Artificial Neural Networks (Gerstner). Exercises for week 1

Week 1: Simple Perceptrons, Geometric interpretation, Discriminant function

1. Gradient of quadratic error function

We define the mean square error in a data base with P patterns as

$$E^{\mathrm{MSE}}(\mathbf{w}) = \frac{1}{2}\frac{1}{P}\sum_{\mu}[t^{\mu} - \hat{y}^{\mu}]^2 \tag{1}$$

where the output is

$$\hat{y}^{\mu} = g(a^{\mu}) = g(\mathbf{w}^T\mathbf{x}^{\mu}) = g(\sum_k w_k x_k^{\mu}) \tag{2}$$

and the input is the pattern $\mathbf{x}^{\mu}$ with components $x_1^{\mu}\dots x_N^{\mu}$.

(a) Calculate the update of weight w_j by gradient descent (batch rule)

$$\Delta w_j = -\eta\,\frac{dE}{dw_j} \tag{3}$$

Hint: Apply chain rule

(b) Rewrite the formula by taking one pattern at a time (stochastic gradient descent). What is the difference to the batch rule? What is the geometric interpretation? Compare with the perceptron algorithm!

2. Perceptron Algorithm as stochastic gradient descent.

We work in $n+1$ dimensions (in other words the threshold is integrated in the weight vector). We define the Heaviside step function $\theta(x) = 1$ for $x > 0$ and zero otherwise. We define $\tilde{t}^{\mu} = 2(t^{\mu} - 0.5)$. Hence $\tilde{t} = \pm 1$.

The classic perceptron has a gain function $g(a) = \theta(a)$.

Show that the perceptron algorithm performs stochastic gradient descent on the (piecewise) linear error function:

$$E^{\mathrm{perc}}(\mathbf{w}) = -\sum_{\mu}\left[\tilde{t}^{\mu}a^{\mu}\right]\theta\left[-\tilde{t}^{\mu}a^{\mu}\right] \tag{4}$$

where $a^{\mu} = \sum_k w_k x_k^{\mu}$.

Hint: Show first that the error function vanishes for all patterns that are correctly classified. It is non–negative for misclassified patterns.

3. Apply the Perceptron Algorithm

A data base $(\mathbf{x}^{\mu}, t^{\mu})$ with $1 \le \mu \le 6$ contains three positive examples ($t^{\mu} = +1$) at the points
$x^1 = (2,-1); x^3 = (2, 0.5); x^5 = (0.5,-1)$
and three negative examples ($t^{\mu} = 0$) at the points
$x^2 = (-1,1); x^4 = (0.2,-0.2); x^6 = (2,1)$.

(a) Construct one of the possible separating hyperplanes by hand. Express the parameters of the hyperplane in terms of the three weight parameters (where the third weight parameter is the threshold) of the perceptron.

(b) A perceptron algorithm which takes patterns sequentially one after the other starting with pattern $\mu = 1$ is applied to the above problem using an initialization $w = (1, 0)$ and threshold $\theta = 0$.

Consider a learning rate $\eta = 2$ and give the resulting weight vector during the first 6 steps of the iteration.

(c) Draw a figure on paper and plot the separating hyperplane before learning $\mathbf{w}(0)$, and immediately after the first change $\mathbf{w}(1)$, (not every iteration leads to a change)

(d) repeat the same with learning rate $\eta = 0.5$ and plot the result in a separate figure.

4. Apply the stochastic gradient descent algorithm from exercise 1

Use a transfer function $g(a) = 1/[1 + exp(-\beta a)]$ and take patterns from Exercise 3 in the same order as you did for the perceptron algorithm. What are the differences to the perceptron algorithm? What are the problems?

Hint: Choose first $\beta = 0.5$ and then $\beta = 5$. Consider a pattern that is correctly classified and one that is misclassified, for example patterns 5 and 6. Evaluate the gradient g' and comment on its absolute value for cases where a pattern is misclassified.

5. Adaline algorithm

A friend of yours argues as follows: in a perceptron with $\hat{y}^\mu = \theta(a^\mu)$, a sufficient condition that all patterns are correctly classified is that $a^\mu = +1$ for all positive examples $t^\mu = +1$, and $a^\mu = -1$ for all negative examples $t^\mu = 0$.

He therefore suggests to directly optimize the error function

$$E^{\mathrm{Ada}}(\mathbf{w}) = \frac{1}{2} \sum_\mu \left[\tilde{t}^\mu - a^\mu \right]^2 \tag{5}$$

where $\tilde{t}^\mu = 2(t^\mu - 0.5)$ and $a^\mu = \sum_k w_k x_k^\mu$.

To convince him that this error function is not a good idea, show him the following example.

A data base $(\mathbf{x}^\mu, t^\mu)$ with $1 \le \mu \le 12$ contains six positive examples ($t^\mu = +1$) at the points $x^1 = (1, 1); x^2 = (1, -1); x^3 = (\alpha, 2); x^4 = (\alpha, 1); x^5 = (\alpha, -2); x^6 = (\alpha, -1)$ where $\alpha \ge 1$ is a parameter; and six negative examples ($\tilde{t}^\mu = -1$) at the points
$x^7 = (-1, 3); x^8 = (-1, 2); x^9 = (-1, 1); x^{10} = (-1, -1); x^{11} = (-1, -2); x^{12} = (-1, -3)$.

(a) Plot the points.

(b) Choose a weight vector $(w, 0, b)$ that gives rise to a linear discriminant function $d(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ which separates positive and negative examples. Does your solution depend on the choice of α?

(c) Insert the data points and the weight vector $(w, 0, b)$ with arbitrary w and b into the error function E^{Ada} and find those w and b that minimize the error function. Express your result as a function of α.

(d) Determine the maximal value of the parameter α for which the optimization in (c) still gives a correct discriminant function.

(e) Conclude the argument.

Remark: $E^{\mathrm{Ada}}(\mathbf{w})$ was called the error function of the historical 'Adaline' algorithm.