lievrouw and Sonia Livingstone, Ch. 12, pp. 206–220, Thousand Oaks, CA, Sage.
- Kathleen M. Carley, Jana Diesner, Jeffrey Reminga, Maksim Tsvetovat, 2008, Toward an Interoperable Dynamic Network Analysis Toolkit, DSS Special Issue on Cyberinfrastructure for Homeland Security: Advances in Information Sharing, Data Mining, and Collaboration Systems. Decision Support Systems 43(4):1324-1347 (article 20)
- Terrill L. Frantz, Kathleen M. Carley. 2009, Toward A Confidence Estimate For The Most-Central-Actor Finding. Academy of Management Annual Conference, Chicago, IL, USA, 7-11 August. (Awarded the Sage Publications/RM Division Best Student Paper Award)

See also

- | | |
|--|--|
| <ul style="list-style-type: none"> • Network dynamics • Sequential dynamical system • Kathleen Carley | <ul style="list-style-type: none"> • Network science • INSNA |
|--|--|

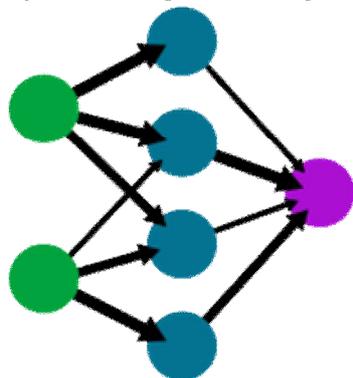
External links

- Radcliffe Exploratory Seminar on Dynamic Networks
- Center for Computational Analysis of Social and Organizational Systems (CASOS)

14. Neural network

A simple neural network

input layer hidden layer output layer



📄 Simplified view of a feedforward artificial neural network

Traditionally, the term **neural network** had been used to refer to a network or circuit of biological neurons^[citation needed]. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term has two distinct usages:

1. Biological neural networks are made up of real biological neurons that are connected or functionally related in the peripheral nervous system or the central nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.
2. Artificial neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real, biological nervous system is highly complex and includes some features that may seem superfluous based on an understanding of artificial networks.

This article focuses on the relationship between the two concepts; for detailed coverage of the two different concepts refer to the separate articles: Biological neural network and Artificial neural network.

Overview

In general a biological neural network is composed of a group or groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons and connections in a network may be extensive. Connections, called synapses, are usually formed from axons to dendrites, though dendrodendritic microcircuits^[1] and other connections are possible. Apart from the electrical signaling, there are other forms of signaling that arise from neurotransmitter diffusion, which have an effect on electrical signaling. As such, neural networks are extremely complex.

Artificial intelligence and cognitive modeling try to simulate some properties of neural networks. While similar in their techniques, the former has the aim of solving particular tasks, while the latter aims to build mathematical

models of biological neural systems.

In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots. Most of the currently employed artificial neural networks for artificial intelligence are based on statistical estimation, optimization and control theory.

The cognitive modelling field involves the physical or mathematical modeling of the behaviour of neural systems; ranging from the individual neural level (e.g. modelling the spike response curves of neurons to a stimulus), through the neural cluster level (e.g. modelling the release and effects of dopamine in the basal ganglia) to the complete organism (e.g. behavioural modelling of the organism's response to stimuli). Artificial intelligence, cognitive modelling, and neural networks are information processing paradigms inspired by the way biological neural systems process data.

History of the neural network analogy

The concept of neural networks started in the late-1800s as an effort to describe how the human mind performed. These ideas started being applied to computational models with Turing's B-type machines and the perceptron.

In early 1950s Friedrich Hayek was one of the first to posit the idea of spontaneous order^[citation needed] in the brain arising out of decentralized networks of simple units (neurons). In the late 1940s, Donald Hebb made one of the first hypotheses for a mechanism of neural plasticity (i.e. learning), Hebbian learning. Hebbian learning is considered to be a 'typical' unsupervised learning rule and it (and variants of it) was an early model for long term potentiation.

The perceptron is essentially a linear classifier for classifying data $x \in \mathbb{R}^n$ specified by parameters $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ and an output function $f = w \cdot x + b$. Its parameters are adapted with an ad-hoc rule similar to stochastic steepest gradient descent. Because the inner product is a linear operator in the input space, the Perceptron can only perfectly classify a set of data for which different classes are linearly separable in the input space, while it often fails completely for non-separable data. While the development of the algorithm initially generated some enthusiasm, partly because of its apparent relation to biological mechanisms, the later discovery of this inadequacy caused such models to be abandoned until the introduction of non-linear models into the field.

The cognitron (1975) was an early multilayered neural network with a training algorithm. The actual structure of the network and the methods used to set the interconnection weights change from one neural strategy to another, each with its advantages and disadvantages. Networks can propagate information in one direction only, or they can bounce back and forth until self-activation at a node occurs and the network settles on a final state. The ability for bi-directional flow of inputs between neurons/nodes was produced with the Hopfield's network (1982), and specialization of these node layers for specific purposes was introduced through the first hybrid network.

The parallel distributed processing of the mid-1980s became popular under the name connectionism.

The rediscovery of the backpropagation algorithm was probably the main reason behind the repopularisation of neural networks after the publication of "Learning Internal Representations by Error Propagation" in 1986 (Though backpropagation itself dates from 1974).

The original network utilised multiple layers of weight-sum units of the type $f = g(w \cdot x + b)$, where g was a sigmoid function or logistic function such as used in logistic regression. Training was done by a form of stochastic steepest gradient descent. The employment of the chain rule of differentiation in deriving the appropriate parameter updates results in an algorithm that seems to 'backpropagate errors', hence the nomenclature. However it is essentially a form of gradient descent. Determining the optimal parameters in a model of this type is not trivial, and steepest gradient descent methods cannot be relied upon to give the solution without a good starting point. In recent times, networks with the same architecture as the backpropagation network are referred to as Multi-Layer Perceptrons. This name does not impose any limitations on the type of algorithm used for learning.

The backpropagation network generated much enthusiasm at the time and there was much controversy about whether such learning could be implemented in the brain or not, partly because a mechanism for reverse signalling was not obvious at the time, but most importantly because there was no plausible source for the 'teaching' or 'target' signal.

The brain, neural networks and computers

Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and brain biological architecture is debated^[citation needed].

A subject of current research in theoretical neuroscience is the question surrounding the degree of complexity and the properties that individual neural elements should have to reproduce something resembling animal intelligence.

Historically, computers evolved from the von Neumann architecture, which is based on sequential processing and execution of explicit instructions. On the other hand, the origins of neural networks are based on efforts to model information processing in biological systems, which may rely largely on parallel processing as well as implicit instructions based on recognition of patterns of 'sensory' input from external sources. In other words, at its very heart a neural network is a complex statistical processor (as opposed to being tasked to sequentially process and execute).

Neural networks and artificial intelligence

A *neural network* (NN), in the case of artificial neurons called *artificial neural network* (ANN) or *simulated neural network* (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

However, the paradigm of neural networks - i.e., *implicit*, and not *explicit* learning is stressed - seems more to correspond to some kind of *natural intelligence* than to the traditional *Artificial Intelligence*, which would stress, instead, rule-based learning.

Background

An artificial neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. Artificial neurons were first proposed in 1943 by Warren McCulloch, a neurophysiologist, and Walter Pitts, an MIT logician.[1] One classical type of artificial neural network is the recurrent Hopfield net.

