

Neural Networks (2007/08)

Written exam, 28-01-2008

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 $\mathcal{D} = \{\xi^\mu, S^\mu\}_{\mu=1}^P$ where $\xi^\mu \in \mathbb{R}^N$ and $S^\mu \in \{+1, -1\}$.

- a) Define and explain the following statements:
 - a.1) \mathcal{D} is inhomogeneously linearly separable
 - a.2) \mathcal{D} is homogeneously linearly separableAlso provide graphical illustrations for $N = 2$.
- b) In this problem, you can assume that \mathcal{D} is homogeneously linearly separable. Define the stability $\kappa(\mathbf{w})$ of a perceptron solution \mathbf{w} with respect to the given set of data \mathcal{D} . Give a geometric interpretation and provide a sketch of an illustration. Explain in words why $\kappa(\mathbf{w})$ quantifies the stability of the perceptron output with respect to noise.
- c) Assume we have found two different solutions $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ of the perceptron storage problem for data set \mathcal{D} . Assume furthermore that $\mathbf{w}^{(1)}$ can be written as a linear combination

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

whereas the difference vector $\mathbf{w}^{(2)} - \mathbf{w}^{(1)}$ is orthogonal to all the vectors $\xi^\mu \in \mathcal{D}$.

Prove that $\kappa(\mathbf{w}^{(1)}) \geq \kappa(\mathbf{w}^{(2)})$ 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 linearly separable data $\mathcal{D} = \{\xi^\mu, S_R^\mu\}_{\mu=1}^P$ where noise free labels $S_R^\mu = \text{sign}[\mathbf{w}^* \cdot \xi^\mu]$ are provided by a teacher vector $\mathbf{w}^* \in \mathbb{R}^N$ with $|\mathbf{w}^*| = 1$. Assume that by some training process perceptron vector $\mathbf{w} \in \mathbb{R}^N$ is obtained.

- a) Define and explain the term *version space* in this context, provide a graphical illustration. Give an argument why one can expect the perceptron of maximum stability to display good generalization behavior.

- b) Assume that random input vectors $\xi \in \mathbb{R}^N$ are generated with equal probability anywhere on a hypersphere of constant radius $|\xi| = 1$. Given w^* and an arbitrary $w \in \mathbb{R}^N$, what is the probability for disagreement, $\text{sign}[w \cdot \xi] \neq \text{sign}[w^* \cdot \xi]$? You should “derive” the result from a sketch of the situation in $N = 2$ dimensions.
- c) Define and explain the (Rosenblatt) *Perceptron* algorithm for a given set of examples \mathcal{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 $\xi \in \mathbb{R}^N$, K hidden units $\sigma_k = \pm 1, k = 1, 2, \dots, K$ and corresponding weight vectors $w_k \in \mathbb{R}^N$. 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 $\mathcal{D} = \{\xi^\mu, S^\mu\}_{\mu=1}^P$.

4) Regression problems

- a) Explain the term *overfitting* in terms of a simple regression problem (e.g. polynomial regression). What is the meaning of *bias* and *variance* in this context? Be precise! What is the bias-variance-dilemma?
- b) 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 $v_1, v_2 \in \mathbb{R}$ 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 ξ^μ and label $\tau^\mu \in \mathbb{R}$, consider the quadratic error measure

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

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