Outline

Overview

- Feed-forward, supervised learning algorithm
- coNN - constructive neural networks. Two main categories:
  - evolutionary based (what is done at Brock)
  - generally constructive. CCNN is main exemplar of this group

Overview II

- creates it's own topology starting with minimal network
  - input and output layers only, as usual connected by weights.
- input of nodes code the problem being presented to the network
- output of nodes code the network's response to the input problem
- Uses backpropagation algorithms. Typically quickprop is used as the learning rule

Step 1 Training input weights

- A new hidden neuron is added one at a time.
- Instead of a minimum, we find a maximum: the maximum correlation between the candidates output and the residual error of the network output.

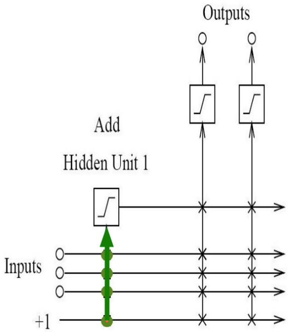
1. generate a population of candidate nodes with randomized input weights. (circa 1990 recommendation of 8)
2. inputs are connected, but not outputs
3. repeat training steps until no correlation improvement
   3.1 one epoch of the training data is run through
   3.2 update input weights using any learning rule, such that correlation is increased

Train input weights - correlation

Goal: Maximize S. Sum over all output units o of the magnitude of correlation between node output V and the error of given output node.

$$S = \sum_o \left| \sum_p (V_p - \overline{V})(E_{p,o} - \overline{E_o}) \right|$$

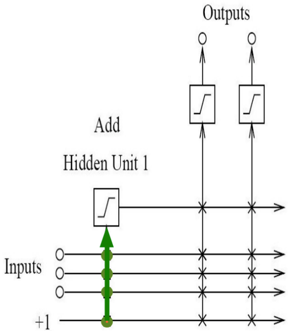
o	network output at which the error is measured
p	the training pattern
V_p	candidate output for input pattern p
$E_{p,o}$	network output error for output o, pattern p
$\overline{V}$	average of candidate output over all patterns
$\overline{E_o}$	average of output errors over all patterns



Train input weights - correlation

$$\frac{\delta S}{\delta w_i} = \sum_{p,o} \sigma_o (E_{p,o} - \overline{E_o}) f_p{}' I_{i,p}$$

w_i	weight to be updated
σ_o	sign of correlation
$f_p{}'$	derivative for pattern p of the nodes fn wrt of inputs
$\mathrm{I}_{\mathrm{i,p}}$	input candidate gets from unit i for pattern p
$\mathrm{E}_{\mathrm{p,o}}$	network output error for output o, pattern p
$\overline{E_o}$	average of output errors over all patterns

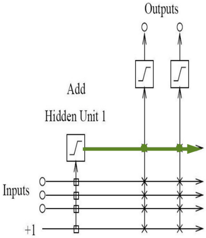


Step 1 Training input weights

- Candidate with highest correlation is put in the network, others are discarded.
- Correlation sign doesn't matter, only magnitude
- The bias unit effectively implements a learnable resting activation level for each hidden and output unit.
- Subsequent hidden neurons are attached to previous hidden neurons - this is where the cascade term comes from.

Step 2 Training output weights

1. Connect output of new node to output layer nodes. Use randomized weights that have adjusted sign to reduce error
2. Input weights are fixed
3. Only output weights are trained using any learning rule, until no improvement in error reduction
4. Once done the new node is fixed permanently.
5. Any new training involves adding another new node on it's own downstream layer



The Step Size Problem

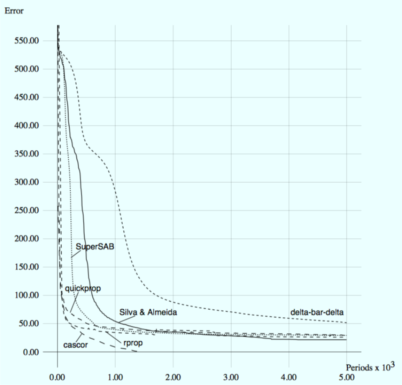
- Vanilla backpropagation requires small steps for convergence - slow
- We do not have the information to pick an optimal learning rate, manually selected