In a neural network model simple nodes, which can be called variously "neurons", "neurodes", "Processing Elements" (PE) or "units", are connected together to form a network of nodes — hence the term "neural network". While a neural network does not have to be adaptive *per se*, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.

In modern software implementations of artificial neural networks the approach inspired by biology has more or less been abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks, or parts of neural networks (such as artificial neurons), are used as components in larger systems that combine both adaptive and non-adaptive elements.

The concept of a neural network appears to have first been proposed by Alan Turing in his 1948 paper "Intelligent Machinery".

Applications of natural and of artificial neural networks

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

Real life applications

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modelling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind signal separation and compression.

Application areas of ANNs include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition, etc.), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

- Moreover, some brain diseases, e.g. Alzheimer, are apparently, and essentially, diseases of the brain's natural NN by damaging necessary prerequisites for the functioning of the mutual interconnections between neurons and/or glia.

Neural network software

Neural network software is used to simulate, research, develop and apply artificial neural networks, biological neural networks and in some cases a wider array of adaptive systems.

Learning paradigms

There are three major learning paradigms, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning. Usually any given type of network architecture can be employed in any of those tasks.

- Supervised learning

In supervised learning, we are given a set of example pairs $(x, y), x \in X, y \in Y$ and the aim is to find a function f in the allowed class of functions that matches the examples. In other words, we wish to *infer* how the mapping implied by the data and the cost function is related to the mismatch between our mapping and the data.

- Unsupervised learning

In unsupervised learning we are given some data x , and a cost function which is to be minimized which can be any function of x and the network's output, f . The cost function is determined by the task formulation. Most applications fall within the domain of estimation problems such as statistical modeling, compression, filtering, blind source separation and clustering.

- Reinforcement learning

In reinforcement learning, data x is usually not given, but generated by an agent's interactions with the environment. At each point in time t , the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t , according to some (usually unknown) dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost, i.e. the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated. ANNs are frequently used in reinforcement learning as part of the overall algorithm. Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

Learning algorithms

There are many algorithms for training neural networks; most of them can be viewed as a straightforward application of optimization theory and statistical estimation. They include: Back propagation by gradient descent, Rprop, BFGS, CG etc.

Evolutionary computation methods, simulated annealing, expectation maximization and non-parametric methods are among other commonly used methods for training neural networks. See also machine learning.

Recent developments in this field also saw the use of particle swarm optimization and other swarm intelligence techniques used in the training of neural networks.

Neural networks and neuroscience

Theoretical and computational neuroscience is the field concerned with the theoretical analysis and computational modeling of biological neural systems. Since neural systems are intimately related to cognitive processes and behaviour, the field is closely related to cognitive and behavioural modeling.

The aim of the field is to create models of biological neural systems in order to understand how biological systems work. To gain this understanding, neuroscientists strive to make a link between observed biological processes (data), biologically plausible mechanisms for neural processing and learning (biological neural network models) and theory (statistical learning theory and information theory).

Types of models

Many models are used in the field, each defined at a different level of abstraction and trying to model different aspects of neural systems. They range from models of the short-term behaviour of individual neurons, through models of how the dynamics of neural circuitry arise from interactions between individual neurons, to models of how behaviour can arise from abstract neural modules that represent complete subsystems. These include models of the long-term and short-term plasticity of neural systems and its relation to learning and memory, from the individual neuron to the system level.

Current research

While initially research had been concerned mostly with the electrical characteristics of neurons, a particularly important part of the investigation in recent years has been the exploration of the role of neuromodulators such

as dopamine, acetylcholine, and serotonin on behaviour and learning.

Biophysical models, such as BCM theory, have been important in understanding mechanisms for synaptic plasticity, and have had applications in both computer science and neuroscience. Research is ongoing in understanding the computational algorithms used in the brain, with some recent biological evidence for radial basis networks and neural backpropagation as mechanisms for processing data.

Criticism

A common criticism of neural networks, particularly in robotics, is that they require a large diversity of training for real-world operation. Dean Pomerleau, in his research presented in the paper "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving," uses a neural network to train a robotic vehicle to drive on multiple types of roads (single lane, multi-lane, dirt, etc.). A large amount of his research is devoted to (1) extrapolating multiple training scenarios from a single training experience, and (2) preserving past training diversity so that the system does not become overtrained (if, for example, it is presented with a series of right turns – it should not learn to always turn right). These issues are common in neural networks that must decide from amongst a wide variety of responses.

A. K. Dewdney, a former *Scientific American* columnist, wrote in 1997, "Although neural nets do solve a few toy problems, their powers of computation are so limited that I am surprised anyone takes them seriously as a general problem-solving tool." (Dewdney, p. 82)

Arguments for Dewdney's position are that to implement large and effective software neural networks, much processing and storage resources need to be committed. While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a most simplified form on Von Neumann technology may compel a NN designer to fill many millions of database rows for its connections - which can lead to abusive RAM and HD necessities. Furthermore, the designer of NN systems will often need to simulate the transmission of signals through many of these connections and their associated neurons - which must often be matched with incredible amounts of CPU processing power and time. While neural networks often yield *effective* programs, they too often do so at the cost of time and money *efficiency*.

Arguments against Dewdney's position are that neural nets have been successfully used to solve many complex and diverse tasks, ranging from autonomously flying aircraft[2] to detecting credit card fraud[3].

Technology writer Roger Bridgman commented on Dewdney's statements about neural nets:

Neural networks, for instance, are in the dock not only because they have been hyped to high heaven, (what hasn't?) but also because you could create a successful net without understanding how it worked: the bunch of numbers that captures its behaviour would in all probability be "an opaque, unreadable table...valueless as a scientific resource". In spite of his emphatic declaration that science is not technology, Dewdney seems here to pillory neural nets as bad science when most of those devising them are just trying to be good engineers. An unreadable table that a useful machine could read would still be well worth having.^[2]

Some other criticisms came from believers of hybrid models (combining neural networks and symbolic approaches). They advocate the intermix of these two approaches and believe that hybrid models can better capture the mechanisms of the human mind (Sun and Bookman 1994).

See also

- ADALINE
- Artificial neural network
- Biological cybernetics
- Biologically inspired computing
- Cerebellar Model Articulation Controller
- Cognitive architecture
- Cognitive science
- Cultured neuronal networks
- In Situ Adaptive Tabulation
- Memristor
- Neural network software
- Neuro-fuzzy
- Neuroscience
- Parallel distributed processing
- Predictive analytics
- Radial basis function network
- Recurrent neural networks
- Simulated reality
- Support vector machine
- Tensor product network
- Time delay neural network
- 20Q is a neural network implementation of the 20 questions game

