

Artificial Neural Networks

Wikibooks.org

March 14, 2013

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. An URI to this license is given in the list of figures on page 87. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 85. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 91, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 87. This PDF was generated by the L^AT_EX typesetting software. The L^AT_EX source code is included as an attachment (`source.7z.txt`) in this PDF file. To extract the source from the PDF file, we recommend the use of <http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/> utility or clicking the paper clip attachment symbol on the lower left of your PDF Viewer, selecting **Save Attachment**. After extracting it from the PDF file you have to rename it to `source.7z`. To uncompress the resulting archive we recommend the use of <http://www.7-zip.org/>. The L^AT_EX source itself was generated by a program written by Dirk Hünninger, which is freely available under an open source license from http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf. This distribution also contains a configured version of the `pdflatex` compiler with all necessary packages and fonts needed to compile the L^AT_EX source included in this PDF file.

Contents

1	Introduction	3
1.1	Introduction	3
1.2	What is This Book About?	3
1.3	Who is This Book For?	3
1.4	What Are The Prerequisites?	3
2	Neural Network Basics	5
2.1	Artificial Neural Networks	5
2.2	Processing Elements	6
2.3	Why Use Neural Nets?	7
2.4	Learning	8
2.5	Network Parameters	9
3	Biological Neural Networks	13
3.1	Biological Neural Nets	13
3.2	Neurons	14
3.3	Biological Networks	14
4	History	15
4.1	Early History	15
4.2	Artificial Neural Networking	15
5	MATLAB Neural Networking Toolbox	17
5.1	MATLAB	17
5.2	Other Neural Network Software	17
6	Activation Functions	19
6.1	Activation Functions	19
6.2	Step Function	19
6.3	Linear combination	20
6.4	Continuous Log-Sigmoid Function	21
6.5	Continuous Tan-Sigmoid Function	22
6.6	Softmax Function	22
7	Feed-Forward Networks	23
7.1	Feedforward Systems	23
7.2	Connection Weights	25
7.3	Mathematical Relationships	25
8	Radial Basis Function Networks	27

9	Recurrent Networks	29
9.1	Recurrent Networks	29
9.2	Simple Recurrent Networks	29
10	Echo State Networks	33
11	Hopfield Networks	35
11.1	Hopfield Networks	35
11.2	Energy Function	35
11.3	Associative Memory	35
12	Self-Organizing Maps	37
12.1	Self-Organizing Maps	37
12.2	Neuron Regions	37
13	Competitive Models	39
13.1	Competitive Networks	39
14	ART Models	41
15	Boltzmann Machines	43
16	Committee of Machines	45
17	Learning Paradigms	47
17.1	Learning Paradigms	47
17.2	Supervised Learning	47
17.3	Unsupervised Learning	47
18	Error-Correction Learning	49
18.1	Error-Correction Learning	49
18.2	Gradient Descent	49
18.3	Backpropagation	50
19	Hebbian Learning	53
19.1	Hebbian Learning	53
19.2	Mathematical Formulation	53
19.3	Plausibility	53
20	Competitive Learning	55
20.1	Competitive Learning	55
20.2	Linear Vector Quantization	55
21	Boltzmann Learning	57
22	ART Learning	59
22.1	ART Learning	59
23	Self-Organizing Maps	61
23.1	Self-Organizing Maps	61

23.2 Neuron Regions	61
24 Pattern Recognition	63
25 Clustering	65
26 Feature Detection	67
27 Series Prediction	69
28 Data Compression	71
29 Curve Fitting	73
29.1 Curve Fitting	73
29.2 Function Approximation	73
30 Optimization	75
31 Control	77
31.1 Control Systems	77
31.2 See Also	77
32 Criticisms and Problems	79
33 Artificial Intelligence	81
34 Resources	83
34.1 Wikibooks	83
34.2 Commons	83
35 Contributors	85
List of Figures	87
36 Licenses	91
36.1 GNU GENERAL PUBLIC LICENSE	91
36.2 GNU Free Documentation License	92
36.3 GNU Lesser General Public License	93

1 Introduction

1.1 Introduction

Artificial neural networks are one of the most popular and promising areas of artificial intelligence research. Artificial Neural Networks are abstract computational models, roughly based on the organizational structure of the human brain. There are a wide variety of network architectures and learning methods that can be combined to produce neural networks with different computational abilities.

1.2 What is This Book About?

This book is going to serve as a general-purpose overview of artificial neural networks, including network construction, use, and applications.

1.3 Who is This Book For?

This book is going to be aimed at advanced undergraduates and graduate students in the areas of computer science, mathematics, engineering, and the sciences.

1.4 What Are The Prerequisites?

Readers of this book are going to require a solid mathematical background that includes, but may not be limited to:

- Linear Algebra¹
- Abstract Algebra²
- Calculus³

Students may also find some benefit in the following engineering texts:

- Signals and Systems⁴
- Engineering Analysis⁵

1 <http://en.wikibooks.org/wiki/Linear%20Algebra>
2 <http://en.wikibooks.org/wiki/Abstract%20Algebra>
3 <http://en.wikibooks.org/wiki/Calculus>
4 <http://en.wikibooks.org/wiki/Signals%20and%20Systems>
5 <http://en.wikibooks.org/wiki/Engineering%20Analysis>

Students who wish to implement the lessons learned in this book should be familiar with at least one general-purpose programming language or have a background in:

- MATLAB Programming⁶
- Programmable Logic⁷

6 <http://en.wikibooks.org/wiki/MATLAB%20Programming>

7 <http://en.wikibooks.org/wiki/Programmable%20Logic>

2 Neural Network Basics

2.1 Artificial Neural Networks

Artificial Neural Networks, also known as “Artificial neural nets”, “neural nets”, or ANN for short, are a computational tool modeled on the interconnection of the neuron in the nervous systems of the human brain and that of other organisms. Biological Neural Nets (BNN) are the naturally occurring equivalent of the ANN. Both BNN and ANN are network systems constructed from atomic components known as “neurons”. Artificial neural networks are very different from biological networks, although many of the concepts and characteristics of biological systems are faithfully reproduced in the artificial systems. Artificial neural nets are a type of non-linear processing system that is ideally suited for a wide range of tasks, especially tasks where there is no existing algorithm for task completion. ANN can be trained to solve certain problems using a teaching method and sample data. In this way, identically constructed ANN can be used to perform different tasks depending on the training received. With proper training, ANN are capable of generalization, the ability to recognize similarities among different input patterns, especially patterns that have been corrupted by noise.

2.1.1 What Are Neural Nets?

The term “Neural Net” refers to both the biological and artificial variants, although typically the term is used to refer to artificial systems only. Mathematically, neural nets are nonlinear. Each layer represents a non-linear combination of non-linear functions from the previous layer. Each neuron is a multiple-input, multiple-output (MIMO) system that receives signals from the inputs, produces a resultant signal, and transmits that signal to all outputs. Practically, neurons in an ANN are arranged into layers. The first layer that interacts with the environment to receive input is known as the input layer. The final layer that interacts with the output to present the processed data is known as the output layer. Layers between the input and the output layer that do not have any interaction with the environment are known as hidden layers. Increasing the complexity of an ANN, and thus its computational capacity, requires the addition of more hidden layers, and more neurons per layer.

Biological neurons are connected in very complicated networks. Some regions of the human brain such as the cerebellum are composed of very regular patterns of neurons. Other regions of the brain, such as the cerebrum have less regular arrangements. A typical biological neural system has millions or billions of cells, each with thousands of interconnections with other neurons. Current artificial systems cannot achieve this level of complexity, and so cannot be used to reproduce the behavior of biological systems exactly.

2.2 Processing Elements

In an artificial neural network, neurons can take many forms and are typically referred to as **Processing Elements** (PE) to differentiate them from the biological equivalents. The PE are connected into a particular network pattern, with different patterns serving different functional purposes. Unlike biological neurons with chemical interconnections, the PE in artificial systems are electrical only, and may be either analog, digital, or a hybrid. However, to reproduce the effect of the synapse, the connections between PE are assigned multiplicative weights, which can be calibrated or “trained” to produce the proper system output.

2.2.1 McCulloch-Pitts Model

Processing Elements are typically defined in terms of two equations that represent the McCulloch-Pitts model of a neuron:

[McCulloch-Pitts Model]

$$\zeta = \sum_i w_i x_i$$

$$y = \sigma(\zeta)$$

Where ζ is the weighted sum of the inputs (the inner product of the input vector and the tap-weight vector), and $\sigma(\zeta)$ is a function of the weighted sum. If we recognize that the weight and input elements form vectors \mathbf{w} and \mathbf{x} , the ζ weighted sum becomes a simple dot product:

$$\zeta = \mathbf{w} \cdot \mathbf{x}$$

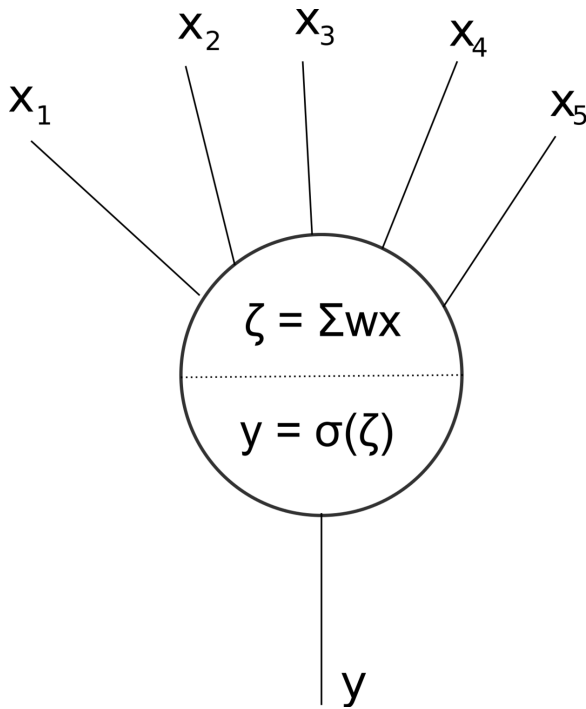


Figure 1

This may be called either the activation function (in the case of a threshold comparison) or a transfer function. The image to the right shows this relationship diagrammatically. The dotted line in the center of the neuron represents the division between the calculation of the input sum using the weight vector, and the calculation of the output value using the activation function. In an actual artificial neuron, this division may not be made explicitly.

The inputs to the network, x , come from an input space and the system outputs are part of the output space. For some networks, the output space Y may be as simple as $\{0, 1\}$, or it may be a complex multi-dimensional space. Neural networks tend to have one input per degree of freedom in the input space, and one output per degree of freedom in the output space.

The tap weight vector is updated during training by various algorithms. One of the more popular of which is the backpropagation algorithm which we will discuss in more detail later.

