Neural Networks.

Hervé Abdi[1]
The University of Texas at Dallas

Introduction

Neural networks are adaptive statistical models based on an analogy with the structure of the brain. They are *adaptive* because they can learn to estimate the parameters of some population using a small number of exemplars (one or a few) at a time. They do not differ *essentially* from standard statistical models. For example, one can find neural network architectures akin to DISCRIMINANT ANALYSIS, PRINCIPAL COMPONENT ANALYSIS, LOGISTIC REGRESSION, and other techniques. In fact, the same mathematical tools can be used to analyze standard statistical models and neural networks. Neural networks are used as *statistical tools* in a variety of fields, including psychology, statistics, engineering, econometrics, and even physics. They are used also as *models* of cognitive processes by neuro- and cognitive scientists.

Basically, neural networks are built from simple units, sometimes called *neurons* or cells by analogy with the real thing. These units are linked by a set of weighted connections. Learning is usually accomplished by modification of the connection weights. Each unit codes or corresponds to a feature or a characteristic of a pattern that we want to analyze or that we want to use as a predictor.

These networks usually organize their units into several layers. The first layer is called the *input* layer, the last one the *output* layer. The intermediate layers (if any) are called the *hidden* layers. The information to be analyzed is fed to the neurons of the first layer and then propagated to the neurons of the second layer for further processing. The result of this processing is then propagated to the next layer and so on until the last layer. Each unit receives some information from other units (or from the external world through some devices) and processes this information, which will be converted into the output of the unit.

The goal of the network is to learn or to discover some association between input and output patterns, or to analyze, or to find the structure of the input patterns. The learning process is achieved through the modification of the connection weights between units. In statistical terms, this is equivalent to

[1] In: Lewis-Beck M., Bryman, A., Futing T. (Eds.) (2003). *Encyclopedia of Social Sciences Research Methods.* Thousand Oaks (CA): Sage.
Address correspondence to
Hervé Abdi
Program in Cognition and Neurosciences, MS: Gr.4.1,
The University of Texas at Dallas,
Richardson, TX 75083–0688, USA
E-mail: `herve@utdallas.edu` `http://www.utdallas.edu/∼herve`

interpreting the value of the connections between units as parameters (*e.g.,* like the values of a and b in the regression equation $\widehat{y} = a + bx$) to be estimated. The learning process specifies the "algorithm" used to estimate the parameters.

The building blocks of neural networks

Neural networks are made of basic units (see Figure 1) arranged in layers. A unit collects information provided by other units (or by the external world) to which it is connected with *weighted* connections called *synapses.* These weights, called *synaptic weights* multiply (i.e., amplify or attenuate) the input information: A positive weight is considered excitatory, a negative weight inhibitory.

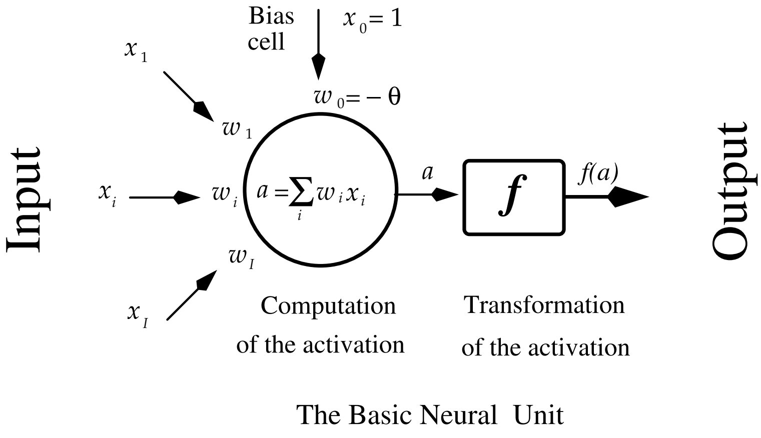

Figure 1: The basic neural unit processes the input information into the output information.