References

1. ^ Arbib, p.666
2. ^ Roger Bridgman's defence of neural networks

Further reading

- Arbib, Michael A. (Ed.) (1995). *The Handbook of Brain Theory and Neural Networks*.
- Alspector, U.S. Patent 4,874,963 "Neuromorphic learning networks". October 17, 1989.
- Agre, Philip E. (1997). *Computation and Human Experience*. Cambridge Univ. Press. ISBN 0-521-38603-9., p. 80
- Yaneer Bar-Yam (2003). *Dynamics of Complex Systems, Chapter 2*.
- Yaneer Bar-Yam (2003). *Dynamics of Complex Systems, Chapter 3*.
- Yaneer Bar-Yam (2005). *Making Things Work*. See chapter 3.
- Bertsekas, Dimitri P. (1999). *Nonlinear Programming*.
- Bertsekas, Dimitri P. & Tsitsiklis, John N. (1996). *Neuro-dynamic Programming*.
- Bhadeshia H. K. D. H. (1992). "Neural Networks in Materials Science". *ISIJ International* **39**: 966–979. doi:10.2355/isijinternational.39.966.
- Boyd, Stephen & Vandenberghe, Lieven (2004). *Convex Optimization*.
- Dewdney, A. K. (1997). *Yes, We Have No Neutrons: An Eye-Opening Tour through the Twists and Turns of Bad Science*. Wiley, 192 pp. See chapter 5.
- Egmont-Petersen, M., de Ridder, D., Handels, H. (2002). "Image processing with neural networks - a review". *Pattern Recognition* **35** (10): 2279–2301. doi:10.1016/S0031-3203(01)00178-9.
- Fukushima, K. (1975). "Cognitron: A Self-Organizing Multilayered Neural Network". *Biological Cybernetics* **20**: 121–136. doi:10.1007/BF00342633.
- Frank, Michael J. (2005). "Dynamic Dopamine Modulation in the Basal Ganglia: A Neurocomputational Account of Cognitive Deficits in Medicated and Non-medicated Parkinsonism". *Journal of Cognitive Neuroscience* **17**: 51–72. doi:10.1162/0898929052880093.
- Gardner, E.J., & Derrida, B. (1988). "Optimal storage properties of neural network models". *Journal of Physics a* **21**: 271–284. doi:10.1088/0305-4470/21/1/031.
- Krauth, W., & Mezard, M. (1989). "Storage capacity of memory with binary couplings". *Journal de Physique* **50**: 3057–3066. doi:10.1051/jphys:0198900500200305700.
- Maass, W., & Markram, H. (2002). "On the computational power of recurrent circuits of spiking neurons". *Journal of Computer and System Sciences* **69**(4): 593–616.
- MacKay, David (2003). *Information Theory, Inference, and Learning Algorithms*.
- Mandic, D. & Chambers, J. (2001). *Recurrent Neural Networks for Prediction: Architectures, Learning algorithms and Stability*. Wiley.
- Minsky, M. & Papert, S. (1969). *An Introduction to Computational Geometry*. MIT Press.
- Muller, P. & Insua, D.R. (1995). "Issues in Bayesian Analysis of Neural Network Models". *Neural Computation* **10**: 571–592.
- Reilly, D.L., Cooper, L.N. & Elbaum, C. (1982). "A Neural Model for Category Learning". *Biological Cybernetics* **45**: 35–41. doi:10.1007/BF00387211.
- Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Books.
- Sun, R. & Bookman, L. (eds.) (1994.). *Computational Architectures Integrating Neural and Symbolic Processes*. Kluwer Academic Publishers, Needham, MA..
- Sutton, Richard S. & Barto, Andrew G. (1998). *Reinforcement Learning: An introduction*.
- Van den Bergh, F. Engelbrecht, AP. *Cooperative Learning in Neural Networks using Particle Swarm Optimizers*. CIRG 2000.
- Wilkes, A.L. & Wade, N.J. (1997). "Bain on Neural Networks". *Brain and Cognition* **33**: 295–305. doi:10.1006/brcg.1997.0869.
- Wasserman, P.D. (1989). *Neural computing theory and practice*. Van Nostrand Reinhold.
- Jeffrey T. Spooner, Manfredi Maggiore, Raul Ord onez, and Kevin M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, John Wiley and Sons, NY, 2002.
- Peter Dayan, L.F. Abbott. *Theoretical Neuroscience*. MIT Press.
- Wulfram Gerstner, Werner Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.
- Steeb, W-H (2008). *The Nonlinear Workbook: Chaos, Fractals, Neural Networks, Genetic Algorithms, Gene Expression Programming, Support Vector Machine, Wavelets, Hidden Markov Models, Fuzzy Logic with C++, Java and SymbolicC++ Programs: 4th edition*. World Scientific Publishing. ISBN 981-281-852-9.

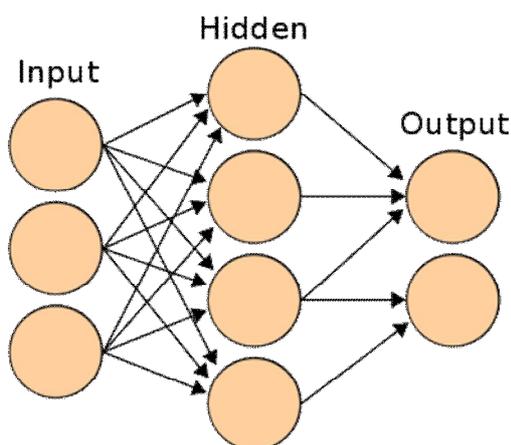
External links

- International Neural Network Society (INNS)
- European Neural Network Society (ENNS)

- Japanese Neural Network Society (JNNS)
- IEEE Computational Intelligence Society (IEEE CIS)
- A Brief Introduction to Neural Networks (D. Kriesel) - Illustrated, bilingual manuscript about artificial neural networks; Topics so far: Perceptrons, Backpropagation, Radial Basis Functions, Recurrent Neural Networks, Self Organizing Maps, Hopfield Networks.
- Flood: An open source neural networks C++ library
- LearnArtificialNeuralNetworks - Robot control and neural networks
- NPatternRecognizer - A fast machine learning algorithm library written in C#. It contains SVM, neural networks, Bayes, boost, k-nearest neighbor, decision tree, ..., etc.
- Review of Neural Networks in Materials Science
- Artificial Neural Networks Tutorial in three languages (Univ. Politécnica de Madrid)
- Introduction to Neural Networks and Knowledge Modeling
- Introduction to Artificial Neural Networks
- In Situ Adaptive Tabulation: - A neural network alternative.
- Another introduction to ANN
- Prediction with neural networks - includes Java applet for online experimenting with prediction of a function
- Next Generation of Neural Networks - Google Tech Talks
- Perceptual Learning - Artificial Perceptual Neural Network used for machine learning to play Chess
- European Centre for Soft Computing
- Performance of Neural Networks
- Neural Networks and Information

15. Artificial neural network

An **artificial neural network (ANN)**, usually called "neural network" (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.



 A neural network is an interconnected group of nodes, akin to the vast network of neurons in the human brain.

Background

There is no precise agreed-upon definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons), which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. The original inspiration for the technique came from examination of the central nervous system and the neurons (and their axons, dendrites and synapses) which constitute one of its most significant information processing elements (see neuroscience). In a neural network model, simple nodes, called variously "neurons", "neurodes", "PEs" ("processing elements")

or "units", are connected together to form a network of nodes — hence the term "neural network". While a neural network does not have to be adaptive per se, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.

These networks are also similar to the biological neural networks in the sense that functions are performed collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned (see also connectionism). Currently, the term Artificial Neural Network (ANN) tends to refer mostly to neural network models employed in statistics, cognitive psychology and artificial intelligence. Neural network models designed with emulation of the central nervous system (CNS) in mind are a subject of theoretical neuroscience (computational neuroscience).

In modern software implementations of artificial neural networks the approach inspired by biology has for the most part been abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do have in common, however, is the principle of non-linear, distributed, parallel and local processing and adaptation.

Models

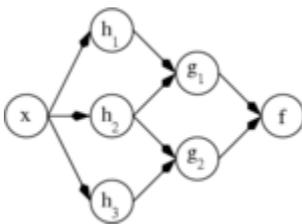
Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function $f : X \rightarrow Y$. Each type of ANN model corresponds to a class of such functions.

