Neural Networks 07/08, Second Hertentamen, June 2008

This exam will only be accepted if it is at most the **second exam** you took in this academic year.
Four problems are to be solved within 3 hours. **The use of supporting material (books, notes, calculators) is not allowed.** In each of the four problems you can achieve up to 2.5 points, with a total maximum of 10 points.

1) Perceptron storage problem

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

a) Formulate the perceptron storage problem as the search for a vector $\boldsymbol{w} \in \mathbb{R}^N$ which satisfies a set of equations. Re-write the problem using a set of inequalities.

b) Assume that you have found a solution $\boldsymbol{w}_1$ of the storage problem satisfies $\boldsymbol{w}_1 \cdot \boldsymbol{\xi}^\mu S^\mu \geq 1$ for all $\mu = 1, \ldots P$. Your partner in the practicals claims he/she has found a vector $\boldsymbol{w}_2$ with $\boldsymbol{w}_2 \cdot \boldsymbol{\xi}^\mu S^\mu \geq 5$ for all μ and argues that, obviously, this solution is the *better one*. Do you agree or disagree? Please give precise arguments for your conclusion.

c) Again, consider two different solutions $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$ of the perceptron storage problem for a given data set $\mathbb{D}$. Assume furthermore that $\boldsymbol{w}^{(1)}$ can be written as a linear combination

$$\boldsymbol{w}^{(1)} = \sum_{\mu=1}^{P} x^\mu \, \boldsymbol{\xi}^\mu \, S^\mu \quad \text{with} \quad x^\mu \in \mathbb{R},$$

whereas the difference vector $\boldsymbol{w}^{(2)} - \boldsymbol{w}^{(1)}$ is orthogonal to all the vectors $\boldsymbol{\xi}^\mu \in \mathbb{D}$. Show that $\kappa(\boldsymbol{w}^{(1)}) \geq \kappa(\boldsymbol{w}^{(2)})$ holds for the stabilities. What does this result imply for the perceptron of optimal stability and potential training algorithms?

2) Learning a linearly separable rule

Here we consider data $\mathbb{D} = \{\boldsymbol{\xi}^\mu, S_R^\mu\}_{\mu=1}^P$ where noise free labels $S_R^\mu = \text{sign}[\boldsymbol{w}^* \cdot \boldsymbol{\xi}^\mu]$ are provided by a teacher vector $\boldsymbol{w}^* \in \mathbb{R}^N$ with $|\boldsymbol{w}^*| = 1$. Assume that by a training process we have obtained some perceptron vector $\boldsymbol{w} \in \mathbb{R}^N$.

a) Define precisely the terms *training error* and *generalization error* in the context of the situation, define and use an appropriate error measure.

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

1

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

3) Classification with multilayer networks

a) Explain the so-called committee machine with inputs $\boldsymbol{\xi} \in \mathbb{R}^N$, K hidden units $\sigma_k = \pm 1, k = 1, 2, \ldots K$ and corresponding weight vectors $\boldsymbol{w}_k \in \mathbb{R}^N$. 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 $\mathbb{D} = \{\boldsymbol{\xi}^\mu, S^\mu\}_{\mu=1}^P$.

4) Regression problems

a) Your partner in the practicals (again...) suggests to employ 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 network, in order to avoid overfitting effects. Why is this not a very convincing idea? Write down the output as a function of the input and start your argument from there. Name and explain at least one strategy which is used in practice to avoid overfitting in multilayered neural networks.

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

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

Here, $\boldsymbol{\xi}$ denotes an N-dim. input vector, $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are N-dim. adaptive weight vectors in the first layer, and $v_1, v_2 \in \mathbb{R}$ 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 $\boldsymbol{\xi}^\mu$ and label $\tau^\mu \in \mathbb{R}$, consider the quadratic error measure

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

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