Faculteit der Exacte Wetenschappen

(Duidelijk en met blokletters invullen)

15.10 | 10 2 | 10 10 | 5 | 4 | 19 | 95

Naam en voorletters:

Val[redacted]

Studentnummer:

Cijfer:

vrije Universiteit amsterdam

Datum: 12-08-03 Jaar van 1e inschrijving: 2000 Studierichting: INFORMATICA

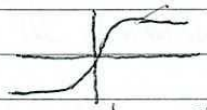
1)a) 15 Adaline is a linear separata that numerize the mean square error. It has a linear neuron model and uses the LMS algorithm. It is used for classificatie and regressen.

pseudo code:

n = 1;
w initialized randomly
while (E_tot unsatisfied and n < max_iter)
    select example (x, d)
    e(n) = d - x(n)
    w(n+1) = w(n) + η(n) · e(n) · x(n)
    n = n+1
end while

b) Perceptron and adaline [redacted] are both single layer neural networks. The Perceptron has a non-linear neuron model and the Adaline a linear neuron model. The algorithm is based on misclassified training examples. The one of adaline (perceptron) is based on the least mean squared. Perceptron is used for classificatie Adaline is also used for classificatie and for regressie.

2a) The activatie function used in the Backprop algorithm is the sigmoid activatie functie. It has the functie:

$$\varphi(v_j) = \frac{1}{1 - e^{av_j}}$$ and is like 

If you take a larger number for a the graph will be stretched out. if a goes to infiniti the functie becomes the sign function.

2b The FFNN creates "one line" with a single hidden layer so it can represent all linear functions (incl. XOR)

3a) The RBF has one hidden layer, with RBF activation function and one output layer with linear activation function. The Gaussian function is used as activation.

$$\varphi(r) = \exp\left(\frac{-R^2}{2\sigma^2}\right) \text{ with } R = \|x - t\|$$

b) parameters are: - centers (by by clustering)
- spread (by by normalization)
- weights (by by LMS)

4a) Unsupervised learning is to find an interesting structure in the unlabeled input examples.

b) initialization: weights are initialized and represent one neuron. each weight vector
sampling: take an x from the input examples
similarity matching: find a neuron, which has closes distance with x $\arg\min_i \|x(n) - w(n)\|$
updating: update the weight using Gaussian neighborhood function $w(n+1) = w(n) + \eta\, h_{i,d_{ij}}(x)(w(n) - x(n))$

continuation: $n = n+1$ until no noticable changes occur in the weights.

5 type of input: is mostly $\{-1, 1\}$ (or $\{0, 1\}$)
number of neurons: depends on the dimension of the input examples.
architecture: every neuron is connected to each neuron (recurrent), except to itself (no self feedback)
neuron state: the output at time n of a neuron
network state: a vector at time n with neurons states
activation function: is a sign function
weight properties: weights stabilise after a matter of time
weight computation: $\Delta w_{ij} = \frac{1}{M}\sum_{n=0}^{M} f_{ij} f_{ij}$
execution: execute until convergence