The network in artificial neural network

The word *network* in the term 'artificial neural network' arises because the function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A

$$f(x) = K \left(\sum_i w_i g_i(x) \right)$$

widely used type of composition is the *nonlinear weighted sum*, where K (commonly referred to as the activation function^[1]) is some predefined function, such as the hyperbolic tangent. It will be convenient for the following to refer to a collection of functions g_i as simply a vector $g = (g_1, g_2, \dots, g_n)$.



ANN dependency graph

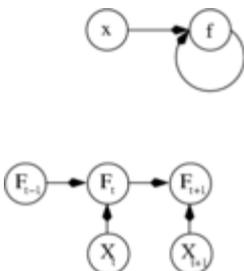
This figure depicts such a decomposition of f , with dependencies between variables indicated by arrows. These can be interpreted in two ways.

The first view is the functional view: the input x is transformed into a 3-dimensional vector h , which is then transformed into a 2-dimensional vector g , which is finally transformed into f . This view is most commonly encountered in the context of optimization.

The second view is the probabilistic view: the random variable $F = f(G)$ depends upon the random variable $G = g(H)$, which depends upon $H = h(X)$, which depends upon the random variable X . This view is most commonly encountered in the context of graphical models.

The two views are largely equivalent. In either case, for this particular network architecture, the components of individual layers are independent of each other (e.g., the components of g are independent of each other given their input h). This naturally enables a degree of parallelism in the implementation.

Recurrent ANN dependency graph



Networks such as the previous one are commonly called feedforward, because their graph is a directed acyclic graph. Networks with cycles are commonly called recurrent. Such networks are commonly depicted in the manner shown at the top of the figure, where f is shown as being dependent upon itself. However, there is an implied temporal dependence which is not shown.

Learning

What has attracted the most interest in neural networks is the possibility of *learning*. Given a specific task to solve, and a class of functions F , learning means using a set of observations to find $f^* \in F$ which solves the task in some optimal sense.

This entails defining a cost function $C : F \rightarrow \mathbb{R}$ such that, for the optimal solution f^* , $C(f^*) \leq C(f) \forall f \in F$ (i.e., no solution has a cost less than the cost of the optimal solution).

The cost function C is an important concept in learning, as it is a measure of how far away a particular solution is from an optimal solution to the problem to be solved. Learning algorithms search through the solution space to find a function that has the smallest possible cost.

For applications where the solution is dependent on some data, the cost must necessarily be a function of the

observations, otherwise we would not be modelling anything related to the data. It is frequently defined as a statistic to which only approximations can be made. As a simple example consider the problem of finding the

model f which minimizes $C = E \left[(f(x) - y)^2 \right]$, for data pairs (x,y) drawn from some distribution \mathcal{D} . In practical situations we would only have N samples from \mathcal{D} and thus, for the above example, we would only

minimize $\hat{C} = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2$. Thus, the cost is minimized over a sample of the data rather than the entire data set.

When $N \rightarrow \infty$ some form of online machine learning must be used, where the cost is partially minimized as each new example is seen. While online machine learning is often used when \mathcal{D} is fixed, it is most useful in the case where the distribution changes slowly over time. In neural network methods, some form of online machine learning is frequently used for finite datasets.

Choosing a cost function

While it is possible to define some arbitrary, ad hoc cost function, frequently a particular cost will be used, either because it has desirable properties (such as convexity) or because it arises naturally from a particular formulation of the problem (e.g., in a probabilistic formulation the posterior probability of the model can be used as an inverse cost). Ultimately, the cost function will depend on the task we wish to perform. The three main categories of learning tasks are overviewed below.

Learning paradigms

There are three major learning paradigms, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning. Usually any given type of network architecture can be employed in any of those tasks.

Supervised learning

In supervised learning, we are given a set of example pairs $(x, y), x \in X, y \in Y$ and the aim is to find a function $f : X \rightarrow Y$ in the allowed class of functions that matches the examples. In other words, we wish to infer the mapping implied by the data; the cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the mean-squared error which tries to minimize the average squared error between the network's output, $f(x)$, and the target value y over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called Multi-Layer Perceptrons, one obtains the common and well-known backpropagation algorithm for training neural networks.

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher," in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

Unsupervised learning

In unsupervised learning we are given some data x and the cost function to be minimized, that can be any function of the data x and the network's output, f .

The cost function is dependent on the task (what we are trying to model) and our *a priori* assumptions (the implicit properties of our model, its parameters and the observed variables).

As a trivial example, consider the model $f(x) = a$, where a is a constant and the cost $C = E[(x - f(x))^2]$. Minimizing this cost will give us a value of a that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between x and y , whereas in statistical modelling, it could be related to the posterior probability of the model given the data. (Note that in both of those examples those quantities would be maximized rather than minimized).

Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

Reinforcement learning

In reinforcement learning, data x are usually not given, but generated by an agent's interactions with the

environment. At each point in time t , the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t , according to some (usually unknown) dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost; i.e., the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated.

More formally, the environment is modeled as a Markov decision process (MDP) with states $s_1, \dots, s_n \in S$ and actions $a_1, \dots, a_m \in A$ with the following probability distributions: the instantaneous cost distribution $P(c_t | s_t)$, the observation distribution $P(x_t | s_t)$ and the transition $P(s_{t+1} | s_t a_t)$, while a policy is defined as conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the policy that minimizes the cost; i.e., the MC for which the cost is minimal.

ANNs are frequently used in reinforcement learning as part of the overall algorithm.

Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

Learning algorithms

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

Most of the algorithms used in training artificial neural networks employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

Evolutionary methods, simulated annealing, expectation-maximization and non-parametric methods are some commonly used methods for training neural networks. See also machine learning.

Temporal perceptual learning relies on finding temporal relationships in sensory signal streams. In an environment, statistically salient temporal correlations can be found by monitoring the arrival times of sensory signals. This is done by the perceptual network.

Employing artificial neural networks

Perhaps the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism which 'learns' from observed data. However, using them is not so straightforward and a relatively good understanding of the underlying theory is essential.

- Choice of model: This will depend on the data representation and the application. Overly complex models tend to lead to problems with learning.
- Learning algorithm: There are numerous tradeoffs between learning algorithms. Almost any algorithm will work well with the *correct hyperparameters* for training on a particular fixed dataset. However selecting and tuning an algorithm for training on unseen data requires a significant amount of experimentation.
- Robustness: If the model, cost function and learning algorithm are selected appropriately the resulting ANN can be extremely robust.

With the correct implementation ANNs can be used naturally in online learning and large dataset applications. Their simple implementation and the existence of mostly local dependencies exhibited in the structure allows for fast, parallel implementations in hardware.

Applications

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

Real life applications

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control), quantum chemistry,^[2] game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Neural network software

Neural network software is used to simulate, research, develop and apply artificial neural networks, biological neural networks and in some cases a wider array of adaptive systems.

Types of neural networks

Feedforward neural network

The feedforward neural network was the first and arguably simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

Radial basis function (RBF) network

Radial Basis Functions are powerful techniques for interpolation in multidimensional space. A RBF is a function which has built into a distance criterion with respect to a center. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in Multi-Layer Perceptrons. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability. Performance in both cases is often improved by shrinkage techniques, known as ridge regression in classical statistics and known to correspond to a prior belief in small parameter values (and therefore smooth output functions) in a Bayesian framework.

RBF networks have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptrons. This is because the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Linearity ensures that the error surface is quadratic and therefore has a single easily found minimum. In regression problems this can be found in one matrix operation. In classification problems the fixed non-linearity introduced by the sigmoid output function is most efficiently dealt with using iteratively re-weighted least squares.

RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centres are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. A common solution is to associate each data point with its own centre, although this can make the linear system to be solved in the final layer rather large, and requires shrinkage techniques to avoid overfitting.

Associating each input datum with an RBF leads naturally to kernel methods such as Support Vector Machines and Gaussian Processes (the RBF is the kernel function). All three approaches use a non-linear kernel function to project the input data into a space where the learning problem can be solved using a linear model. Like Gaussian Processes, and unlike SVMs, RBF networks are typically trained in a Maximum Likelihood framework by maximizing the probability (minimizing the error) of the data under the model. SVMs take a different approach to avoiding overfitting by maximizing instead a margin. RBF networks are outperformed in most classification applications by SVMs. In regression applications they can be competitive when the dimensionality of the input space is relatively small.

Kohonen self-organizing network

The self-organizing map (SOM) invented by Teuvo Kohonen performs a form of unsupervised learning. A set of artificial neurons learn to map points in an input space to coordinates in an output space. The input space can have different dimensions and topology from the output space, and the SOM will attempt to preserve these.

Recurrent network

Contrary to feedforward networks, recurrent neural networks (RNs) are models with bi-directional data flow. While a feedforward network propagates data linearly from input to output, RNs also propagate data from later

processing stages to earlier stages.

Simple recurrent network

A *simple recurrent network* (SRN) is a variation on the Multi-Layer Perceptron, sometimes called an "Elman network" due to its invention by Jeff Elman. A three-layer network is used, with the addition of a set of "context units" in the input layer. There are connections from the middle (hidden) layer to these context units fixed with a weight of one. At each time step, the input is propagated in a standard feed-forward fashion, and then a learning rule (usually back-propagation) is applied. The fixed back connections result in the context units always maintaining a copy of the previous values of the hidden units (since they propagate over the connections before the learning rule is applied). Thus the network can maintain a sort of state, allowing it to perform such tasks as sequence-prediction that are beyond the power of a standard Multi-Layer Perceptron.

In a *fully recurrent network*, every neuron receives inputs from every other neuron in the network. These networks are not arranged in layers. Usually only a subset of the neurons receive external inputs in addition to the inputs from all the other neurons, and another disjunct subset of neurons report their output externally as well as sending it to all the neurons. These distinctive inputs and outputs perform the function of the input and output layers of a feed-forward or simple recurrent network, and also join all the other neurons in the recurrent processing.

Hopfield network

The Hopfield network is a recurrent neural network in which all connections are symmetric. Invented by John Hopfield in 1982, this network guarantees that its dynamics will converge. If the connections are trained using Hebbian learning then the Hopfield network can perform as robust content-addressable (or associative) memory, resistant to connection alteration.

Echo state network

The echo state network (ESN) is a recurrent neural network with a sparsely connected random hidden layer. The weights of output neurons are the only part of the network that can change and be learned. ESN are good to (re)produce temporal patterns.

Long short term memory network

The Long short term memory is an artificial neural net structure that unlike traditional RNNs doesn't have the problem of vanishing gradients. It can therefore use long delays and can handle signals that have a mix of low and high frequency components.

Stochastic neural networks

A stochastic neural network differs from a typical neural network because it introduces random variations into the network. In a probabilistic view of neural networks, such random variations can be viewed as a form of statistical sampling, such as Monte Carlo sampling.

Boltzmann machine

The Boltzmann machine can be thought of as a noisy Hopfield network. Invented by Geoff Hinton and Terry Sejnowski in 1985, the Boltzmann machine is important because it is one of the first neural networks to demonstrate learning of latent variables (hidden units). Boltzmann machine learning was at first slow to simulate, but the contrastive divergence algorithm of Geoff Hinton (circa 2000) allows models such as Boltzmann machines and *products of experts* to be trained much faster.

Modular neural networks

Biological studies have shown that the human brain functions not as a single massive network, but as a collection of small networks. This realization gave birth to the concept of modular neural networks, in which several small networks cooperate or compete to solve problems.

Committee of machines

A committee of machines (CoM) is a collection of different neural networks that together "vote" on a given example. This generally gives a much better result compared to other neural network models. Because neural networks suffer from local minima, starting with the same architecture and training but using different initial random weights often gives vastly different networks^[citation needed]. A CoM tends to stabilize the result.

The CoM is similar to the general machine learning *bagging* method, except that the necessary variety of machines in the committee is obtained by training from different random starting weights rather than training on different

randomly selected subsets of the training data.

Associative neural network (ASNN)

The ASNN is an extension of the *committee of machines* that goes beyond a simple/weighted average of different models. ASNN represents a combination of an ensemble of feed-forward neural networks and the k-nearest neighbor technique (kNN). It uses the correlation between ensemble responses as a measure of **distance** amid the analyzed cases for the kNN. This corrects the bias of the neural network ensemble. An associative neural network has a memory that can coincide with the training set. If new data become available, the network instantly improves its predictive ability and provides data approximation (self-learn the data) without a need to retrain the ensemble. Another important feature of ASNN is the possibility to interpret neural network results by analysis of correlations between data cases in the space of models. The method is demonstrated at www.vcllab.org, where you can either use it online or download it.

Physical neural network

A physical neural network includes electrically adjustable resistance material to simulate artificial synapses. Examples include the ADALINE neural network developed by Bernard Widrow in the 1960's and the memristor based neural network developed by Greg Snider of HP Labs in 2008.

Other types of networks

These special networks do not fit in any of the previous categories.

Holographic associative memory

Holographic associative memory represents a family of analog, correlation-based, associative, stimulus-response memories, where information is mapped onto the phase orientation of complex numbers operating.

Instantaneously trained networks

Instantaneously trained neural networks (ITNNs) were inspired by the phenomenon of short-term learning that seems to occur instantaneously. In these networks the weights of the hidden and the output layers are mapped directly from the training vector data. Ordinarily, they work on binary data, but versions for continuous data that require small additional processing are also available.

Spiking neural networks

Spiking neural networks (SNNs) are models which explicitly take into account the timing of inputs. The network input and output are usually represented as series of spikes (delta function or more complex shapes). SNNs have an advantage of being able to process information in the time domain (signals that vary over time). They are often implemented as recurrent networks. SNNs are also a form of pulse computer.

Spiking neural networks with axonal conduction delays exhibit polychronization, and hence could have a very large memory capacity.^[3]

Networks of spiking neurons — and the temporal correlations of neural assemblies in such networks — have been used to model figure/ground separation and region linking in the visual system (see, for example, Reitboeck et al. in Haken and Stadler: *Synergetics of the Brain*. Berlin, 1989).

In June 2005 IBM announced construction of a Blue Gene supercomputer dedicated to the simulation of a large recurrent spiking neural network.^[4]

Gerstner and Kistler have a freely available online textbook on Spiking Neuron Models.

Dynamic neural networks

Dynamic neural networks not only deal with nonlinear multivariate behaviour, but also include (learning of) time-dependent behaviour such as various transient phenomena and delay effects.

Cascading neural networks

Cascade-Correlation is an architecture and supervised learning algorithm developed by Scott Fahlman and Christian Lebiere. Instead of just adjusting the weights in a network of fixed topology, Cascade-Correlation begins with a minimal network, then automatically trains and adds new hidden units one by one, creating a multi-layer structure. Once a new hidden unit has been added to the network, its input-side weights are frozen. This unit then becomes a permanent feature-detector in the network, available for producing outputs or for creating other, more complex feature detectors. The Cascade-Correlation architecture has several advantages over existing algorithms: it learns very quickly, the network determines its own size and topology, it retains the structures it has built even if the training set changes, and it requires no back-propagation of error signals through the connections of the network.