2.3 Why Use Neural Nets?

Artificial neural nets have a number of properties that make them an attractive alternative to traditional problem-solving techniques. The two main alternatives to using neural nets are to develop an algorithmic solution, and to use an expert system.

Algorithmic methods arise when there is sufficient information about the data and the underlying theory. By understanding the data and the theoretical relationship between the data, we can directly calculate unknown solutions from the problem space. Ordinary von

Neumann computers can be used to calculate these relationships quickly and efficiently from a numerical algorithm.

Expert systems, by contrast, are used in situations where there is insufficient data and theoretical background to create any kind of a reliable problem model. In these cases, the knowledge and rationale of human experts is codified into an expert system. Expert systems emulate the deduction processes of a human expert, by collecting information and traversing the solution space in a directed manner. Expert systems are typically able to perform very well in the absence of an accurate problem model and complete data. However, where sufficient data or an algorithmic solution is available, expert systems are a less than ideal choice.

Artificial neural nets are useful for situations where there is an abundance of data, but little underlying theory. The data, which typically arises through extensive experimentation may be non-linear, non-stationary, or chaotic, and so may not be easily modeled. Input-output spaces may be so complex that a reasonable traversal with an expert system is not a satisfactory option. Importantly, neural nets do not require any a priori assumptions about the problem space, not even information about statistical distribution. Though such assumptions are not required, it has been found that the addition of such a priori information as the statistical distribution of the input space can help to speed training. Many mathematical problem models tend to assume that data lies in a standard distribution pattern, such as Gaussian or Maxwell-Boltzmann distributions. Neural networks require no such assumption. During training, the neural network performs the necessary analytical work, which would require non-trivial effort on the part of the analyst if other methods were to be used.

2.4 Learning

Learning is a fundamental component to an intelligent system, although a precise definition of learning is hard to produce. In terms of an artificial neural network, learning typically happens during a specific training phase. Once the network has been trained, it enters a production phase where it produces results independently. Training can take on many different forms, using a combination of learning paradigms, learning rules, and learning algorithms. A system which has distinct learning and production phases is known as a static network. Networks which are able to continue learning during production use are known as dynamical systems.

A learning paradigm is supervised, unsupervised or a hybrid of the two, and reflects the method in which training data is presented to the neural network. A method that combines supervised and unsupervised training is known as a hybrid method. A learning rule is a model for the types of methods to be used to train the system, and also a goal for what types of results are to be produced. The learning algorithm is the specific mathematical method that is used to update the inter-neuronal synaptic weights during each training iteration. Under each learning rule, there are a variety of possible learning algorithms for use. Most algorithms can only be used with a single learning rule. Learning rules and learning algorithms can typically be used with either supervised or unsupervised learning paradigms, however, and each will produce a different effect.

Overtraining is a problem that arises when too many training examples are provided, and the system becomes incapable of useful generalization. This can also occur when there are too many neurons in the network and the capacity for computation exceeds the dimensionality of the input space. During training, care must be taken not to provide too many input examples and different numbers of training examples could produce very different results in the quality and robustness of the network.

2.5 Network Parameters

There are a number of different parameters that must be decided upon when designing a neural network. Among these parameters are the number of layers, the number of neurons per layer, the number of training iterations, et cetera. Some of the more important parameters in terms of training and network capacity are the number of hidden neurons, the learning rate and the momentum parameter.

2.5.1 Number of neurons in the hidden layer

Hidden neurons are the neurons that are neither in the input layer nor the output layer. These neurons are essentially hidden from view, and their number and organization can typically be treated as a black box to people who are interfacing with the system. Using additional layers of hidden neurons enables greater processing power and system flexibility. This additional flexibility comes at the cost of additional complexity in the training algorithm. Having too many hidden neurons is analogous to a system of equations with more equations than there are free variables: the system is over specified, and is incapable of generalization. Having too few hidden neurons, conversely, can prevent the system from properly fitting the input data, and reduces the robustness of the system.

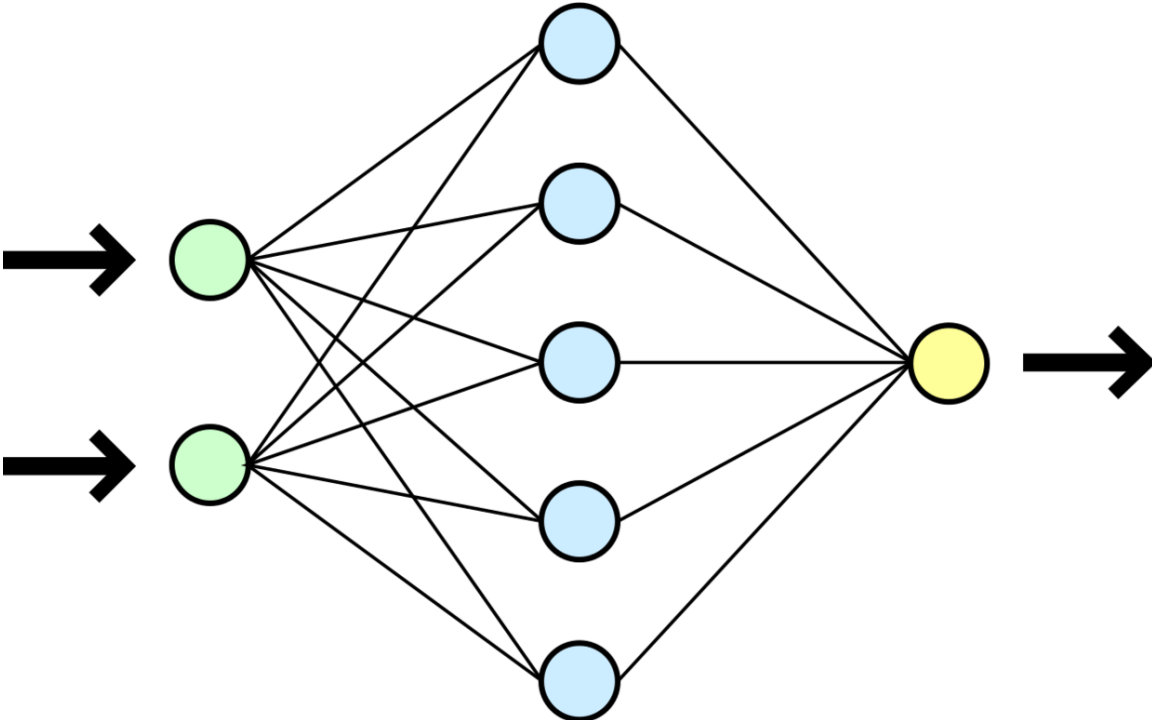


Figure 2

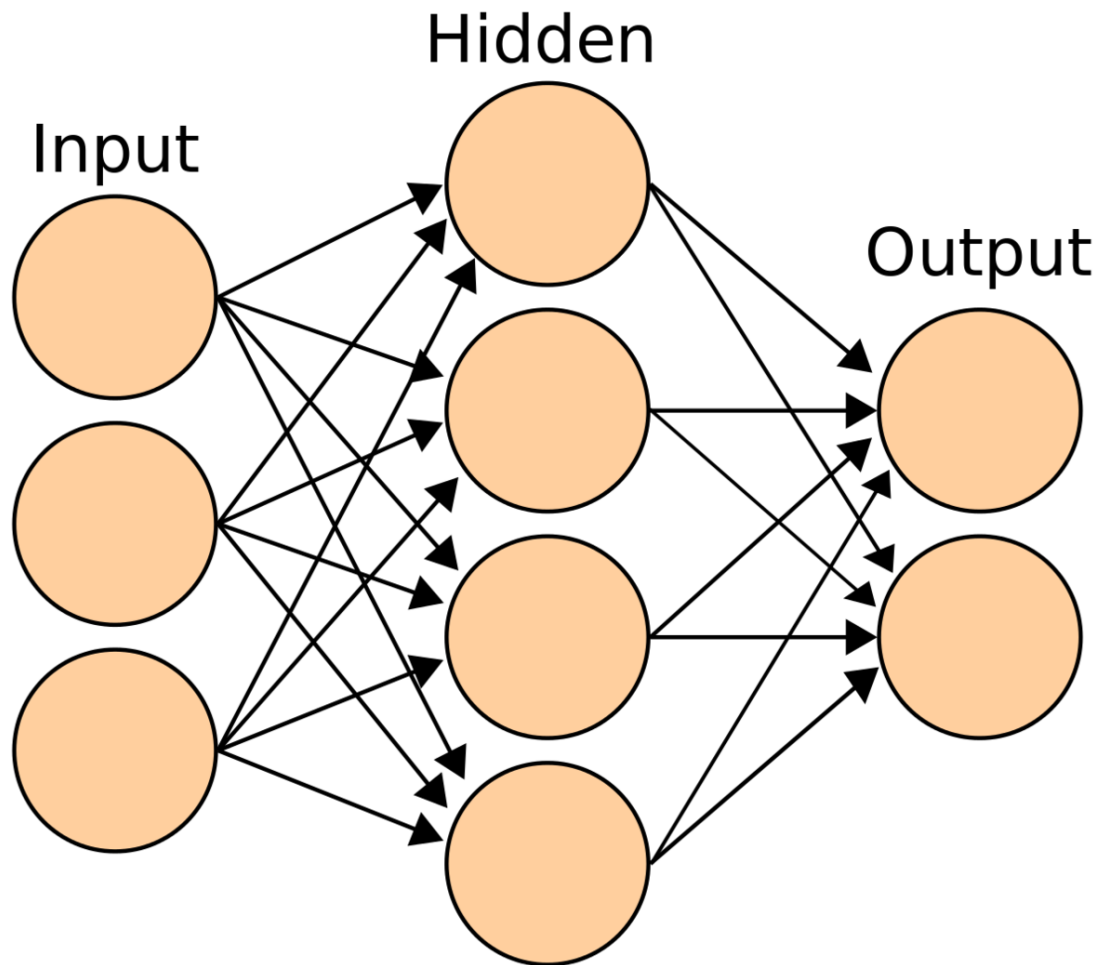


Figure 3

Data type: Integer Domain: $[1, \infty]$ Typical value: 8

Meaning: Number of neurons in the hidden layer (additional layer to the input and output layers, not connected externally).

2.5.2 Learning Rate

Data type: Real Domain: $[0, 1]$ Typical value: 0.3

Meaning: Learning Rate. Training parameter that controls the size of weight and bias changes during learning.

2.5.3 Momentum

Data type: Real Domain: $[0, 1]$ Typical value: 0.05

Meaning: Momentum simply adds a fraction m of the previous weight update to the current one. The momentum parameter is used to prevent the system from converging to a local minimum or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable. A momentum coefficient that is too low cannot reliably avoid local minima, and can also slow down the training of the system.

