Cascade Back-Propagation Learning in Neural Networks

This method would be implemented in VLSI circuitry.

Artificial neural networks are trained by example: A network is presented with training inputs for which the correct outputs are known, and the algorithm strives to adjust the weights of synaptic connections in the network to make the actual outputs approach the correct outputs. The input data are generally divided into three parts. Two of the parts, called the “training” and “cross-validation” sets, respectively, must be such that the corresponding input/output pairs are known. During training, the cross-validation set enables verification of the status of the input-to-output transformation learned by the network to avoid overlearning. The third part of the data, termed the “test” set, consists of the inputs that are required to be transformed into outputs; this set may or may not include the training set and/or the cross-validation set.

Proposed neural-network circuitry for on-chip learning would be divided into two distinct networks; one for training and one for validation. Both networks would share the same synaptic weights. During training iterations, these weights would be continuously modulated according to the CBP algorithm, which is so named because it combines features of the back-propagation and cascade-correlation algorithms. Like other algorithms for learning in artificial neural networks, the CBP algorithm specifies an iterative process for adjusting the weights of synaptic connections by descent along the gradient of an error measure in the

---

**Figure 1.** The Cascade Back-Propagation Algorithm provides the theoretical basis for design of an analog neural network that learns rapidly.
vector space of synaptic-connection weights. The error measure is usually a quadratic function of the differences between the actual and the correct outputs.

The CBP algorithm (see Figure 1) begins with calculation of the weights between the input and output layers of neurons by use of a pseudo-inverse technique. Then learning proceeds by gradient descent with the existing neurons as long as the rate of learning remains above a specified threshold level. When the rate of learning falls below this level, a new hidden neuron is added. When the quadratic error measure has descended to a value based on a predetermined criterion, the rate of learning is frozen. Thereafter, the network keeps learning endlessly with the existing neurons.

Figure 2 illustrates the cascade aspect of the CBP algorithm. To each newly added hidden neuron there are not only weighted connections from all the inputs but also a new dimension of inputs from the previous hidden neurons. The cascade aspect provides two important benefits: (1) it enables the network to get out of local minima of the quadratic error measure and (2) it accelerates convergence by eliminating the waste of time that would occur if gradient descent were allowed to occur in many equivalent subspaces of synaptic-connection-weight space. The cascade scheme concentrates learning into one subspace that is a cone of a hypercube.

The gradient descent involves, among other things, computation of derivatives of neuron transfer curves. The proposed analog implementation would provide the effectively high resolution that is needed for such computations. Provisions for addition of neurons at learning-rate-threshold levels could be made easily in hardware.

This work was done by Tuan A. Duong of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This invention is owned by NASA, and a patent application has been filed. Inquiries concerning nonexclusive or exclusive license for its commercial development should be addressed to the Patent Counsel, NASA Management Office–JPL; (818) 354-7770. Refer to NPO-19289.