See: Cascade correlation algorithm.

Neuro-fuzzy networks

A neuro-fuzzy network is a fuzzy inference system in the body of an artificial neural network. Depending on the FIS type, there are several layers that simulate the processes involved in a *fuzzy inference* like fuzzification, inference, aggregation and defuzzification. Embedding an FIS in a general structure of an ANN has the benefit of using available ANN training methods to find the parameters of a fuzzy system.

Compositional pattern-producing networks

Compositional pattern-producing networks (CPPNs) are a variation of ANNs which differ in their set of activation functions and how they are applied. While typical ANNs often contain only sigmoid functions (and sometimes Gaussian functions), CPPNs can include both types of functions and many others. Furthermore, unlike typical ANNs, CPPNs are applied across the entire space of possible inputs so that they can represent a complete image. Since they are compositions of functions, CPPNs in effect encode images at infinite resolution and can be sampled for a particular display at whatever resolution is optimal.

One-shot associative memory

This type of network can add new patterns without the need for re-training. It is done by creating a specific memory structure, which assigns each new pattern to an orthogonal plane using adjacently connected hierarchical arrays^[5]. The network offers real-time pattern recognition and high scalability, it however requires parallel processing and is thus best suited for platforms such as Wireless sensor networks (WSN), Grid computing, and GPGPUs.

Theoretical properties

Computational power

The multi-layer perceptron (MLP) is a universal function approximator, as proven by the Cybenko theorem. However, the proof is not constructive regarding the number of neurons required or the settings of the weights.

Work by Hava Siegelmann and Eduardo D. Sontag has provided a proof that a specific recurrent architecture with rational valued weights (as opposed to the commonly used floating point approximations) has the full power of a Universal Turing Machine^[6] using a finite number of neurons and standard linear connections. They have further shown that the use of irrational values for weights results in a machine with super-Turing power.

Capacity

Artificial neural network models have a property called 'capacity', which roughly corresponds to their ability to model any given function. It is related to the amount of information that can be stored in the network and to the notion of complexity.

Convergence

Nothing can be said in general about convergence since it depends on a number of factors. Firstly, there may exist many local minima. This depends on the cost function and the model. Secondly, the optimization method used might not be guaranteed to converge when far away from a local minimum. Thirdly, for a very large amount of data or parameters, some methods become impractical. In general, it has been found that theoretical guarantees regarding convergence are an unreliable guide to practical application.

Generalisation and statistics



In applications where the goal is to create a system that generalises well in unseen examples, the problem of overtraining has emerged.

Confidence analysis of a neural network

This arises in overcomplex or overspecified systems when the capacity of the network significantly exceeds the needed free parameters. There are two schools of thought for avoiding this problem: The first is to use cross-validation and similar techniques to check for the presence of overtraining and optimally select hyperparameters such as to minimize the generalisation error. The second is to use some form of *regularisation*. This is a concept that emerges naturally in a

probabilistic (Bayesian) framework, where the regularisation can be performed by selecting a larger prior probability over simpler models; but also in statistical learning theory, where the goal is to minimize over two quantities: the 'empirical risk' and the 'structural risk', which roughly correspond to the error over the training set and the predicted error in unseen data due to overfitting.

Supervised neural networks that use an MSE cost function can use formal statistical methods to determine the confidence of the trained model. The MSE on a validation set can be used as an estimate for variance. This value can then be used to calculate the confidence interval of the output of the network, assuming a normal distribution. A confidence analysis made this way is statistically valid as long as the output probability distribution stays the same and the network is not modified.

By assigning a softmax activation function on the output layer of the neural network (or a softmax component in a component-based neural network) for categorical target variables, the outputs can be interpreted as posterior probabilities. This is very useful in classification as it gives a certainty measure on classifications.

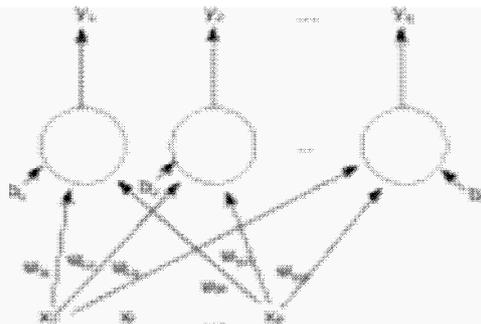
$$y_i = \frac{e^{x_i}}{\sum_{j=1}^c e^{x_j}}$$

The softmax activation function is:

Dynamic properties

Various techniques originally developed for studying disordered magnetic systems (i.e., the spin glass) have been successfully applied to simple neural network architectures, such as the Hopfield network. Influential work by E. Gardner and B. Derrida has revealed many interesting properties about perceptrons with real-valued synaptic weights, while later work by W. Krauth and M. Mezard has extended these principles to binary-valued synapses.

Gallery

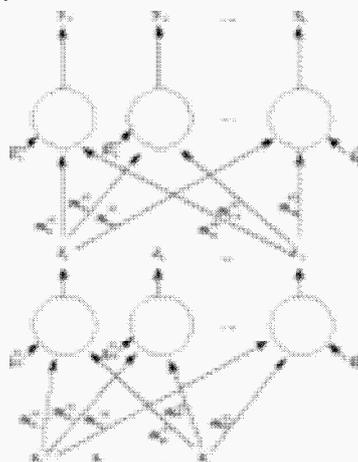


A single-layer feedforward artificial neural network.

Arrows originating from x_2 are omitted for clarity. There are p inputs to this network and q outputs. There is no activation function (or equivalently, the activation function is $g(x) = x$). In this system, the value of the q th output,

$$y_q \text{ would be calculated as } y_q = \sum (x_i * w_{iq})$$

A two-layer feedforward artificial neural network.



See also

- Adaptive resonance theory
- Artificial life
- Associative memory
- Autoencoder
- Biological neural network
- Biologically inspired computing
- Blue brain
- Clinical decision support system
- Connectionist expert system
- Decision tree
- Expert system
- Fuzzy logic
- Genetic algorithm
- In Situ Adaptive Tabulation
- Linear discriminant analysis
- Logistic regression
- Memristor
- Multilayer perceptron
- Nearest neighbor (pattern recognition)
- Neural network
- Neuroevolution, NeuroEvolution of Augmented Topologies (NEAT)
- Neural network software
- N1000 chip
- Optical neural network
- Particle swarm optimization
- Perceptron
- Predictive analytics
- Principal components analysis
- Regression analysis
- Simulated annealing
- Systolic array
- Time delay neural network (TDNN)

References

1. ^ "The Machine Learning Dictionary". <http://www.cse.unsw.edu.au/~billw/mldict.html#activnfn>.
2. ^ Roman M. Balabin, Ekaterina I. Lomakina (2009). "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies". *J. Chem. Phys.* **131** (7): 074104. doi:10.1063/1.3206326.
3. ^ Izhikevich EM (February 2006). "Polychronization: computation with spikes". *Neural Comput* **18** (2): 245–82. doi:10.1162/089976606775093882. PMID 16378515 16378515.
4. ^ "IBM Research | Press Resources | IBM and EPFL Join Forces to Uncover the Secrets of Cognitive Intelligence". http://domino.research.ibm.com/comm/pr.nsf/pages/news.20050606_CognitiveIntelligence.html. Retrieved 2009-05-02.
5. ^ B.B. Nasution, A.I. Khan, A Hierarchical Graph Neuron Scheme for Real-Time Pattern Recognition, IEEE Transactions on Neural Networks, vol 19(2), 212-229, Feb. 2008
6. ^ Siegelmann, H.T.; Sontag, E.D. (1991). "Turing computability with neural nets". *Appl. Math. Lett.* **4** (6): 77–80. http://www.math.rutgers.edu/~sontag/FTP_DIR/aml-turing.pdf.

