Exercises for
Pattern Recognition
Peter Fischer, Shiyang Hu
Assignment 6, 25./28. 11.2014

SOLUTION

General Information:

Exercises (1 SWS):	Tue 12:15 – 13:45 (0.154-115) and Fri 08:15 – 09:45 (0.151-115)
Certificate:	Oral exam at the end of the semester
Contact:	peter.fischer@fau.de, shiyang.hu@fau.de

Perceptron and MLP

Exercise 1 Create a classifier that implements *Rosenblatt's Perceptron Learning Algorithm*. You have to create the necessary files for the toolbox and make it accessible to the Matlab GUI.

(a) Add the appropriate line to the file `Classification.txt` before the `None` statement
(you may have to double click classifier.mat to see the new line in the algorithm list):
`RPerceptron@Num of iterations:@500@S` (for example)

(b) Create an `.m`-File that has the same name as the classifier you already specified and create the according classifier function with return values and arguments.

(c) Implement the stochastic gradient descent algorithm and test it with varying numbers of iterations and feature sets.
See RPerceptron.m

Exercise 2 This exercise deals with the Multi-Layer Perceptron (MLP). There are several activation functions that can be used with MLP, e.g. linear, sigmoid and hyperbolic tangent.

(a) The activation function of the output layer often differs from the activation function of the hidden layers. What kind of output layer activation function would you use for regression? What is your choice for classification?

- Regression: linear output activation (real output variables)
- Binary classification: depends on class labels
  - 0,1 classes $\rightarrow$ logistic sigmoid
  - -1,1 classes $\rightarrow$ hyperbolic tangent
  - Multiclass classification: same as binary classification, but multiple output nodes (1-of-K coding)

(b) Show that a two-layer neural network with linear activation function in the hidden units is equivalent to a single-layer neural network. Hint: compute $\text{net}_j^{(2)}$ as a function of the inputs x_k.
The activation function is linear, i.e. $y_j^{(l)} = \text{net}_j^{(l)}$.

$$\begin{aligned}
\text{net}_j^{(2)} &= \sum\nolimits_{i=1}^{M^{(1)}} y_i^{(1)} w_{ij}^{(2)} - w_{0j}^{(2)} \\
&= \sum\nolimits_{i=1}^{M^{(1)}} \text{net}_i^{(1)} w_{ij}^{(2)} - w_{0j}^{(2)} \\
&= \sum\nolimits_{i=1}^{M^{(1)}} \left[\sum\nolimits_{k=1}^{M^{(0)}} y_k^{(0)} w_{ki}^{(1)} - w_{0i}^{(1)}\right] w_{ij}^{(2)} - w_{0j}^{(2)} \\
&= \sum\nolimits_{i=1}^{M^{(1)}} \left[\sum\nolimits_{k=1}^{M^{(0)}} x_k w_{ki}^{(1)} - w_{0i}^{(1)}\right] w_{ij}^{(2)} - w_{0j}^{(2)} \\
&= \sum\nolimits_{k=1}^{M^{(0)}} x_k \sum\nolimits_{i=1}^{M^{(1)}} w_{ki}^{(1)} w_{ij}^{(2)} - \sum\nolimits_{i=1}^{M^{(1)}} w_{0i}^{(1)} w_{ij}^{(2)} - w_{0j}^{(2)} \\
&= \sum\nolimits_{k=1}^{M^{(0)}} x_k p_{kj} - p_{0j}
\end{aligned}$$

This property holds for MLP with linear activation functions in general. Explanation: the composition of linear transformations is itself a linear transformation.

(c) Consider a neural network with a hyperbolic tangent activation function in the hidden units

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

By computing the relationship between $\tanh(a)$ and $\sigma(a)$, show that there exists an equivalent network which uses logistic sigmoid activations

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

Necessary identity for $\sigma(-a)$

$$\begin{aligned}
\sigma(-a) &= \frac{1}{1+e^a} + 1 - 1 = \frac{1}{1+e^a} - \frac{1+e^a}{1+e^a} + 1 \\
&= \frac{1 - 1 - e^a}{1+e^a} + 1 = -\frac{e^a}{1+e^a} + 1 \\
&= 1 - \frac{1}{1+e^{-a}} = 1 - \sigma(a)
\end{aligned}$$

Relate $\tanh(a)$ and $\sigma(a)$

$$\begin{aligned}
\tanh(a) &= \frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{1 - e^{-2a}}{1 + e^{-2a}} \\
&= \frac{1}{1+e^{-2a}} - \frac{e^{-2a}}{1+e^{-2a}} = \sigma(2a) - \sigma(-2a) \\
&= \sigma(2a) - 1 + \sigma(2a) = 2\sigma(2a) - 1
\end{aligned}$$