Each of these units is a simplified model of a neuron and transforms its input information into an output response. This transformation involves two steps: First, the activation of the neuron is computed as the weighted sum of it inputs, and second this activation is transformed into a response by using a *transfer* function. Formally, if each input is denoted x_i, and each weight w_i, then the activation is equal to $a = \sum x_i w_i$, and the output denoted o is obtained as $o = f(a)$. Any function whose domain is the real numbers can be used as a transfer function. The most popular ones are the linear function ($o \propto a$), the step function (activation values less than a given threshold are set to 0 or to -1 and the other values are set to $+1$), the logistic function $\left[f(x) = \dfrac{1}{1 + \exp\{-x\}}\right]$ which maps the real numbers into the interval $[-1 + 1]$ and whose derivative, needed for learning, is easily computed $\{f'(x) = f(x)\,[1 - f(x)]\}$, and the normal or Gaussian function $[o = (\sigma\sqrt{2\pi})^{-1} \times \exp\{-\frac{1}{2}(a/\sigma)^2\}]$. Some of these functions can include probabilistic variations; for example, a neuron can transform its activation into the response $+1$ with a probability of $\frac{1}{2}$ when the activation is larger than a given threshold.

The architecture (i.e., the pattern of connectivity) of the network, along with the transfer functions used by the neurons and the synaptic weights, completely specify the behavior of the network.

Learning rules

Neural networks are adaptive statistical devices. This means that they can change iteratively the values of their parameters (i.e., the synaptic weights) as a function of their performance. These changes are made according to *learning rules* which can be characterized as *supervised* (when a desired output is known and used to compute an error signal) or *unsupervised* (when no such error signal is used).

The Widrow-Hoff (a.k.a., gradient descent or Delta rule) is the most widely known supervised learning rule. It uses the difference between the actual input of the cell and the desired output as an error signal for units in the output layer. Units in the hidden layers cannot compute directly their error signal but estimate it as a function (e.g., a weighted average) of the error of the units in the following layer. This adaptation of the Widrow-Hoff learning rule is known as *error backpropagation*. With Widrow-Hoff learning, the correction to the synaptic weights is proportional to the error signal multiplied by the value of the activation given by the derivative of the transfer function. Using the derivative has the effect of making finely tuned corrections when the activation is near its extreme values (minimum or maximum) and larger corrections when the activation is in its middle range. Each correction has the immediate effect of making the error signal *smaller* if a similar input is applied to the unit. In general, supervised learning rules implement optimization algorithms akin to *descent* techniques because they search for a set of values for the free parameters (i.e., the synaptic weights) of the system such that some error function computed for the whole network is minimized.

The Hebbian rule is the most widely known unsupervised learning rule, it is based on work by the Canadian neuropsychologist Donald Hebb, who theorized that neuronal learning (i.e., synaptic change) is a local phenomenon expressible in terms of the temporal correlation between the activation values of neurons. Specifically, the synaptic change depends on both presynaptic and postsynaptic activities and states that the change in a synaptic weight is a function of the temporal correlation between the presynaptic and postsynaptic activities. Specifically, the value of the synaptic weight between two neurons increases whenever they are in the same state and decreases when they are in different states.

Some important neural network architecture

One the most popular architectures in neural networks is the *multi-layer perceptron* (see Figure 2). Most of the networks with this architecture use the Widrow-Hoff rule as their learning algorithm and the logistic function as the transfer function of the units of the hidden layer (the transfer function is in general non-linear for these neurons). These networks are very popular because they can approximate any multivariate function relating the input to the output. In a statistical framework, these networks are akin to MULTIVARIATE NON-LINEAR REGRESSION. When the input patterns are the same are the output patterns, these networks are called *auto-associators*. They are closely related to linear (if the hidden units are linear) or non-linear (if not) PRINCIPAL COMPONENT ANALYSIS and other statistical techniques linked to the GENERAL LINEAR MODEL (see Abdi et al., 1996), such as DISCRIMINANT ANALYSIS or CORRESPONDENCE ANALYSIS.