External links

- Performance comparison of neural network algorithms tested on UCI data sets
- A close view to Artificial Neural Networks Algorithms
- Neural Networks at the Open Directory Project
- A Brief Introduction to Neural Networks (D. Kriesel) - Illustrated, bilingual manuscript about artificial neural networks; Topics so far: Perceptrons, Backpropagation, Radial Basis Functions, Recurrent Neural Networks, Self Organizing Maps, Hopfield Networks.
- Neural Networks in Materials Science
- A practical tutorial on Neural Networks
- Applications of neural networks

Further reading

- Dedicated issue of *Philosophical Transactions B* on Neural Networks and Perception. Some articles are freely available.

Bibliography

- Bar-Yam, Yaneer (2003). *Dynamics of Complex Systems, Chapter 2, 3*.
- Bar-Yam, Yaneer (2005). *Making Things Work*. Please see Chapter 3
- Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science". *ISIJ International* **39**: 966–979. doi:10.2355/isijinternational.39.966.
- Bhagat, P.M. (2005) *Pattern Recognition in Industry*, Elsevier. ISBN 0-08-044538-1
- Bishop, C.M. (1995) *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press. ISBN 0-19-

- 853849-9 (hardback) or ISBN 0-19-853864-2 (paperback)
- Cybenko, G.V. (1989). Approximation by Superpositions of a Sigmoidal function, *Mathematics of Control, Signals and Systems*, Vol. 2 pp. 303–314. electronic version
 - Duda, R.O., Hart, P.E., Stork, D.G. (2001) *Pattern classification (2nd edition)*, Wiley, ISBN 0-471-05669-3
 - Egmont-Petersen, M., de Ridder, D., Handels, H. (2002). "Image processing with neural networks - a review". *Pattern Recognition* 35 (10): 2279–2301. doi:10.1016/S0031-3203(01)00178-9+.
 - Gurney, K. (1997) *An Introduction to Neural Networks* London: Routledge. ISBN 1-85728-673-1 (hardback) or ISBN 1-85728-503-4 (paperback)
 - Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN 0-13-273350-1
 - Fahlman, S, Lebiere, C (1991). *The Cascade-Correlation Learning Architecture*, created for National Science Foundation, Contract Number EET-8716324, and Defense Advanced Research Projects Agency (DOD), ARPA Order No. 4976 under Contract F33615-87-C-1499. electronic version
 - Hertz, J., Palmer, R.G., Krogh. A.S. (1990) *Introduction to the theory of neural computation*, Perseus Books. ISBN 0-201-51560-1
 - Lawrence, J. (1994) *Introduction to Neural Networks*, California Scientific Software Press. ISBN 1-883157-00-5
 - Masters, T. (1994) *Signal and Image Processing with Neural Networks*, J. Wiley & Sons, Inc. ISBN 0-471-04963-8
 - Ness, Erik. 2005. SPIDA-Web. *Conservation in Practice* 6(1):35-36. On the use of artificial neural networks in species taxonomy.
 - Ripley, Brian D. (1996) *Pattern Recognition and Neural Networks*, Cambridge
 - Siegelmann, H.T. and Sontag, E.D. (1994). Analog computation via neural networks, *Theoretical Computer Science*, v. 131, no. 2, pp. 331–360. electronic version
 - Sergios Theodoridis, Konstantinos Koutroumbas (2009) "Pattern Recognition" , 4th Edition, Academic Press, ISBN: 978-1-59749-272-0.
 - Smith, Murray (1993) *Neural Networks for Statistical Modeling*, Van Nostrand Reinhold, ISBN 0-442-01310-8
 - Wasserman, Philip (1993) *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, ISBN 0-442-00461-3

16. Perceptron

The **perceptron** is a type of artificial neural network invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt. It can be seen as the simplest kind of feedforward neural network: a linear classifier.

Definition

The Perceptron is a binary classifier that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value) across the matrix.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{else} \end{cases}$$

where w is a vector of real-valued weights and $w \cdot x$ is the dot product (which computes a weighted sum). b is the 'bias', a constant term that does not depend on any input value.

The value of $f(x)$ (0 or 1) is used to classify x as either a positive or a negative instance, in the case of a binary classification problem. The bias can be thought of as offsetting the activation function, or giving the output neuron a "base" level of activity. If b is negative, then the weighted combination of inputs must produce a positive value greater than $|b|$ in order to push the classifier neuron over the 0 threshold. Spatially, the bias alters the position (though not the orientation) of the decision boundary.

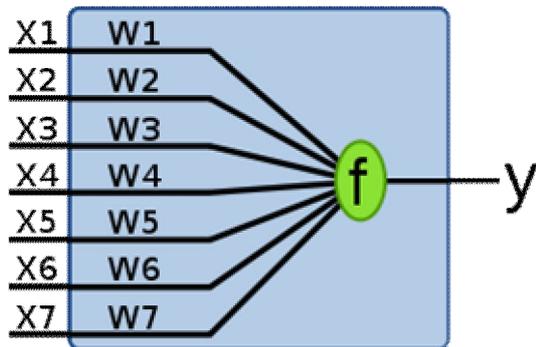
Since the inputs are fed directly to the output unit via the weighted connections, the perceptron can be considered the simplest kind of feed-forward neural network.

Learning algorithm

The learning algorithm is the same across all neurons, therefore everything that follows is applied to a single neuron in isolation. We first define some variables:

- $x(j)$ denotes the j -th item in the n -dimensional input vector
- $w(j)$ denotes the j -th item in the weight vector
- $f(x)$ denotes the output from the neuron when presented with input x
- α is a constant where $0 < \alpha \leq 1$ (learning rate)

Further, assume for convenience that the bias term b is zero. This is not a restriction since an extra dimension $n + 1$ can be added to the input vectors x with $x(n + 1) = 1$, in which case $w(n + 1)$ replaces the bias term.



The appropriate weights are applied to the inputs, and the resulting weighted sum passed to a function which produces the output y .

Learning is modeled as the weight vector being updated for multiple iterations over all training examples. Let $D_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ denote a training set of m training examples, where x_i is the input vector to the perceptron and y_i is the desired output value of the perceptron for that input vector.

Each iteration the weight vector is updated as follows:

For each (x, y) pair in

$$D_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$$

$$w_{t+1}(j) = w_t(j) + \alpha(y - f(x))x(j) \quad (j = 1, \dots, n)$$

Note that this means that a change in the weight vector will only take place for a given training example (x, y) if its output $f(x)$ is different from the desired output y .

The initialization of w is usually performed simply by setting $w(j) = 0$ for all elements $w(j)$.

Separability and Convergence

The training set D_m is said to be linearly separable if there exists a positive constant γ and a weight vector w such that $y_i \cdot (\langle w, x_i \rangle + b) > \gamma$ for all i . That is, if we say that w is the weight vector to the perceptron, then the output of the perceptron, $\langle w, x_i \rangle + b$, multiplied by the desired output of the perceptron, y_i , must be greater than the positive constant, γ , for all input-vector/output-value pairs (x_i, y_i) in D_m .