2.5.4 Training type

Data type: Integer Domain: $[0, 1]$ Typical value: 1

Meaning: 0 = train by epoch, 1 = train by minimum error

2.5.5 Epoch

Data type: Integer Domain: $[1, \infty]$ Typical value: 5000000

Meaning: Determines when training will stop once the number of iterations exceeds epochs. When training by minimum error, this represents the maximum number of iterations.

2.5.6 Minimum Error

Data type: Real Domain: $[0, 0.5]$ Typical value: 0.01

Meaning: Minimum mean square error of the epoch. Square root of the sum of squared differences between the network targets and actual outputs divided by number of patterns (only for training by minimum error).

3 Biological Neural Networks

3.1 Biological Neural Nets

In the case of a biological neural net, neurons are living cells with axons and dendrites that form interconnections through electro-chemical synapses. Signals are transmitted through the cell body (soma), from the dendrite to the axon as an electrical impulse. In the pre-synaptic membrane of the axon, the electrical signal is converted into a chemical signal in the form of various neurotransmitters. These neurotransmitters, along with other chemicals present in the synapse form the message that is received by the post-synaptic membrane of the dendrite of the next cell, which in turn is converted to an electrical signal.

This page is going to provide a brief overview of biological neural networks, but the reader will have to find a better source for a more in-depth coverage of the subject.

3.1.1 Synapses

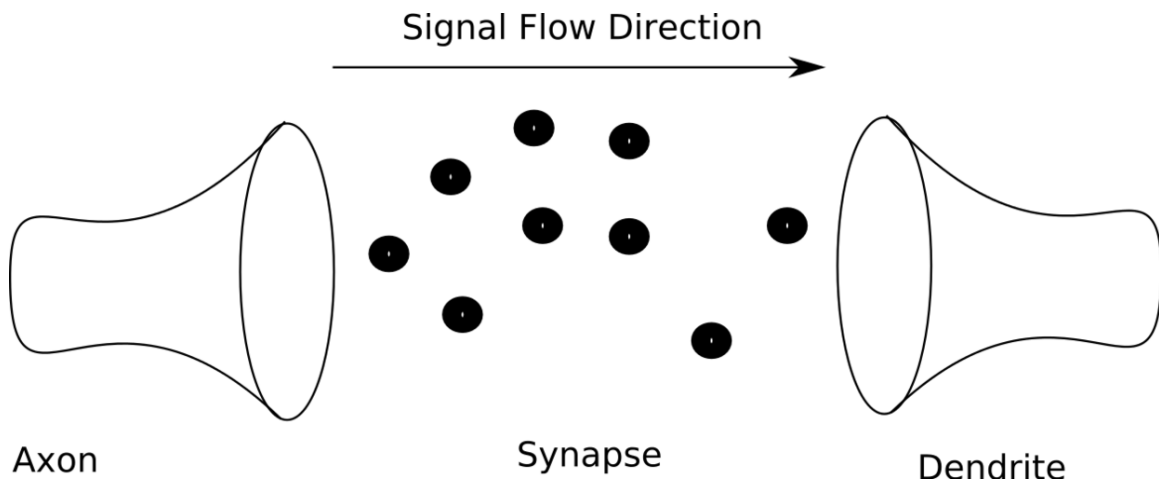


Figure 4

The figure above shows a model of the synapse showing the chemical messages of the synapse moving from the axon to the dendrite. Synapses are not simply a transmission medium for chemical signals, however. A synapse is capable of modifying itself based on the signal traffic that it receives. In this way, a synapse is able to “learn” from its past activity. This learning happens through the strengthening or weakening of the connection. External factors can also affect the chemical properties of the synapse, including body chemistry and medication.

3.2 Neurons

Cells have multiple dendrites, each receives a weighted input. Inputs are weighted by the strength of the synapse that the signal travels through. The total input to the cell is the sum of all such synaptic weighted inputs. Neurons utilize a threshold mechanism, so that signals below a certain threshold are ignored, but signals above the threshold cause the neuron to fire. Neurons follow an “all or nothing” firing scheme, and are similar in this respect to a digital component. Once a neuron has fired, a certain refraction period must pass before it can fire again.

3.3 Biological Networks

Biological neural systems are heterogeneous, in that there are many different types of cells with different characteristics. Biological systems are also characterized by macroscopic order, but nearly random interconnection on the microscopic layer. The random interconnection at the cellular level is rendered into a computational tool by the learning process of the synapse, and the formation of new synapses between nearby neurons.

4 History

4.1 Early History

The history of neural networking arguably started in the late 1800s with scientific attempts to study the workings of the human brain. In 1890, William James published the first work about brain activity patterns. In 1943, McCulloch and Pitts produced a model of the neuron that is still used today in artificial neural networking. This model is broken into two parts: a summation over weighted inputs and an output function of the sum.

4.2 Artificial Neural Networking

In 1949, Donald Hebb published *The Organization of Behavior*, which outlined a law for synaptic neuron learning. This law, later known as Hebbian Learning in honor of Donald Hebb is one of the simplest and most straight-forward learning rules for artificial neural networks.

In 1951, Marvin Minsky created the first ANN while working at Princeton.

In 1958 *The Computer and the Brain* was published posthumously, a year after John von Neumann's death. In that book, von Neumann proposed many radical changes to the way in which researchers had been modeling the brain.

4.2.1 Mark I Perceptron

The Mark I Perceptron was also created in 1958, at Cornell University by Frank Rosenblatt. The Perceptron was an attempt to use neural network techniques for character recognition. The Mark I Perceptron was a linear system, and was useful for solving problems where the input classes were linearly separable in the input space. In 1960, Rosenblatt published the book *Principles of Neurodynamics*, containing much of his research and ideas about modeling the brain.

The Perceptron was a linear system with a simple input-output relationship defined as a McCulloch-Pitts neuron with a step activation function. In this model, the weighted inputs were compared to a threshold θ . The output, y , was defined as a simple step function:

$$y = \begin{cases} 1 & \text{if } \zeta \geq \theta \\ 0 & \text{if } \zeta < \theta \end{cases}$$

Despite the early success of the Perceptron and artificial neural network research, there were many people who felt that there was limited promise in these techniques. Among these were Marvin Minsky and Seymour Papert, whose 1969 book *Perceptrons* was used to discredit ANN research and focus attention on the apparent limitations of ANN work. One of the limitations that Minsky and Papert pointed out most clearly was the fact that the Perceptron was not able to classify patterns that are not linearly separable in the input space. Below, the figure on the left shows an input space with a linearly separable classification problem. The figure on the right, in contrast, shows an input space where the classifications are not linearly separable.

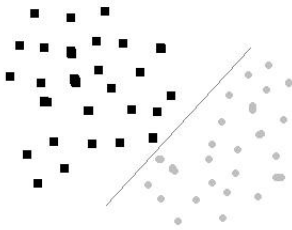


Figure 5

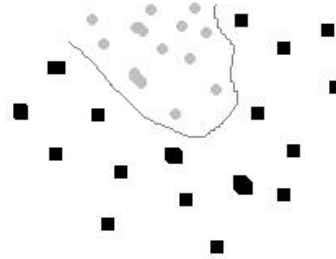


Figure 6

Despite the failure of the Mark I Perceptron to handle non-linearly separable data, it was not an inherent failure of the technology, but a matter of scale. The Mark I was a two layer Perceptron, Hecht-Nielsen showed in 1990 that a three layer machine (multi layer Perceptron, or MLP) was capable of solving nonlinear separation problems. *Perceptrons* ushered in what some call the “quiet years”, where ANN research was at a minimum of interest. It wasn’t until the rediscovery of the backpropagation algorithm in 1986 that the field gained widespread interest again.

4.2.2 Backpropagation and Rebirth

The backpropagation algorithm, originally discovered by Werbos in 1974 was rediscovered in 1986 with the book *Learning Internal Representation by Error Propagation* by Rumelhart, Hinton and Williams. Backpropagation is a form of the gradient descent algorithm used with artificial neural networks for minimization and curve-fitting.

In 1987 the IEEE annual international ANN conference was started for ANN researchers. In 1987 the International Neural Network Society (INNS) was formed, along with the INNS Neural Networking journal in 1988.

5 MATLAB Neural Networking Toolbox

5.1 MATLAB

MATLAB[®] is an ideal tool for working with artificial neural networks for a number of reasons. First, MATLAB is highly efficient in performing vector and matrix calculations. Second, MATLAB comes with a specialized Neural Network Toolbox[®] which contains a number of useful tools for working with artificial neural networks.

This book is going to utilize the MATLAB programming environment and the Neural Network Toolbox to do examples and problems throughout the book.

5.2 Other Neural Network Software

Even though this book is going to focus on MATLAB for its problems and examples, there are a number of other tools that can be used for constructing, testing, and implementing neural networks.

6 Activation Functions

6.1 Activation Functions

There are a number of common activation functions in use with neural networks. This is not an exhaustive list.