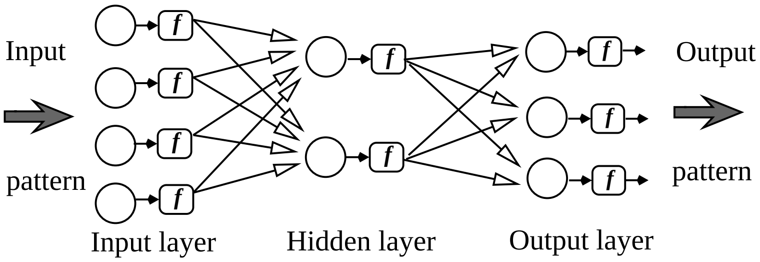

Figure 2: A multi-layer perceptron.

A recent development generalizes the *radial basis function* networks (RBF) (see Abdi, Valentin, & Edelman, 1999) and integrates them with *statistical learning theory* (see Vapnik, 1999) under the name of *support vector machine* or SVM (see Schölkopf & Smola, 2003). In these networks, the hidden units (called the support vectors) represent possible (or even real) input patterns and their response is a function to their similarity to the input pattern under consideration. The similarity is evaluated by a *kernel* function (e.g., dot product; in the radial basis function the kernel is the Gaussian transformation of the Euclidean distance between the support vector and the input). In the specific case of RBF networks—that we will use as an example of SVM—the output of the units of the hidden layers are connected to an output layer composed of linear units. In fact, these networks work by breaking the difficult problem of a nonlinear approximation into two more simple ones. The first step is a simple nonlinear mapping (the Gaussian transformation of the distance from the kernel to the input pattern), the second step corresponds to a linear transformation from the hidden layer to the output layer. Learning occurs at the level of the output layer. The main difficulty with these architectures resides in the choice of the support vectors and the specific kernels to use. These networks are used for pattern recognition, classification, and for clustering data.

Validation

From a statistical point a view, neural networks represent a class of nonparametric adaptive models. In this framework, an important issue is to evaluate the performance of the model. This is done by separating the data into two sets: the training set and the testing set. The parameters (i.e., the value of the synaptic weights) of the network are computed using the training set. Then learning is stopped and the network is evaluated with the data from the testing set. This cross-validation approach is akin to the BOOTSTRAP or the JACKKNIFE.

Useful references

Neural networks theory connects several domains from the neurosciences to engineering including statistical theory. This diversity of sources creates also a real heterogeneity in the presentation of the material as textbooks often try to address only one portion of the interested readership. The following references should be of interest for the reader interested in the statistical properties of neural networks: Abdi et al. (1999), Bishop (1995), Cherkassky and Mulier (1998), Duda, Hart & Stork (2001), Hastie, Tibshirani, & Friedman (2002), Looney (1997), Ripley (1996), and Vapnik (1999).

**References*

[1] Abdi, H., Valentin, D., & Edelman, B. (1999). *Neural networks.* Thousand Oaks (CA): Sage.

[2] Abdi, H., Valentin, D., Edelman, B., O'Toole. A.J. (1996). A Widrow-Hoff learning rule for a generalization of the linear auto-associator. *Journal of Mathematical Psychology,* **40**, 175–182.

[3] Bishop, C. M. (1995) *Neural networks for pattern recognition.* Oxford, UK: Oxford University Press.

[4] Cherkassky, V., & Mulier, F. (1998). *Learning from data.* New York: Wiley.

[5] Duda, R., Hart, P.E., Stork, D.G. (2001) *Pattern classification.* New York: Wiley.

[6] Hastie T., Tibshirani R., Friedman J. (2001). *The elements of statistical learning.* New-Yrok: Springer-Verlag

[7] Ripley, B.D. (1996) *Pattern recognition and neural networks.* Cambridge, MA: Cambridge University Press.

[8] Schölkopf B., Smola, A.J. (2003). *learning with kernels.* Cambridge (MA): MIT Press.

[9] Vapnik, V. N. (1999) *Statistical learning theory.* New York: Wiley.