

**Maschinelles Lernen und Data Mining**  
Sommersemester 2009  
**Übungsblatt 1**

*Besprechung des Übungsblattes am 07.05.2009*

**Aufgabe 1-1** Die ADALINE Lernregel  
*schriftlich bearbeiten*

Der Perceptron Trainings-Algorithmus kann verallgemeinert werden zum *least mean square* Verfahren. Dieses definiert die Kostenfunktion für  $N$  Trainingselemente als

$$\text{cost} = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

wobei  $y_i$  das tatsächliche und  $\hat{y}_i$  das berechnete Klassenlabel von Muster  $i$  ist.

Diese Kostenfunktion wird im *adaptive linear element* (ADALINE) Modell angewendet: Es verwendet analog zum einfachen Perceptron die Aktivierungsfunktion  $h_i = \sum_{j=0}^{M-1} w_j x_{i,j}$ , jedoch erfolgt die Klassifizierung nicht über eine Signum-Funktion sondern direkt:  $\hat{y} = h$ .

(Zur Erinnerung:  $M$  ist die Anzahl der Merkmale der Muster  $x_i \in \mathbb{R}^M$  und des Gewichtungsvektors  $w \in \mathbb{R}^M$ , wobei  $x_0 = 1$  konstant ist und dem Bias entspricht.)

- Leiten Sie ein gradientenbasiertes Lernverfahren für einen ADALINE Prozess ab. (Optimieren Sie analog zur Perceptron-Lernregel nach  $w$ .)
- Geben Sie hierzu eine musterbasierte Lernregel an.
- Welche Vorteile haben musterbasierte Lernregeln?
- Nennen Sie die wichtigsten Unterschiede des ADALINE-Modells zur in der Vorlesung kennengelernten Perceptron-Lernregel.

**Aufgabe 1-2** Anwendung der Perceptron-Lernregel  
*schriftlich bearbeiten*

Gegeben seien zwei Klassen  $A$  und  $B$ , die jeweils 2 Muster umfassen:

$$A = \left\{ p_1 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, p_2 = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \right\}, B = \left\{ p_3 = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix}, p_4 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \right\}$$

Die Klassenlabels  $y$  der Klasse  $A$  sind deklariert mit 1, die von  $B$  mit  $-1$ .

Lösen Sie die folgenden Aufgaben mittels einer Programmiersprache Ihrer Wahl oder mit der guten alten Pen-and-Paper-Methode und geben Sie die Zwischenschritte graphisch mit an: *b.w.*

- a) Wieviele Iterations-Schritte benötigt die musterbasierte Perceptron-Lernregel um die Klassen  $A$  und  $B$  zu separieren wenn der Gewichtungsvektor  $w$  mit  $(0, 1, -1)$  initialisiert wird und eine Schrittweite  $\eta = 0.1$  verwendet wird?
- b) Wieviele Iterations-Schritte brauchen wir für  $\eta = 0.25$ ? Spielt hierbei die Reihenfolge der betrachteten Muster eine Rolle? Wenn ja geben Sie ein Beispiel zweier unterschiedlicher Einfügereihenfolgen, ansonsten beweisen Sie das Gegenteil.
- c) Nach wievielen Iterationen terminiert die gradientenbasierte Lernregel jeweils für beide  $\eta$ ? Spielt hier die Einfügereihenfolge eine Rolle?

*Kleiner Hinweis:* Sie sollten für die korrekte Bearbeitung dieser Aufgaben nicht mehr als jeweils 10 Iterationen benötigen.