Novikoff (1962) proved that the perceptron algorithm converges after a finite number of iterations if the data set is linearly separable. The idea of the proof is that the weight vector is always adjusted by a bounded amount in a

direction that it has a negative dot product with, and thus can be bounded above by $O(\sqrt{t})$, where t is the number of changes to the weight vector, but it can also be bounded below by $O(t)$ because if there exists an (unknown) satisfactory weight vector, then every change makes progress in this (unknown) direction by a positive amount that depends only on the input vector. This can be used to show that the number of mistakes (changes to the weight vector, i.e. t) is bounded by $(2R / \gamma)^2$ where R is the maximum norm of an input vector. However, if the training set is not linearly separable, the above online algorithm will not converge.

Note that the decision boundary of a perceptron is invariant with respect to scaling of the weight vector, i.e. a perceptron trained with initial weight vector w and learning rate α is an identical estimator to a perceptron trained with initial weight vector w / α and learning rate 1. Thus, since the initial weights become irrelevant with increasing number of iterations, the learning rate does not matter in the case of the perceptron and is usually just set to one.

Variants

The pocket algorithm with ratchet (Gallant, 1990) solves the stability problem of perceptron learning by keeping the best solution seen so far "in its pocket". The pocket algorithm then returns the solution in the pocket, rather than the last solution.

The α -perceptron further utilised a preprocessing layer of fixed random weights, with thresholded output units. This enabled the perceptron to classify analogue patterns, by projecting them into a binary space. In fact, for a projection space of sufficiently high dimension, patterns can become linearly separable.

As an example, consider the case of having to classify data into two classes. Here is a small such data set, consisting of two points coming from two Gaussian distributions.

A linear classifier can only separate things with a hyperplane, so it's not possible to classify all the examples perfectly. On the other hand, we may project the data into a large number of dimensions. In this case a random matrix was used to project the data linearly to a 1000-dimensional space; then each resulting data point was transformed through the hyperbolic tangent function. A linear classifier can then separate the data, as shown in the third figure. However the data may still not be completely separable in this space, in which the perceptron algorithm would not converge. In the example shown, stochastic steepest gradient descent was used to adapt the parameters.

Furthermore, by adding nonlinear layers between the input and output, one can separate all data and indeed, with enough training data, model any well-defined function to arbitrary precision. This model is a generalization known as a multilayer perceptron.

It should be kept in mind, however, that the best classifier is not necessarily that which classifies all the training data perfectly. Indeed, if we had the prior constraint that the data come from equi-variant Gaussian distributions, the linear separation in the input space is optimal.

Other training algorithms for linear classifiers are possible: see, e.g., support vector machine and logistic regression.

Multiclass perceptron

Like most other techniques for training linear classifiers, the perceptron generalizes naturally to multiclass classification. Here, the input x and the output y are drawn from arbitrary sets. A feature representation function $f(x,y)$ maps each possible input/output pair to a finite-dimensional real-valued feature vector. As before, the feature vector is multiplied by a weight vector w , but now the resulting score is used to choose among many possible outputs:

$$\hat{y} = \operatorname{argmax}_y f(x, y) \cdot w$$

Learning again iterates over the examples, predicting an output for each, leaving the weights unchanged when the predicted output matches the target, and changing them when it does not. The update becomes:

$$w_{t+1} = w_t + f(x, y) - f(x, \hat{y})$$

This multiclass formulation reduces to the original perceptron when x is a real-valued vector, y is chosen from $\{0,1\}$, and $f(x,y) = yx$.

For certain problems, input/output representations and features can be chosen so that $\operatorname{argmax}_y f(x, y) \cdot w$ can be found efficiently even though y is chosen from a very large or even infinite set.

In recent years, perceptron training has become popular in the field of natural language processing for such tasks as part-of-speech tagging and syntactic parsing (Collins, 2002).

History

Although the perceptron initially seemed promising, it was eventually proved that perceptrons could not be trained to recognise many classes of patterns. This led to the field of neural network research stagnating for many years, before it was recognised that a feedforward neural network with two or more layers (also called a multilayer perceptron) had far greater processing power than perceptrons with one layer (also called a single layer perceptron). Single layer perceptrons are only capable of learning linearly separable patterns; in 1969 a famous book entitled **Perceptrons** by Marvin Minsky and Seymour Papert showed that it was impossible for these classes of network to learn an XOR function. They conjectured (incorrectly) that a similar result would hold for a perceptron with three or more layers. Three years later Stephen Grossberg published a series of papers introducing networks capable of modelling differential, contrast-enhancing and XOR functions. (The papers were published in 1972 and 1973, see e.g.: Grossberg, Contour enhancement, short-term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics*, 52 (1973), 213-257, online [1]). Nevertheless the often-cited Minsky/Papert text caused a significant decline in interest and funding of neural network research. It took ten more years until neural network research experienced a resurgence in the 1980s. This text was reprinted in 1987 as "Perceptrons - Expanded Edition" where some errors in the original text are shown and corrected.

More recently, interest in the perceptron learning algorithm has increased again after Freund and Schapire (1998) presented a voted formulation of the original algorithm (attaining large margin) and suggested that one can apply the kernel trick to it.

References

- Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408. doi:10.1037/h0042519.
- Minsky M. L. and Papert S. A. 1969. *Perceptrons*. Cambridge, MA: MIT Press.
- Freund, Y. and Schapire, R. E. 1998. Large margin classification using the perceptron algorithm. In Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT' 98). ACM Press.
- Freund, Y. and Schapire, R. E. 1999. Large margin classification using the perceptron algorithm. In *Machine Learning* 37(3):277-296, 1999.
- Gallant, S. I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, vol. 1, no. 2,

pp. 179-191.

- Novikoff, A. B. (1962). On convergence proofs on perceptrons. Symposium on the Mathematical Theory of Automata, 12, 615-622. Polytechnic Institute of Brooklyn.
- Widrow, B., Lehr, M.A., "30 years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proc. IEEE*, vol 78, no 9, pp. 1415-1442, (1990).
- Collins, M. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with the perceptron algorithm in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '02)
- Yin, Hongfeng (1996), Perceptron-Based Algorithms and Analysis, Spectrum Library, Concordia University, Canada

External links

- SergeiAlderman-ANN.rtf
- Chapter 3 Weighted networks - the perceptron and chapter 4 Perceptron learning of *Neural Networks - A Systematic Introduction* by Raúl Rojas (ISBN 978-3540605058)
- Pithy explanation of the update rule by Charles Elkan
- C# implementation of a perceptron
- History of perceptrons
- Mathematics of perceptrons

Perceptron demo applet and an introduction by examples

~~~~~

Kiadja a TCC COMPUTER STÚDIÓ, az 1988-ban alapított TCC-GROUP tagja, 2000 Szentendre, Pannónia u. 11.

tel: +36-26-314-590, e-mail: [tcc66@hotmail.com](mailto:tcc66@hotmail.com).

A kiadásért felel a Stúdió igazgatója, felelős szerkesztő Terjék Aranka.

Nyomta és kötötte a CORBIS DIGITAL – digitális és offszet nyomtatás,  
1033 Budapest, Szentendrei út 89-93. PP Center, III. épület (rózsaszín) , I.emelet

tel: +36-1-999-6593, e-mail: [corbiskft@gmail.com](mailto:corbiskft@gmail.com), [www.corbis-digital.hu](http://www.corbis-digital.hu),

Nyomtatásért felel a nyomda igazgatója.

Megjelent 8 (A/4) ív terjedelemben.

ISBN 978-963-xx-xxxx-x

