



Neuronale Netze, Sommersemester 2012

Übungsblatt 4

Abgabe: 24.05.2011, Besprechung: 31.05.2011

Aufgabe 10 Encoder Netzwerke [10 Punkte]

Die Klasse der N-M-N-Encoder-Netzwerke ist eine Familie von feedforward-Netzwerken mit jeweils N Eingabe-, M versteckten und N Ausgabezellen, wobei jede Schicht von Zellen vollständig mit der nächsten verbunden ist (keine „shortcut connections“). Die Eingabe des Netzwerks sind N binäre Vektoren, wobei jeder Vektor eine 1 und sonst nur Nullen enthält, z.B. $\underline{x}_3 = (0, 0, 1, 0, 0, 0, 0, 0)$. Die erwünschte Ausgabe sind die gleichen Vektoren (bzw. Näherungen davon, da float-Werte). Das Problem ist hierbei, daß die Zwischenschicht viel weniger Zellen enthält als die Eingabe- und Ausgabeschicht, üblicherweise $M \approx \log_2 N$. Das heißt, das Netzwerk muß aus der Eingangs-/Ausgangs-Repräsentation selbständig eine viel kompaktere Repräsentation in den hidden units bilden und gleichzeitig auch noch die Umkehrabbildung dieser kompakten Repräsentation in die ursprüngliche Repräsentation lernen. Man nennt Encoder der Art N-M-N eng (tight), falls $M = \log_2 N$; sehr eng (ultra tight), falls $M < \log_2 N$ und lose (loose), falls $M > \log_2 N$. Wir betrachten hier als konkrete Beispiele die 8-2-8, 8-3-8- und 8-4-8-Encoder.

Laden Sie JavaNNS von <http://www.cogsys.cs.uni-tuebingen.de/software/JavaNNS> herunter und installieren Sie es auf Ihrem Rechner. Damit das Programm funktioniert, muss es in einer 32bit Umgebung ausgeführt werden, d.h. es muss auch ein 32bit Java verfügbar sein. Starten Sie JavaNNS im Installationsverzeichnis und machen Sie sich mit dem Programm vertraut. Anleitungen finden Sie auf der Webseite. Öffnen Sie das Error Graph Fenster in JavaNNS und das Log Fenster, um den Trainingsvorgang verfolgen zu können. Kopieren Sie die Datei *encoder.pat* aus den *examples* in Ihr Arbeitsverzeichnis.

- (a) Finden Sie durch Versuche eine optimale Wahl der Lernrate η (LEARN, 1. Parameter) heraus, bei dem der 8-3-8-Encoder mit dem Lernverfahren Backpropagation und der logistischen Aktivierungsfunktion den Fehler $SSE/out = 0.1$ in möglichst kurzer Zeit erreicht. Führen Sie für jede Parameterwahl drei Versuche mit jeweils zufälliger Neuinitialisierung der Gewichte aus und notieren Sie die mittlere Anzahl der erforderlichen Lernzyklen in einer Tabelle. (3 Punkte)
- (b) Notieren Sie für einen der trainierten 8-3-8-Encoder die Repräsentation in der verdeckten Schicht (Aktivierungen der verdeckten Neuronen) für alle Eingangsmuster. (2 Punkte)
- (c) Notieren Sie für einen der trainierten 8-2-8-Encoder die Repräsentation in der verdeckten Schicht (Aktivierungen der verdeckten Neuronen) für alle Eingangsmuster. Wieso funktioniert der 8-2-8 Encoder? (2 Punkte)
- (d) Verwenden Sie die zuvor bestimmten Parameter für das Training eines losen 8-4-8-Encoders. Notieren Sie die mittlere Anzahl der erforderlichen Trainingszyklen. Notieren Sie die Repräsentation der verdeckten Schicht. (2 Punkte)

Aufgabe 11 Implementierung von Backpropagation [10 Punkte]

Laden Sie sich von der Übungsseite zur Vorlesung die Datei `MultiLayerPerceptron.zip` herunter. Das Archiv enthält das Grundgerüst einer JAVA-Klasse `MultiLayerPerceptron` für mehrschichtige Perzeptrons.

- (a) Bitte erklären Sie anhand des schon vorhandenen Codes, wie die Gewichte abgespeichert werden. Gibt es Vorteile gegenüber einer einfachen großen Gewichts-Matrix? (1 Punkt)
- (b) Implementieren sie die logistische Aktivierungsfunktion und ihre Ableitung. (2 Punkte)
- (c) Vervollständigen Sie den Code in der Methode `predict`. Tips: Vergessen Sie das On-Neuron nicht. Schauen Sie sich die Formeln für Backpropagation vorher noch einmal genau an. Welche Werte können sie im Trainingsschritt wiederverwenden? (2 Punkte)
- (d) Vervollständigen Sie den Code in der Methode `trainpattern`. (3 Punkte)
- (e) Testen Sie Ihren Code bitte mit `mlptraininput.txt/mlptrainoutput.txt` (Trainingsmenge) und `mlptestinput.txt/mlptestoutput.txt` (Testmenge) mit verschiedenen Topologien:
 - Topologie mit einer Zwischenschicht mit 10 Neuronen (2-10-2)
 - eine Topologie mit zwei Zwischenschichten (2-10-4-2)

Trainieren Sie über 10.000 Iterationen. Kopieren Sie die Ausgabe jeweils in eine Datei `output-2-10-2.txt` bzw. `output-2-10-4-2.txt`. Bitte achten Sie darauf, dass Sie vorher den von Ihnen hinzugefügten Debug-Output ausschalten. (2 Punkte)

Schicken Sie das Programm (`MultiLayerPerceptron.java`), die Dateien `output-2-10-2.txt` und `output-2-10-4-2.txt` an Yasir Khan (yasir.khan@uni-tuebingen.de) und Lars Rosenbaum (lars.rosenbaum@uni-tuebingen.de).

Die Betreff-Zeile lautet diesmal "NN-Uebung04".