The Moving Target Problem

- Complication when many factors changing at the same time
- Error signal defines problem unit trying to solve, but this keeps changing
- Dramatic slowdown of training with increasing number of hidden layers
- Herd effect:
  - 2 tasks A, B. If A has bigger effect, all nodes redundantly train for A, ignoring B
  - But when all nodes move toward B at once, problem A response becomes worse.
  - Eventually nodes split to train for separate problems A and B, but it takes a long time
  - A randomly initialized network prevents nodes from behaving identically, but this tends to dissipate as the network is trained
- One way to combat: allow only a few weights to change while keeping the others constant

Pros

- At elast 10 times faster than standard backpropagation



- The network determines its own size and topologies
- Incremental/life long learning: new training, new information can be added, with an already trained network
- Effectively deals with the step size and moving target problems

Cons

- Very susceptible to overfitting

Applications

In spite of the many CoNN algorithms surveyed in (Kwok & Yeung, 1997a), the most popular for regression problems is no doubt the Cascade Correlation algorithm and maybe the second most popular is the DNC… The popularity of Cascade Correlation can be attested by the various ways this algorithm has inspired new variations and also has been used in the combined approaches between learning methods. [Franco et al, 2009]

The use of cascade-correlation neural networks in University fund raising

BK Wong*[1], TA Bodnovich[2] and VS-K Lai[3]

[1]*Lingnan University, China;* [2]*Youngstown State University, USA;* [3]*The Chinese University of Hong Kong, China*

In recent years, many Colleges and Universities in the USA have been facing a serious financial crisis since many state governments have reduced their support for higher education. There is no doubt that one of the solutions to this crisis depends on the successful implementation of University fund raising programs. Identifying the potential donors is an important part of this process. The objective of this research was to develop a cascade-correlation neural network to predict the types of people who would most likely be potential donors. A comparison of the classification accuracy between neural networks and multiple discriminant analyses (MDA) was also conducted. Our results indicated that neural networks could perform as well as MDA in overall accuracy. Furthermore, neural networks could predict with a lot more accuracy the actual donor (Type I hit) than MDA. Our study is the first published case study on the use of artificial neural networks for University fund raising.

Keywords: neural network; cascade-correlation; education; University fund raising

References I

S. E. Fahlman and C. Lebiere, “The cascade-correlation learning architecture.”

K. Khatter and J. Kaur, “Global Journal of Engineering Science and Research Management.”

S. K. Sharma and P. Chandra, “CONSTRUCTIVE NEURAL NETWORKS: A REVIEW,” International Journal of Engineering Science and Technology, vol. 1, no. 2, pp. 7847–7855.

L. Franco, D. A. Elizondo, and J. M. Jerez, Constructive Neural Networks. 2009.

T.-Y. Kwok and D. Y. Yeung, “Constructive feedforward neural networks for regression problems: A survey,” 1995.

T.-Y. Kwok and D.-Y. Yeung, “Constructive algorithms for structure learning in feedforward neural networks for regression problems,” IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 630–645, 1997.

G. Balázs, “Cascade-Correlation Neural Networks: A Survey.”

Y. Guo, L. Bai, S. Lao, S. Wu, and M. S. Lew, “A Comparison between Artificial Neural Network and Cascade-Correlation Neural Network in Concept Classification,” in Advances in Multimedia Information Processing – PCM 2014, 2014, pp. 248–253.

B. K. Wong, T. A. Bodnovich, and V. S.-K. Lai, “The Use of Cascade-Correlation Neural Networks in University Fund Raising,” The Journal of the Operational Research Society, vol. 51, no. 8, pp. 913–920, 2000.