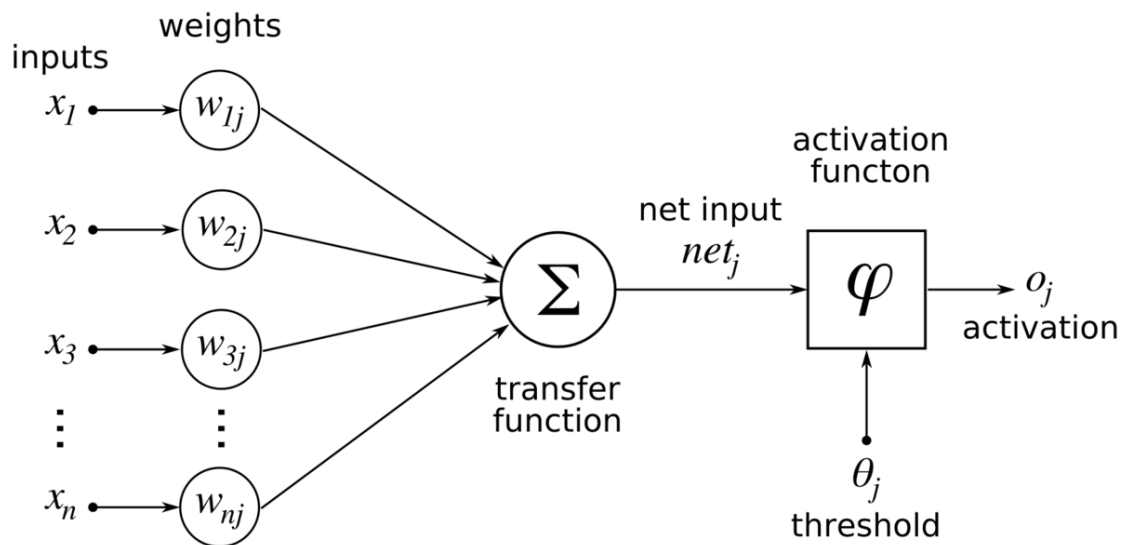
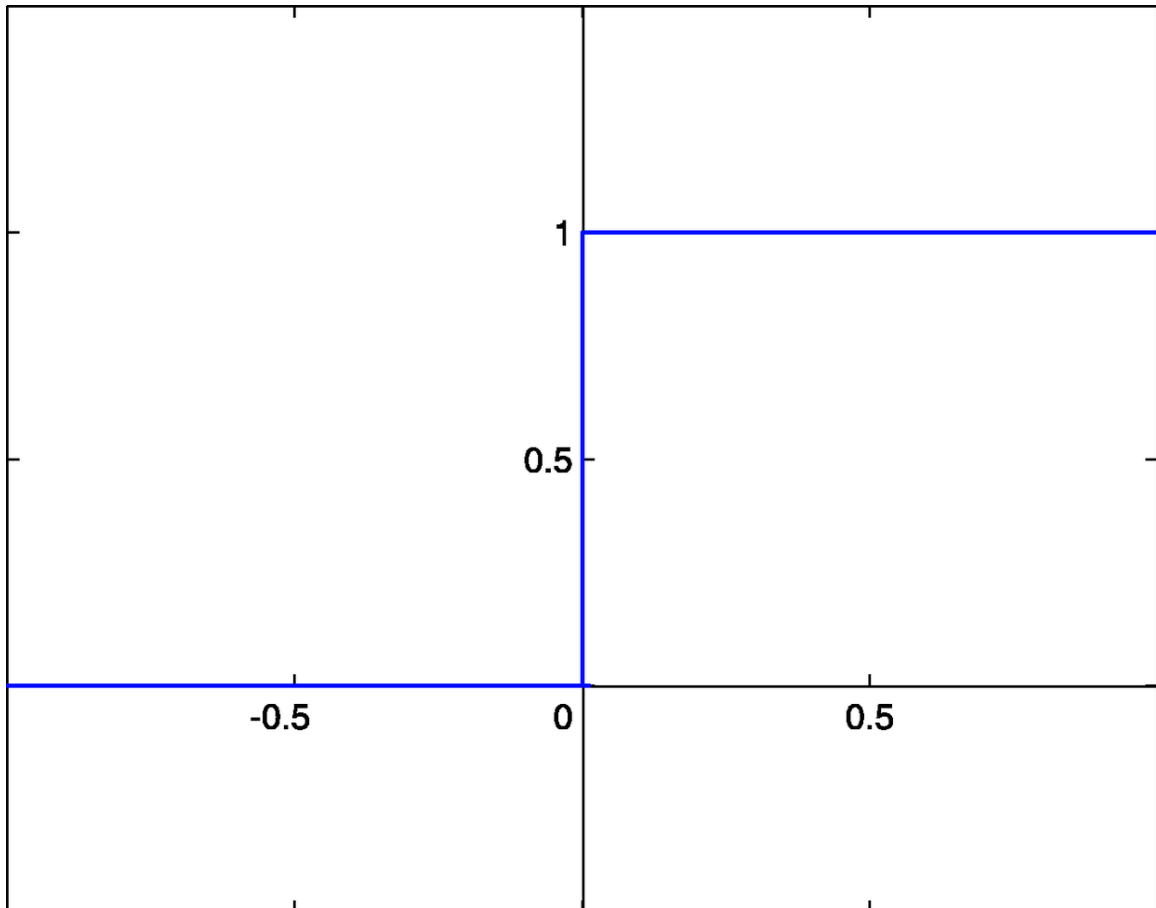


Figure 7

6.2 Step Function

A step function is a function like that used by the original Perceptron. The output is a certain value, A_1 , if the input sum is above a certain threshold and A_0 if the input sum is below a certain threshold. The values used by the Perceptron were $A_1 = 1$ and $A_0 = 0$.

**Figure 8**

These kinds of step activation functions are useful for binary classification schemes. In other words, when we want to classify an input pattern into one of two groups, we can use a binary classifier with a step activation function. Another use for this would be to create a set of small **feature identifiers**. Each identifier would be a small network that would output a 1 if a particular input feature is present, and a 0 otherwise. Combining multiple feature detectors into a single network would allow a very complicated clustering or classification problem to be solved.

6.3 Linear combination

A linear combination is where the weighted sum input of the neuron plus a linearly dependant bias becomes the system output. Specifically:

$$y = \zeta + b$$

In these cases, the sign of the output is considered to be equivalent to the 1 or 0 of the step function systems, which enables the two methods be to equivalent if

$$\theta = -b$$

6.4 Continuous Log-Sigmoid Function

A log-sigmoid function, also known as a logistic function, is given by the relationship:

$$\sigma(t) = \frac{1}{1 + e^{-\beta t}}$$

Where β is a slope parameter. This is called the log-sigmoid because a sigmoid can also be constructed using the hyperbolic tangent function instead of this relation, in which case it would be called a tan-sigmoid. Here, we will refer to the log-sigmoid as simply “sigmoid”. The sigmoid has the property of being similar to the step function, but with the addition of a region of uncertainty. Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons, although not exactly the same. Below is the graph of a sigmoid function.

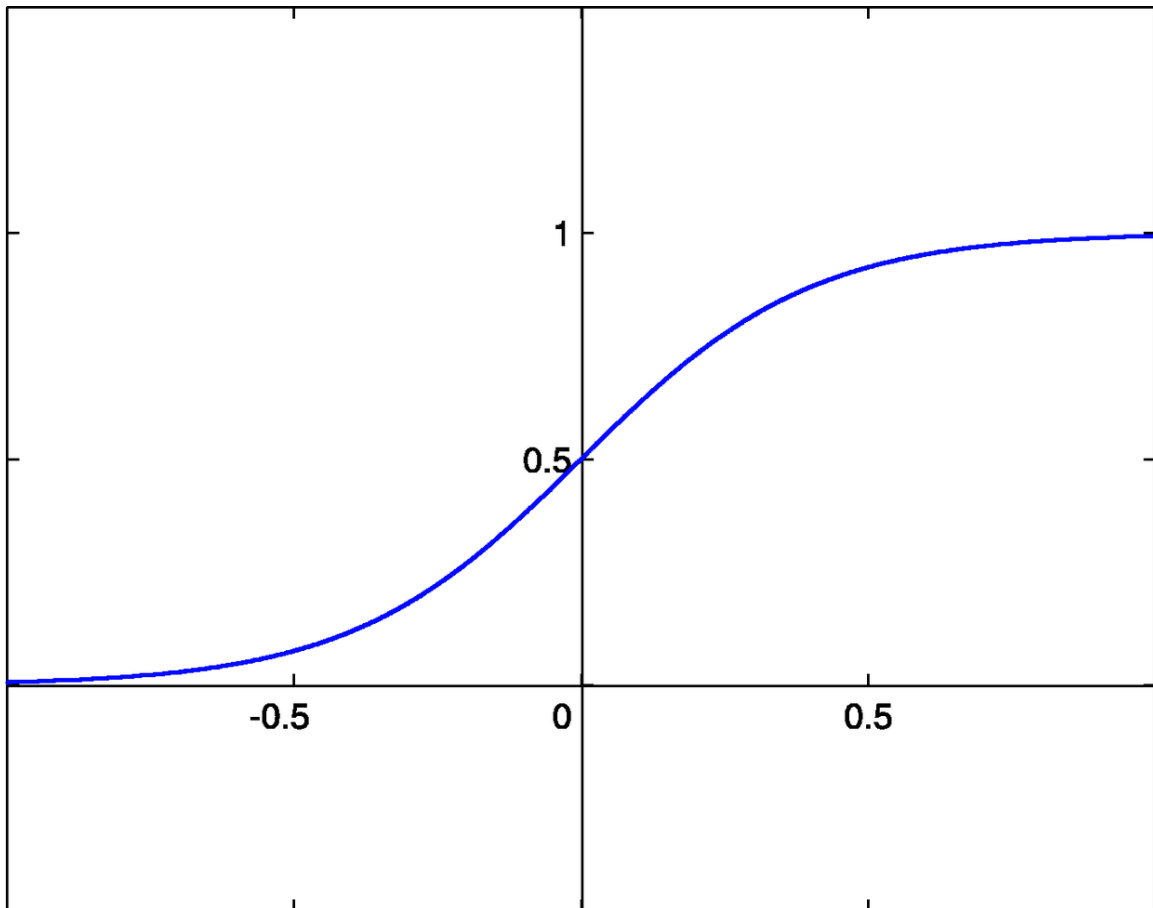


Figure 9

Sigmoid functions are also prized because their derivatives are easy to calculate, which is helpful for calculating the weight updates in certain training algorithms. The derivative when $\beta = 1$ is given by:

$$\frac{d\sigma(t)}{dt} = \sigma(t)[1 - \sigma(t)]$$

When $\beta \neq 1$, using $\sigma(\beta, t) = \frac{1}{1+e^{-\beta t}}$, the derivative is given by:

$$\frac{d\sigma(\beta, t)}{dt} = \beta[\sigma(\beta, t)[1 - \sigma(\beta, t)]]$$

6.5 Continuous Tan-Sigmoid Function

$$\sigma(t) = \tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$$

Its derivative is:

$$\frac{d\sigma(t)}{dt} = 1 - \tanh^2(t) = \operatorname{sech}^2(t) = 1 - \frac{(e^t - e^{-t})^2}{(e^t + e^{-t})^2}$$

6.6 Softmax Function

The softmax activation function is useful predominantly in the output layer of a clustering system. Softmax functions convert a raw value into a posterior probability. This provides a measure of certainty. The softmax activation function is given as:

$$y_i = \frac{e^{\zeta_i}}{\sum_{j \in L} e^{\zeta_j}}$$

L is the set of neurons in the output layer.

7 Feed-Forward Networks

7.1 Feedforward Systems

Feed-forward neural networks are the simplest form of ANN. Shown below, a feed-forward neural net contains only forward paths.

