

# Vorlesung Adaptive Systeme WS 13/14

## Übungsblatt 5

Ausgabe: 26.11.2013

Abgabe: 03.12.2013

### Adaptive Systeme 1

#### **Aufgabe 5.1 Approximieren & Stopped Training (10 Punkte)**

Bisher haben wir mit unseren neuronalen Netzen nur klassifiziert, nun wollen wir uns „Stopped Training“ ansehen und bei der Gelegenheit eine Gerade approximieren.

Laden Sie sich von der Webseite das zip-Archiv mit Datensätzen herunter. Es enthält 4 Dateien:

- function\_training.txt (Trainingsdaten)
- function\_validation.txt (Validierungsdaten)
- function\_test.txt (Testdaten)
- function\_original.txt (Originalgerade)

Trainieren Sie ein AdaLinE darauf, die Gerade, die in function\_original.txt dargestellt ist, zu approximieren. Verwenden sie zum Training jedoch nur die Trainingsdaten! Dieser Datensatz enthält zufällige Punkte aus der Geraden, die zufällig verrauscht wurden.

Achtung: Als Ausgabefunktion benutzen Sie  $S(z) = z$ , da wir diskrete Ausgabewerte benötigen und keine binäre Klassifikation!

Als Trainingsmethode benutzen wir „Stopped-Training“. Bilden Sie nach jedem Trainingsmuster den mittleren, quadratischen Fehler über die Menge der Validierungsdaten. Beenden sie das Training sobald dieser nicht mehr sinkt, sondern wieder steigt. Falls der Fehlerverlauf sehr unregelmäßig ist und ihr Training dadurch zu früh endet, bilden sie einen gleitenden Mittelwert, z.B. über die letzten 5 Werte.

Nach dem Training geben sie den erwarteten quadratischen Fehler für Trainings-, Validierungs- und Testmenge aus. Stellen sie auch die gewünschte und die gelernte Gerade grafisch dar (matplotlib).

#### **Aufgabe 5.2 Oja, Kovarianzmatrix und Eigenvektoren (10 Punkte)**

Nehmen Sie die verrauschten Daten aus Aufgabe 5.1 (also alles außer function\_original) und bilden sie eine gemeinsame Menge. Berechnen Sie für diese Daten die Kovarianzmatrix (kann mithilfe eines Programms ermittelt werden). Wenden sie dann die Oja Lernregel auf den Datensatz an. Stellen Sie die gelernten Gewichte grafisch (als Gerade) dar.

Geben Sie den gefundenen Eigenvektor und den dazugehörigen Eigenwert an.

## Adaptive Systeme 2

### **Aufgabe 5.3 Stochastische Klassifikation** (10 Punkte)

Gegeben sind die folgenden Wahrscheinlichkeiten bezüglich Kinofilmen\*:

$P(\text{Film basiert nicht auf einem Roman})$ : 0.7  
 $P(\text{Film ist gut} \mid \text{Film basiert auf einem Roman})$ : 0.05  
 $P(\text{Film ist gut} \mid \text{Film basiert nicht auf einem Roman})$ : 0.2  
 $P(\text{Film bekommt Fortsetzung} \mid \text{Film ist gut})$ : 0.5  
 $P(\text{Film bekommt Fortsetzung} \mid \text{Film ist schlecht})$ : 0.4

- Wie groß ist die Wahrscheinlichkeit dass ein guter Film auf einem Buch basiert?
- Wie groß ist die Wahrscheinlichkeit, dass ein Film der eine Fortsetzung bekommt, gut ist?

\* Im Sinne von „Traue keiner Statistik die du nicht selbst gefälscht hast!“ sind alle hier genannten Daten frei erfunden.

### **Aufgabe 5.4 ROC-Kurven** (10 Punkte)

Laden Sie sich von der Webseite die neuen Irisdaten herunter und trainieren Sie ein Perzeptron darauf diese Muster nach den Merkmalen PetalLength und PetalWidth zu klassifizieren. *Achtung*: Diese Daten sind nicht linear separierbar! Trainieren Sie das Netz solange bis es 90% der Muster richtig klassifiziert.

Erstellen sie nun eine ROC-Kurve mit 100 Datenpunkten für ihr neuronales Netz. Als Parameter passen Sie das Bias-Gewicht an. Stellen sie die ROC-Kurve grafisch dar.

### **Aufgabe 5.x Bonusaufgabe** (5 Punkte)

Ermitteln Sie näherungsweise die AUC ihres Systems.