

11. Übung zur Vorlesung Künstliche Intelligenz

Institut für Informatik, FU Berlin, SoSe 2005
Prof. Dr. Raúl Rojas, Marco Block, Ernesto Tapia

Neben der schriftlichen Abgabe, sind die Programmieraufgaben **zusätzlich** per e-mail an den Tutor zu schicken. Eine e-mail ersetzt nicht die schriftliche Abgabe! Die folgenden Programmieraufgaben können mit Prolog, C, C++, C#, Delphi oder Java bearbeitet werden. Die beschriebenen Daten sind verfügbar unter

"<http://www.inf.fu-berlin.de/lehre/SS05/KI/Daten/>".

1. Aufgabe (10 Zusatzpunkte)

In der Vorlesung wurde das Gradientenabstiegsverfahren (*Backpropagation*-Algorithmus) vorgestellt. Sie sollen dieses Verfahren implementieren. Dazu bauen Sie ein Neuronales Netz mit 3 Schichten, bestehend aus der Inputschicht ($N + 1$ Neuronen), der versteckten Schicht (26 Neuronen) und der Outputschicht (10 Neuronen). Als Aktivierungsfunktion verwenden Sie die Sigmoidfunktion $s_c : \mathcal{R} \rightarrow (0, 1)$ mit $c = 1$,

$$s_c(x) = \frac{1}{1 + e^{-cx}}.$$

Die Ableitung, die Sie für das Verfahren brauchen lautet:

$$s'_1(x) = s_1(x)(1 - s_1(x)).$$

Die Gewichte werden zufällig gewählt. In der Trainingsphase sollen nun die Gewichte "*online*" gelernt werden, das heißt, ein Datenvektor \vec{x} wird zufällig aus der Trainingsmenge (*pendigits-training-norm.txt*) gewählt und dient als Eingabe für das Netz. Im Feedforward-Schritt wird nun der Ausgabe-Fehler bestimmt. Die Gewichte werden entsprechend der Vorschrift

$$w_{ij}^{(k)} = w_{ij}^{(k)} - \gamma \frac{\partial E}{\partial w_{ij}^{(k)}}, \text{ mit } 0 < \gamma < 1$$

angepasst. Usw. Denken Sie daran, dass die Lernkonstante γ nach jedem Schritt kleiner werden sollte.

- Trainieren Sie das Netz mit diesen Trainingsdaten: *pendigits-training-norm.txt*
- Überprüfen Sie die Qualität des gelernten Netzes mit diesen Daten: *pendigits-testing-norm.txt* und geben Sie die Erkennungsrate prozentual an.

Abgabe: (Dienstag) 12.07.2005, 12:00 Uhr (s.t.)
(verspätete Abgaben werden nicht mehr entgegen genommen!)