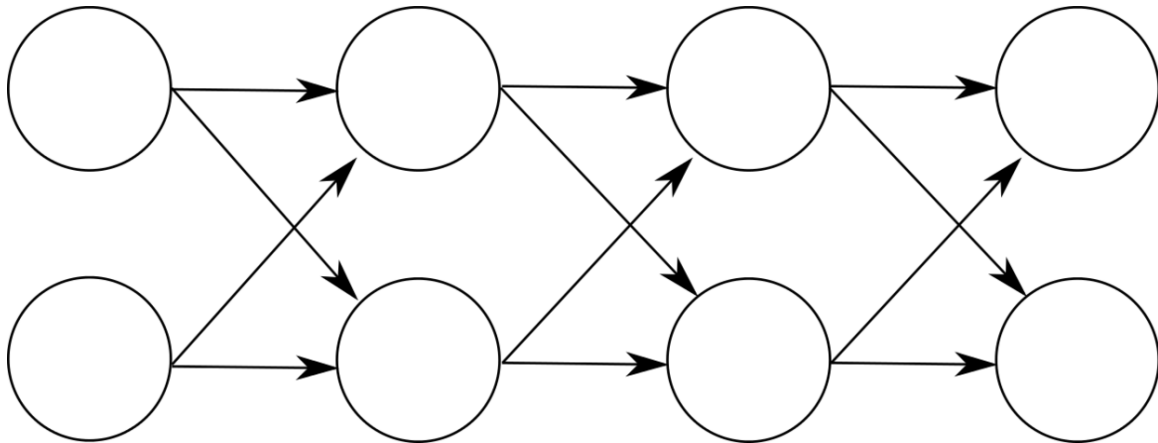


Figure 10

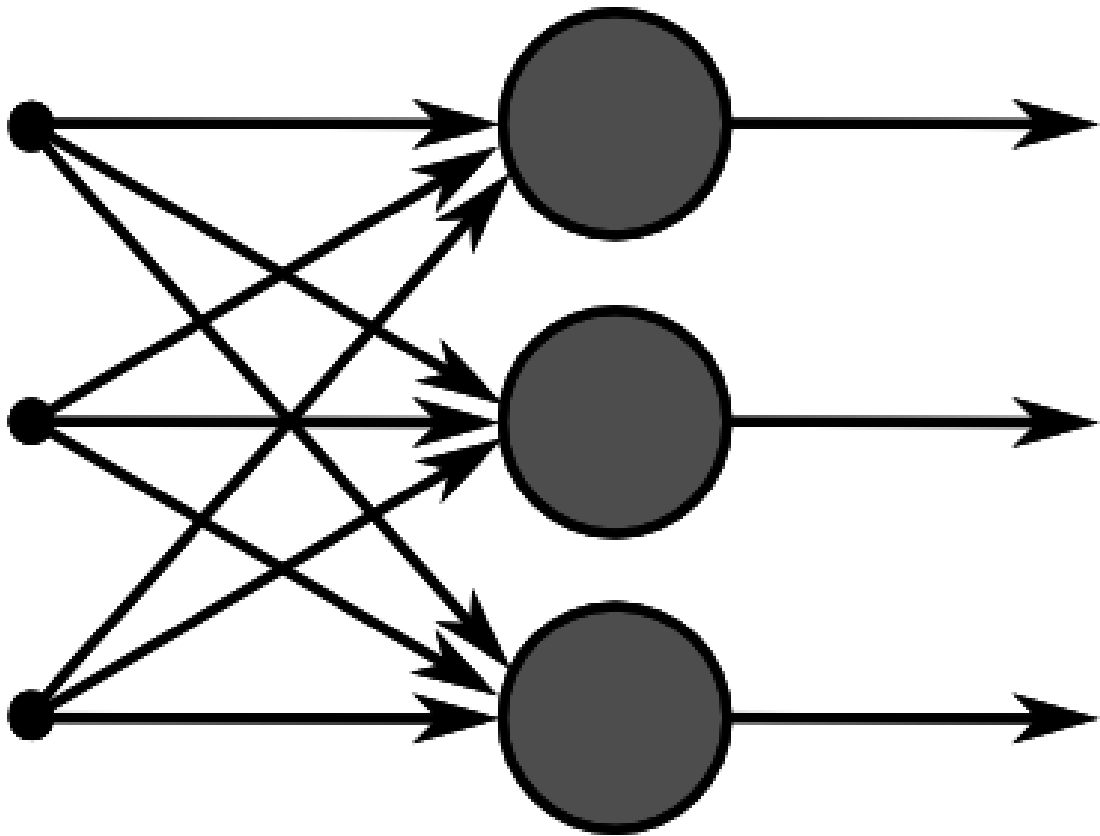


Figure 11

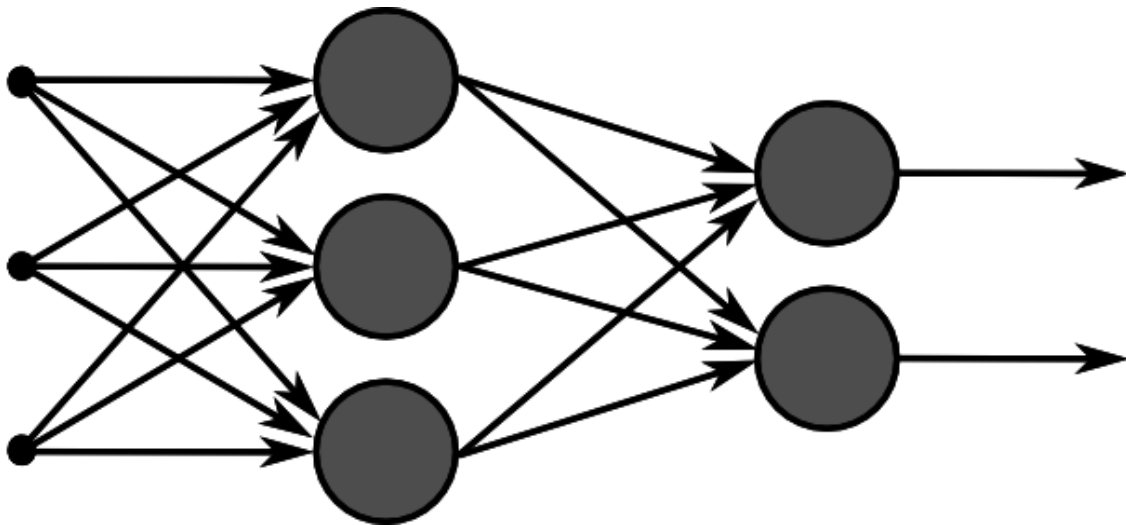


Figure 12

7.2 Connection Weights

In a feed-forward system PE are arranged into distinct layers with each layer receiving input from the previous layer and outputting to the next layer. There is no feedback. This means that signals from one layer are not transmitted to a previous layer. This can be stated mathematically as:

$$w_{ij} = 0 \text{ if } i = j$$

$$w_{ij} = 0 \text{ if } \textit{layer}(i) \leq \textit{layer}(j)$$

Weights of direct feedback paths, from a neuron to itself, are zero. Weights from a neuron to a neuron in a previous layer are also zero. Notice that weights for the forward paths may also be zero depending on the specific network architecture, but they do not need to be. A network without all possible forward paths is known as a sparsely connected network, or a non-fully connected network. The percentage of available connections that are utilized is known as the connectivity of the network.

7.3 Mathematical Relationships

The weights from each neuron in layer $l - 1$ to the neurons in layer l are arranged into a matrix \mathbf{w}_l . Each row corresponds to a neuron in $l - 1$, and each column corresponds to a neuron in l . The input signal from $l - 1$ to l is the vector \mathbf{x}_l . If ρ_l is a vector of activation functions $[\sigma_1 \sigma_2 \dots \sigma_n]$ that acts on each row of input and \mathbf{b}_l is an arbitrary offset vector (for generalization) then the total output of layer l is given as:

$$\mathbf{y}_l = \rho_l(\mathbf{w}_l \mathbf{x}_l + \mathbf{b}_l)$$

Two layers of output can be calculated by substituting the output from the first layer into the input of the second layer:

$$\mathbf{y}_l = \rho_l(\mathbf{w}_l \rho_{l-1}(\mathbf{w}_{l-1} \mathbf{x}_{l-1} + \mathbf{b}_{l-1}) + \mathbf{b}_l)$$

This method can be continued to calculate the output of a network with an arbitrary number of layers. Notice that as the number of layers increases, so does the complexity of this calculation. Sufficiently large neural networks can quickly become too complex for direct mathematical analysis.

8 Radial Basis Function Networks

In a radial basis function (RBF) networks are neural nets with three layers. The first input layer feeds data to a hidden intermediate layer. The hidden layer processes the data and transports it to the output layer. Only the tap weights between the hidden layer and the output layer are modified during training. Each hidden layer neuron represents a basis function of the output space, with respect to a particular center in the input space. The activation function chosen is commonly a Gaussian kernel:

$$\sigma(\zeta) = e^{-\beta\zeta^2}$$

This kernel is centered at the point in the input space specified by the weight vector. The closer the input signal is to the current weight vector, the higher the output of the neuron will be. Radial basis function networks are used commonly in function approximation and series prediction.

9 Recurrent Networks

9.1 Recurrent Networks

In a recurrent network, the weight matrix for each layer l contains input weights from all other neurons in the network, not just neurons from the previous layer. The additional complexity from these feedback paths can have a number of advantages and disadvantages in the network.

9.2 Simple Recurrent Networks

Recurrent networks, in contrast to feed-forward networks, do have feedback elements that enable signals from one layer to be fed back to a previous layer. A basic recurrent network is shown in figure 6. A simple recurrent network is one with three layers, an input, an output, and a hidden layer. A set of additional context units are added to the input layer that receive input from the hidden layer neurons. The feedback paths from the hidden layer to the context units have a fixed weight of unity.

A fully recurrent network is one where every neuron receives input from all other neurons in the system. Such networks cannot be easily arranged into layers. A small subset of neurons receives external input, and another small subset produce system output. A recurrent network is known as symmetrical network if:

$$w_{ij} = w_{ji} \forall i, j$$

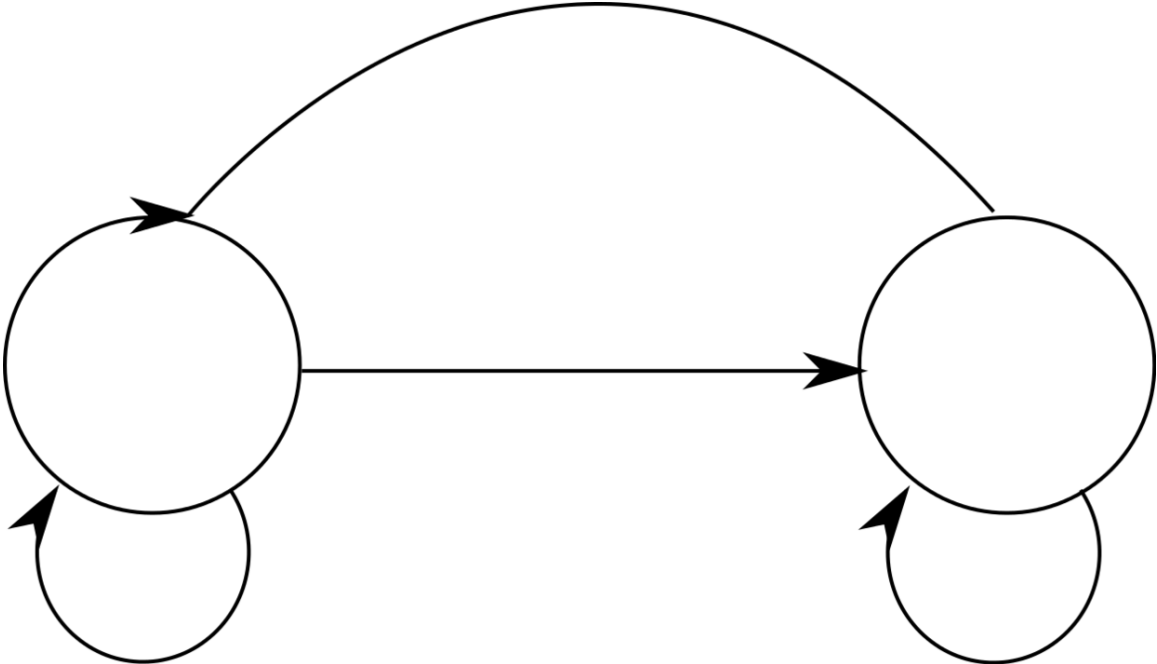


Figure 13

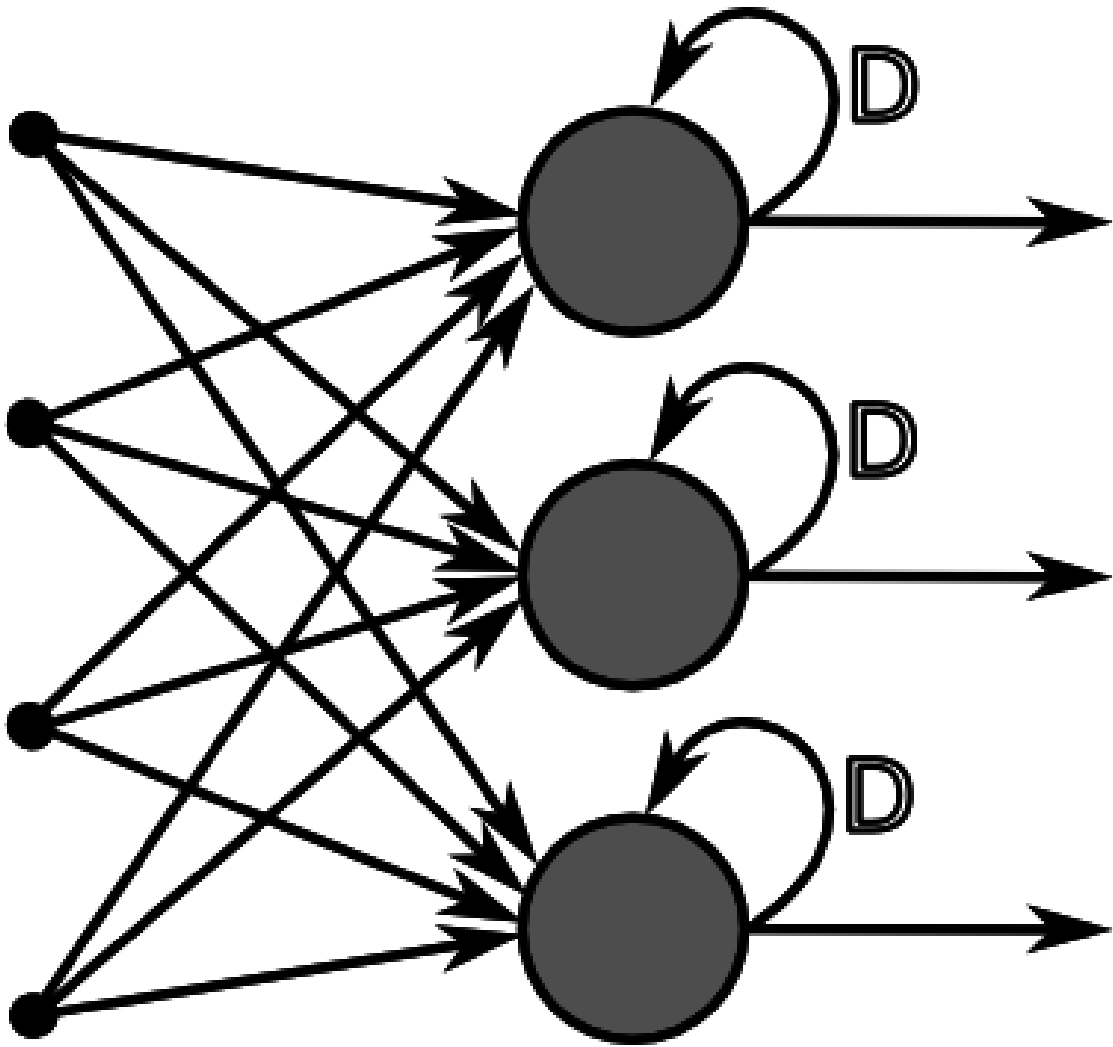


Figure 14

10 Echo State Networks

Echo state networks are recurrent networks where the hidden layer neurons are not completely connected to all input neurons. Such networks where all possible connections are not made are known as sparsely connected networks. Only the weights from the hidden layer to the output layer may be altered during training.

Echo state networks are useful for matching and reproducing specific input patterns. Because the only tap weights modified during training are the output layer tap weights, training is typically quick and computationally efficient in comparison to other multi-layer networks that are not sparsely connected.

11 Hopfield Networks

11.1 Hopfield Networks

Hopfield networks are one of the oldest and simplest networks. Hopfield networks utilize a network energy function. The activation function of a binary Hopfield network is given by the signum function of a biased weighted sum:

$$y_i = (\zeta_i - \theta_i)$$

Hopfield networks are frequently binary-valued, although continuous variants do exist. Binary networks are useful for classification and clustering purposes.

11.2 Energy Function

The energy function for the network is given as:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} y_i y_j$$

Here, the y parameters are the outputs of the i th and j th units. During training the network energy should decrease until it reaches a minimum. This minimum is known as the attractor of the network. As a Hopfield network progresses, the energy minimizes itself. This means that mathematical minimization or optimization problems can be solved automatically by the Hopfield network if that problem can be formulated in terms of the network energy.

11.3 Associative Memory

Hopfield networks can be used as an associative memory network for data storage purposes. Each attractor represents a different data value that is stored in the network, and a range of associated patterns can be used to retrieve the data pattern. The number of distinct patterns p that can be stored in such a network is given approximately as:

$$p_{max} \approx 0.15n$$

Where n is the number of neurons in the network.

12 Self-Organizing Maps

12.1 Self-Organizing Maps

Self-organizing maps (SOM), sometimes called Kohonen SOM after their creator¹, are used with unsupervised learning. SOM are modeled on biological neural networks, where groups of neurons appear to self organize into specific regions with common functionality.

12.2 Neuron Regions

Different regions of the SOM network are trained to be detectors for distinct features from the input set. Initial network weights are either set randomly, or are based off the eigenvectors of the input space. The Euclidean distance from each input sample to the weight vector of each neuron is computed, and the neuron whose weight vector is most similar to the input is declared the best match unit (BMU). The update formula is given as:

$$w_j[n+1] = w_j[n] + \Theta[j, n]\alpha[n](x[n] - w_j[n])$$

Here, w is the weight vector at time n . α is a monotonically decreasing function that ensures the learning rate will decrease over time. x is the input vector, and $\Theta[j, n]$ is a measure of the distance between the BMU and neuron j at iteration n . As can be seen from this algorithm, the amount by which the neuron weight vectors change is based on the distance from the BMU, and the amount of time. Decreasing the potential for change over time helps to reduce volatility during training, and helps to ensure that the network converges.

¹ <http://en.wikipedia.org/wiki/Teuvo%20Kohonen>

13 Competitive Models

13.1 Competitive Networks

Competitive networks are networks where neurons compete with one another. The weight vector is treated as a "prototype", and is matched against the input vector. The "winner" of each training session is the neuron whose weight vector is most similar to the input vector.

An example of a competitive network is shown below.

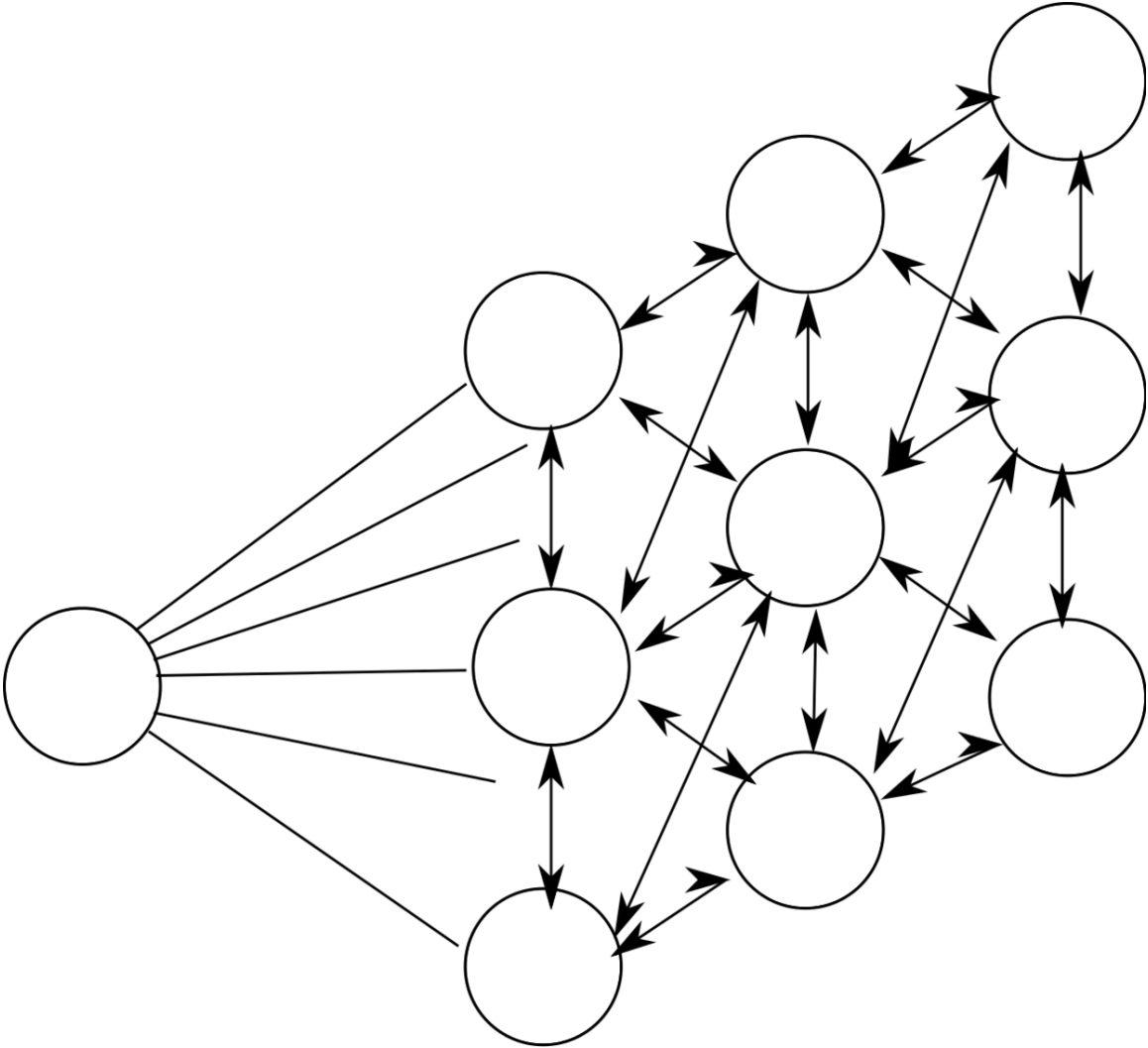


Figure 15

14 ART Models

In adaptive resonance theory (ART) networks, an overabundance of neurons leads some neurons to be committed (active) and others to be uncommitted (inactive). The weight vector, also known as the prototype, is said to resonate with the input vector if the two are sufficiently similar. Weights are only updated if they are resonating in the current iteration. ART networks commit an uncommitted neuron when a new input pattern is detected that does not resonate with any of the existing committed neurons. ART networks are fully-connected networks, in that all possible connections are made between all nodes.

