Neural Networks 2008/09, Second Exam, April 2009

Four problems are to be solved within 3 hours. **The use of supporting material (books, notes, calculators) is not allowed.** In total, you can achieve a maximum of 9 points, the grade for the exam will be determined as "1 + number of points".

---

1) Perceptron storage problem (2 points)

Consider a set of data $I\!D = \{\xi^\mu, S^\mu\}_{\mu=1}^P$ where $\xi^\mu \in I\!R^N$ and $S^\mu \in \{+1, -1\}$. In this problem, you can assume that $I\!D$ is homogeneously linearly separable.

a) Assume that you have found a solution w_1 of the perceptron storage problem which satisfies $w_1 \cdot \xi^\mu S^\mu \geq 1$ for all $\mu = 1, \ldots P$. Your partner in the practicals has found a vector w_2 with $w_2 \cdot \xi^\mu S^\mu \geq 5$ for all μ and claims that, obviously, this solution is *better* then yours. Do you agree or disagree? Give precise arguments for your conclusion!

b) Define precisely the following terms:
(I) the stability κ^μ of an example $\{\xi^\mu, S^\mu\}$
(II) the stability of a perceptron vector w
Provide a graphical illustration of (I) and (II) based on the geometrical interpretation of linearly separable functions.

c) While experimenting with the Rosenblatt perceptron in the practicals, your partner has another brilliant idea: the use of a larger learning rate. His/her argument: updating w by Hebbian terms of the form $\eta\, \xi^\mu\, S^\mu$ with a large η should give (I) faster convergence and (II) a better perceptron vector. Are you convinced? Give arguments for yor answer!

2) Learning a linearly separable rule (2 points)

Here we consider data $I\!D = \{\xi^\mu, S_R^\mu\}_{\mu=1}^P$ where noise free labels $S_R^\mu = \text{sign}[w^* \cdot \xi^\mu]$ are provided by an unknown teacher vector $w^* \in I\!R^N$ with $|w^*| = 1$.

a) Define the term *version space* in this context. Also provide a graphical illustration in terms of the *dual* geometrical interpretation discussed in class. Explain why the perceptron of optimal stability can be expected to give low generalization error.

b) Assume that a new, random input vectors $\xi \in I\!R^N$ is generated with equal probability anywhere on a hypersphere of constant radius $|\xi| = 1$. Given w^* and an arbitrary $w \in I\!R^N$, what is the probability for disagreement, $\text{sign}[w \cdot \xi] \neq \text{sign}[w^* \cdot \xi]$? You can "derive" the result from a sketch of the situation in $N = 2$ dimensions.

c) Define and explain the *Minover* algorithm for a given set of examples $I\!D$. Be precise, for instance by writing it in a few lines of *pseudocode*.

3) **Classification with multilayer networks (2 points)**

a) Explain the so-called committee machine with inputs $\xi \in I\!R^N$, K hidden units $\sigma_k = \pm 1, k = 1, 2, \ldots K$ and corresponding weight vectors $w_k \in I\!R^N$. Define the output $S(\xi)$ as a function of the input.

b) Now consider the so-called parity machine with N inputs and K hidden units. Define its output $S(\xi)$ as a function of the input.

c) Illustrate the case $K = 3$ for parity and committee machine in terms of a geometric interpretation. Why would you expect that the parity machine should have a greater storage capacity in terms of implementing random data sets $I\!D = \{\xi^\mu, S^\mu\}_{\mu=1}^P$.

4) **Regression and overfitting (3 points)**

a) Your partner in the practicals (again...) wants to use a multilayered neural network with N input nodes, K hidden units and 1 output node ($N-K-1$ architecture) in a regression problem. He/she suggests to use only linear activation functions in the entire network, in order to avoid overfitting effects. Why is this not a very convincing idea, in general? Write down the output as a function of the input and start your argument from there.

b) Explain the method of k-fold cross validation, for instance in terms of training a neural network from a given data set. How can you use cross validation to obtain information about *bias* and *variance* of the system? Explain also how cross validation can be employed for model selection.

c) Consider a feed-forward continuous neural network (N-2-1-architecture) with output

$$\sigma(\xi) = \sum_{j=1}^{2} v_j \, g(w^j \cdot \xi).$$

Here, ξ denotes an N-dim. input vector, w^1 and w^2 are N-dim. adaptive weight vectors in the first layer, and $v_1, v_2 \in I\!R$ are adaptive hidden-to-output weights. Assume the transfer function $g(x)$ has the known derivate $g'(x)$.

Given a single training example, i.e. input ξ^μ and label $\tau^\mu \in I\!R$, consider the quadratic error measure

$$\epsilon^\mu = \frac{1}{2}\left(\sigma(\xi^\mu) - \tau^\mu\right)^2.$$

Derive a gradient descent learning step for all adaptive weights with respect to the (single example) cost function ϵ^μ.