T-61.231 Principles of Pattern Recognition

Exercise 7: 11.11.2002

1. When a MLP is used for a classification task, the number of output units is usually same as the number of classes. The desired output is zero for all but one neuron at a time and each output neuron corresponds to one of the classes. The input is classified into that class whose output neuron is most active.

   Let us consider a single output neuron whose output is $y(x)$ when the input of the network is x and the desired output is d. The cost functional concerning only this single neuron and which is minimized by back-propagation algorithm has the following form:

   $$J = \frac{1}{N} \sum_{k=1}^{N} \left(y(x^k) - d^k \right)^2 \ ,$$

   where N is the number of learning samples. If N is very large, the cost functional approximates the following expectation:

   $$J = E_{x,d}[(y(x) - d)^2] \ .$$

   Show that the solution which minimizes the cost functional is the optimal discriminant function of Bayes classifier:

   $$y(x) = P(d = 1|x) \ .$$

2. Output of the perceptron unit is y and its inputs $x_1, ..., x_n$ are continuous-valued. Neuron calculates its output according to the following function:

   $$y = \tanh(\sum_{i=1}^{n} w_i x_i - \theta) \ .$$

   Neuron tries to learn to give desired output d for inputs $x_1, ..., x_n$. One method to do this is to minimize function $(y - d)^2$. When a gradient descent method is used for the minimization task, it can be shown that the gradient step has the form $\Delta w_i = f(y, d) x_i$. Derive function $f(y, d)$ in this case.

3. Let us consider back-propagation algorithm in a 2-layer MLP, which has 2 neurons in both output layer and hidden layer and 2 inputs. W_{ij} are the weights of the output layer and Θ_j are the offset parameters, where $j = 1, 2$ is the index of the neuron and $i = 1, 2$ the index of the hidden unit, where the input comes from. Similarly w_{kl} and θ_l are the weights and offsets of the hidden layer. All neurons have 'logsig' as an activation function.

   Derive back-propagation algorithm to update all the parameters. Let us assume on-line learning, which means that the network learns immediately after the new (input,output)-pair has been given.

4. Show that if the cost function, optimized by a multilayer perceptron, is the cross-entropy

$$J = -\sum_{i=1}^{N} \sum_{k=1}^{k_L} y_k(i) \ln \frac{\hat{y}_k(i)}{y_k(i)}$$

and the activation function is the sigmoid $f(x) = \frac{1}{1+\exp(-ax)}$, then the gradient

$$\delta_j^L(i) = \frac{\partial \mathcal{E}(i)}{\partial v_j^L(i)}$$

becomes $\delta_j^L(u) = a(1 - \hat{y}_j(i))y_j(i)$. *(Theodorodis 4.6, p. 130)*

5. Repeat the previous problem for the softmax activation function

$$\hat{y}_k(i) = \frac{e^{v_k^L}}{\sum_{k'=1}^{L} e^{v_{k'}^L}} \ .$$

(Theodorodis 4.7, p. 130; note the (probable) error in the book's exercise (the answer is $\hat{y}_j(i)y_j(i) - y_j(i)$*))*

6. The following scheme for adaptation for the learning parameter μ has been proposed by C. Darken and J. Moody (1991):

$$\mu = \mu_0 \frac{1}{1 + \frac{t}{t_0}}$$

Verify that, for large enough values of t_0 (eg. $300 \leq t_0 \leq 500$), the learning parameter is approximately constant for the early stages of training (small values of iteration step t) and decreases in inverse proportion to t for large values. The first phase is called *search phase* and the latter *convergence phase*. Comment on the rationale of such a procedure. *(Theodorodis 4.16, p. 131)*