15 Boltzmann Machines

Boltzmann learning compares the input data distribution P with the output data distribution of the machine, Q [24]. The distance between these distributions is given by the Kullback-Leibler distance:

$$w_{ij}[n+1] = w_{ij}[n] - \frac{\partial G}{\partial w_{ij}}$$

Where:

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}[p_{ij} - q_{ij}]$$

Here, p_{ij} is the probability that elements i and j will both be on when the system is in its training phase (positive phase), and q_{ij} is the probability that both elements i and j will be on during the production phase (negative phase). The probability that element j will be on, p_i , is given by:

$$p_i = \frac{1}{1 + e^{-\frac{\Delta E_i}{T}}}$$

T is a scalar constant known as the temperature of the system. Boltzmann learning is very powerful, but the complexity of the algorithm increases exponentially as more neurons are added to the network. To reduce this effect, a restricted Boltzmann machine (RBM) can be used. The hidden nodes in an RBM are not interconnected as they are in regular Boltzmann networks. Once trained on a particular feature set, these RBM can be combined together into larger, more diverse machines.

Because Boltzmann machine weight updates only require looking at the expected distributions of surrounding neurons, it is a plausible model for how actual biological neural networks learn.

16 Committee of Machines

Artificial neural networks can have very different properties depending on how they are constructed and how they are trained. Even in the case where two networks are trained on the same set of input data, different training algorithms can produce different systems with different characteristics. By combining multiple ANN into a single system, a committee of machines is formed. The result of a committee system is a combination of the results of the various component systems. For instance, the most common answer among a discrete set of answers in the committee can be taken as the overall answer, or the average answer can be taken.

Committee of machines (COM) systems tend to be more robust than the individual component systems, but they can also lose some of the “expertise” of the individual systems when answers are averaged out.

17 Learning Paradigms

17.1 Learning Paradigms

There are three different learning paradigms that can be used to train a neural network. Supervised and unsupervised learning are the most common, with hybrid approaches between the two becoming increasingly common as well.

17.2 Supervised Learning

Supervised learning is a technique where the input and expected output of the system are provided, and the ANN is used to model the relationship between the two. Given an input set x , and a corresponding output set y , an optimal rule is to be determined such that:

$$y = f(x) + e$$

Here, e is an approximation error that needs to be minimized. The input values are provided to the network which produces a result. This result is compared to the desired result, and this error signal is used to update the network weight vectors. Supervised learning is useful when we want the network to reproduce the characteristics of a certain relationship

17.3 Unsupervised Learning

In unsupervised learning, the data and a cost function are provided that is a function of the system input and output. The ANN is trained to minimize the cost function by finding a suitable input-output relationship.

Given an input set x , and a cost function $g(x, y)$ of the input and output sets, the goal is to minimize the cost function through a proper selection of f (the relationship between x , and y). At each training iteration, the trainer provides the input to the network, and the network produces a result. This result is put into the cost function, and the total cost is used to update the weights. Weights are continually updated until the system output produces a minimal cost. Unsupervised learning is useful in situations where a cost function is known, but a data set is not know that minimizes that cost function over a particular input space.

18 Error-Correction Learning

18.1 Error-Correction Learning

Error-Correction Learning, used with supervised learning, is the technique of comparing the system output to the desired output value, and using that error to direct the training. In the most direct route, the error values can be used to directly adjust the tap weights, using an algorithm such as the backpropagation algorithm. If the system output is y , and the desired system output is known to be d , the error signal can be defined as:

$$e = d - y$$

Error correction learning algorithms attempt to minimize this error signal at each training iteration. The most popular learning algorithm for use with error-correction learning is the backpropagation algorithm, discussed below.

18.2 Gradient Descent

The gradient descent algorithm is not specifically an ANN learning algorithm. It has a large variety of uses in various fields of science, engineering, and mathematics. However, we need to discuss the gradient descent algorithm in order to fully understand the backpropagation algorithm. The gradient descent algorithm is used to minimize an error function $g(y)$, through the manipulation of a weight vector w . The cost function should be a linear combination of the weight vector and an input vector x . The algorithm is:

$$w_{ij}[n+1] = w_{ij}[n] + \eta g(w_{ij}[n])$$

Here, η is known as the step-size parameter, and affects the rate of convergence of the algorithm. If the step size is too small, the algorithm will take a long time to converge. If the step size is too large the algorithm might oscillate or diverge.

The gradient descent algorithm works by taking the gradient of the weight space to find the path of steepest descent. By following the path of steepest descent at each iteration, we will either find a minimum, or the algorithm could diverge if the weight space is infinitely decreasing. When a minimum is found, there is no guarantee that it is a global minimum, however.

18.3 Backpropagation

The **backpropagation** algorithm, in combination with a supervised error-correction learning rule, is one of the most popular and robust tools in the training of artificial neural networks. Back propagation passes error signals backwards through the network during training to update the weights of the network. Because of this dependence on bidirectional data flow during training, backpropagation is not a plausible reproduction of biological learning mechanisms. When talking about backpropagation, it is useful to define the term **interlayer** to be a layer of neurons, and the corresponding input tap weights to that layer. We use a superscript to denote a specific interlayer, and a subscript to denote the specific neuron from within that layer. For instance:

$$\zeta_j^l = \sum_{i=1}^{N^{l-1}} w_{ij}^l x_i^{l-1} \quad (1)$$

$$x_j^l = \sigma(\zeta_j^l) \quad (2)$$

Where x_i^{l-1} are the outputs from the previous interlayer (the inputs to the current interlayer), w_{ij}^l is the tap weight from the i input from the previous interlayer to the j element of the current interlayer. N^{l-1} is the total number of neurons in the previous interlayer.

The backpropagation algorithm specifies that the tap weights of the network are updated iteratively during training to approach the minimum of the error function. This is done through the following equation:

$$w_{ij}^l[n] = w_{ij}^l[n-1] + \delta w_{ij}^l[n] \quad (3)$$

$$w_{ij}^{l-1}[n] = \eta \delta_j^l x_i^{l-1}[n] + \mu \Delta w_{ij}^l[n-1] \quad (3)$$

The relationship between this algorithm and the gradient descent algorithm should be immediately apparent. Here, η is known as the learning rate, not the step-size, because it affects the speed at which the system learns (converges). The parameter μ is known as the momentum parameter. The momentum parameter forces the search to take into account its movement from the previous iteration. By doing so, the system will tend to avoid local minima or saddle points, and approach the global minimum. We will discuss these terms in greater detail in the next section.

The parameter δ is what makes this algorithm a “back propagation” algorithm. We calculate it as follows:

$$\delta_j^l = \frac{dx_j^l}{dt} \sum_{k=1}^r \delta_k^{l+1} w_{kj}^{l+1} \quad (4)$$

The δ function for each layer depends on the δ from the previous layer. For the special case of the output layer (the highest layer), we use this equation instead:

$$\delta_j^l = \frac{dx_j^l}{dt} (x_j^l - y_j) \quad (5)$$

In this way, the signals propagate backwards through the system from the output layer to the input layer. This is why the algorithm is called the backpropagation algorithm.

18.3.1 Log-Sigmoid Backpropagation

If we use log-sigmoid activation functions for our neurons, the derivatives simplify, and our backpropagation algorithm becomes:

$$\delta_j^l = x_j^l(1 - x_j^l)(x_j^l - y_j)$$

For the output layer, and

$$\delta_j^l = x_j^l(1 - x_j^l) \sum_{k=1}^r \delta_k^{l+1} w_{kj}^{l+1}$$

for all the hidden inner layers. This property makes the sigmoid function desirable for systems with a limited ability to calculate derivatives.

18.3.2 Learning Rate

The **learning rate** is a common parameter in many of the learning algorithms, and affects the speed at which the ANN arrives at the minimum solution. In backpropagation, the learning rate is analogous to the step-size parameter from the gradient-descent algorithm. If the step-size is too high, the system will either oscillate about the true solution, or it will diverge completely. If the step-size is too low, the system will take a long time to converge on the final solution.

18.3.3 Momentum Parameter

The **momentum parameter** is used to prevent the system from converging to a local minimum or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable. A momentum coefficient that is too low cannot reliably avoid local minima, and also can slow the training of the system.

19 Hebbian Learning

19.1 Hebbian Learning

Hebbian learning is one of the oldest learning algorithms, and is based in large part on the dynamics of biological systems. A synapse between two neurons is strengthened when the neurons on either side of the synapse (input and output) have highly correlated outputs. In essence, when an input neuron fires, if it frequently leads to the firing of the output neuron, the synapse is strengthened. Following the analogy to an artificial system, the tap weight is increased with high correlation between two sequential neurons.

19.2 Mathematical Formulation

Mathematically, we can describe Hebbian learning as:

$$w_{ij}[n+1] = w_{ij}[n] + \eta x_i[n]x_j[n]$$

Here, η is a learning rate coefficient, and x are the outputs of the i th and j th elements.

19.3 Plausibility

The Hebbian learning algorithm is performed locally, and doesn't take into account the overall system input-output characteristic. This makes it a plausible theory for biological learning methods, and also makes Hebbian learning processes ideal in VLSI hardware implementations where local signals are easier to obtain.

20 Competitive Learning

20.1 Competitive Learning

Competitive learning is a rule based on the idea that only one neuron from a given iteration in a given layer will fire at a time. Weights are adjusted such that only one neuron in a layer, for instance the output layer, fires. Competitive learning is useful for classification of input patterns into a discrete set of output classes. The “winner” of each iteration, element i^* , is the element whose total weighted input is the largest. Using this notation, one example of a competitive learning rule can be defined mathematically as:

$$w_{ij}[n+1] = w_{ij}[n] + \Delta w_{ij}[n]$$

$$\Delta w_{ij} = \begin{cases} \eta(x_i - w_{ij}) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

20.2 Linear Vector Quantization

In a learning vector quantization (LVQ) machine, the input values are compared to the weight vector of each neuron. Neurons who most closely match the input are known as the best match unit (BMU) of the system. The weight vector of the BMU and those of nearby neurons are adjusted to be closer to the input vector by a certain step size. Neurons become trained to be individual feature detectors, and a combination of feature detectors can be used to identify large classes of features from the input space. The LVQ algorithm is a simplified precursor to more advanced learning algorithms, such as the self-organizing map. LVQ training is a type of competitive learning rule.

21 Boltzmann Learning

Boltzmann learning is statistical in nature, and is derived from the field of thermodynamics. It is similar to error-correction learning and is used during supervised training. In this algorithm, the state of each individual neuron, in addition to the system output, are taken into account. In this respect, the Boltzmann learning rule is significantly slower than the error-correction learning rule. Neural networks that use Boltzmann learning are called Boltzmann machines.

Boltzmann learning is similar to an error-correction learning rule, in that an error signal is used to train the system in each iteration. However, instead of a direct difference between the result value and the desired value, we take the difference between the probability distributions of the system.

22 ART Learning

22.1 ART Learning

Adaptive Resonance Theory (ART) learning algorithms compare the weight vector, known as the prototype, to the current input vector to produce a distance, r . The distance is compared to a specified scalar, the vigilance parameter p . All output nodes start off in the uncommitted state. When a new input sequence is detected that does not resonate with any committed nodes, an uncommitted node is committed, and its prototype vector is set to the current input vector.

23 Self-Organizing Maps

23.1 Self-Organizing Maps

Self-organizing maps (SOM), sometimes called Kohonen SOM after their creator¹, are used with unsupervised learning. SOM are modeled on biological neural networks, where groups of neurons appear to self organize into specific regions with common functionality.

23.2 Neuron Regions

Different regions of the SOM network are trained to be detectors for distinct features from the input set. Initial network weights are either set randomly, or are based off the eigenvectors of the input space. The Euclidean distance from each input sample to the weight vector of each neuron is computed, and the neuron whose weight vector is most similar to the input is declared the best match unit (BMU). The update formula is given as:

$$w_j[n+1] = w_j[n] + \Theta[j, n]\alpha[n](x[n] - w_j[n])$$

Here, w is the weight vector at time n . α is a monotonically decreasing function that ensures the learning rate will decrease over time. x is the input vector, and $\Theta[j, n]$ is a measure of the distance between the BMU and neuron j at iteration n . As can be seen from this algorithm, the amount by which the neuron weight vectors change is based on the distance from the BMU, and the amount of time. Decreasing the potential for change over time helps to reduce volatility during training, and helps to ensure that the network converges.

¹ <http://en.wikipedia.org/wiki/Teuvo%20Kohonen>

24 Pattern Recognition

Artificial neural networks are useful for pattern matching applications. Pattern matching consists of the ability to identify the class of input signals or patterns. Pattern matching ANN are typically trained using supervised learning techniques. One application where artificial neural nets have been applied extensively is optical character recognition (OCR). OCR has been a very successful area of research involving artificial neural networks.

An example of a pattern matching neural network is that used by VISA for identifying suspicious transactions and fraudulent purchases. When input symbols do not match an accepted pattern, the system raises a warning flag that indicates a potential problem.

25 Clustering

Similar to pattern matching, clustering is the ability to associate similar input patterns together, based on a measurement of their similarity or dissimilarity. An example of a clustering problem is the Netflix Prize, a competition to improve the Netflix recommendation system. Competitors are attempting to produce software systems that can suggest movies based on movies the viewer has already rated. Essentially, clustering movies based on the predicted like or dislike of the viewer.

26 Feature Detection

Feature detection or “association” networks are trained using non-noisy data, in order to recognize similar patterns in noisy or incomplete data. Correctly detecting features in the presence of noise can be used as an important tool for noise reduction and filtering.

For example, neural nets have been used successfully in a number of medical applications for detection of disease. One particular example is the use of ANN to detect breast cancer in mammography images.

27 Series Prediction

ANN can be trained to match the statistical properties of a particular input signal, and can even be used to predict future values of time series. There are a number of applications of feature prediction that have received significant research attention. Among these are financial prediction, meteorological prediction, and electric load prediction. ANN have shown themselves to be very robust in predicting complicated series, including non-linear non-stationary chaotic systems.

Financial prediction is useful for anticipating events in the economic system, which is considered to be a chaotic system. ANN have been used to predict the performance and failure rate of companies, changes in the exchange rate, and other economic metrics.

Meteorological prediction is a difficult process because current atmospheric models rely on highly recursive sets of differential equations which can be difficult to calculate, and which propagate errors through the successive iterations. Using neural nets for meteorological prediction is a non-trivial task, and there have been many conflicting reports of the efficacy of the technique.

28 Data Compression

29 Curve Fitting

29.1 Curve Fitting

Curve-fitting problems represent an attempt for the neural network to identify and approximate an arbitrary input-output relation. Once the relation has been modeled to the necessary accuracy by the network, it can be used for a variety of tasks, such as series prediction, function approximation, and function optimization.

29.2 Function Approximation

Function approximation or modeling is the act of training a neural network using a given set of input-output data (typically through supervised learning) in order to deduce the relationship between the input and the output. After training, such an ANN can be used as a black box with an input-output characteristic approximately equal to the relationship of the training problems. Because of the modular and non-linear nature of artificial neural nets, they are considered to be able to approximate any arbitrary function to an arbitrary degree of accuracy. More accuracy in this case represents a tradeoff with system complexity, and the ability to generalize.

30 Optimization

Optimization ANNs are concerned with the minimization of a particular cost function with respect to certain constraints. ANN are shown to be capable of highly efficient optimization.

31 Control

31.1 Control Systems

Artificial neural networks have been employed for use in control systems¹ because of their ability to identify patterns, and to match arbitrary non-linear response curves.

31.2 See Also

- Control Systems²

¹ <http://en.wikibooks.org/wiki/Control%20Systems>

² <http://en.wikibooks.org/wiki/Control%20Systems>

32 Criticisms and Problems

33 Artificial Intelligence

34 Resources

34.1 Wikibooks

- Artificial Intelligence¹
- Linear Algebra²
- Abstract Algebra³
- Calculus⁴
- Signals and Systems⁵
- Engineering Analysis⁶
- MATLAB Programming⁷

34.2 Commons

- Commons:Artificial neural network⁸
- Commons:Category:Neural networks⁹

1 <http://en.wikibooks.org/wiki/Artificial%20Intelligence>
2 <http://en.wikibooks.org/wiki/Linear%20Algebra>
3 <http://en.wikibooks.org/wiki/Abstract%20Algebra>
4 <http://en.wikibooks.org/wiki/Calculus>
5 <http://en.wikibooks.org/wiki/Signals%20and%20Systems>
6 <http://en.wikibooks.org/wiki/Engineering%20Analysis>
7 <http://en.wikibooks.org/wiki/MATLAB%20Programming>
8 <http://en.wikibooks.org/wiki/Artificial%20neural%20network>
9 <http://en.wikibooks.org/wiki/Category%3A%20Neural%20networks>

35 Contributors

Edits	User
1	Avicennasis ¹
1	HethrirBot ²
1	Markun ³
2	Panic2k4 ⁴
1	Recent Runes ⁵
1	Sabita ⁶
85	Whiteknight ⁷
2	Xania ⁸

¹ <http://en.wikibooks.org/w/index.php?title=User:Avicennasis>
² <http://en.wikibooks.org/w/index.php?title=User:HethrirBot>
³ <http://en.wikibooks.org/w/index.php?title=User:Markun>
⁴ <http://en.wikibooks.org/w/index.php?title=User:Panic2k4>
⁵ http://en.wikibooks.org/w/index.php?title=User:Recent_Runes
⁶ <http://en.wikibooks.org/w/index.php?title=User:Sabita>
⁷ <http://en.wikibooks.org/w/index.php?title=User:Whiteknight>
⁸ <http://en.wikibooks.org/w/index.php?title=User:Xania>

List of Figures

- GFDL: Gnu Free Documentation License. <http://www.gnu.org/licenses/fdl.html>
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. <http://creativecommons.org/licenses/by-sa/3.0/>
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. <http://creativecommons.org/licenses/by-sa/2.5/>
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. <http://creativecommons.org/licenses/by-sa/2.0/>
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. <http://creativecommons.org/licenses/by-sa/1.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/deed.en>
- cc-by-2.5: Creative Commons Attribution 2.5 License. <http://creativecommons.org/licenses/by/2.5/deed.en>
- cc-by-3.0: Creative Commons Attribution 3.0 License. <http://creativecommons.org/licenses/by/3.0/deed.en>
- GPL: GNU General Public License. <http://www.gnu.org/licenses/gpl-2.0.txt>
- LGPL: GNU Lesser General Public License. <http://www.gnu.org/licenses/lgpl.html>
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. <http://artlibre.org/licence/lal/de>
- CFR: Copyright free use.

- EPL: Eclipse Public License. <http://www.eclipse.org/org/documents/epl-v10.php>

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses⁹. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrowser.

1		GFDL
2	Dake ¹⁰ , Mysid ¹¹	
3	en:User:Cburnett ¹²	GFDL
4		GFDL
5		PD
6		PD
7	Chrislb ¹³	GFDL
8	Chrislb ¹⁴	GFDL
9	Chrislb ¹⁵	GFDL
10		GFDL
11	Chrislb ¹⁶	GFDL
12	Chrislb ¹⁷	GFDL
13		GFDL
14	Chrislb ¹⁸	GFDL
15		GFDL

-
- 10 <http://en.wikibooks.org/wiki/User%3ADake>
11 <http://en.wikipedia.org/wiki/User%3AMysid>
12 <http://en.wikibooks.org/wiki/%3Aen%3AUser%3ACburnett>
13 <http://en.wikibooks.org/wiki/User%3AChrislb>
14 <http://en.wikibooks.org/wiki/User%3AChrislb>
15 <http://en.wikibooks.org/wiki/User%3AChrislb>
16 <http://en.wikibooks.org/wiki/User%3AChrislb>
17 <http://en.wikibooks.org/wiki/User%3AChrislb>
18 <http://en.wikibooks.org/wiki/User%3AChrislb>

36 Licenses

36.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer

network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sub-licensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work)

from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest had or could give if it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

36.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a worldwide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on, the exercise of, one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject (The Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero or more Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of the numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal ef-

fect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year>
<name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you

must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition of the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title

Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History". Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add an-

other; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 1) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4e, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

36.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.