

9. Übungszettel Mustererkennung WS15/16

Prof. Raúl Rojas, Fritz Ulbrich
Institut für Informatik, Freie Universität Berlin
Abgabe Online bis Mittwoch, 20.01.16, 10 Uhr

Bitte laden Sie ihre Lösung der Aufgaben als **pdf-Datei** hoch.
Quellcode können Sie optional als Archiv anhängen.

1. Aufgabe (5 Punkte):

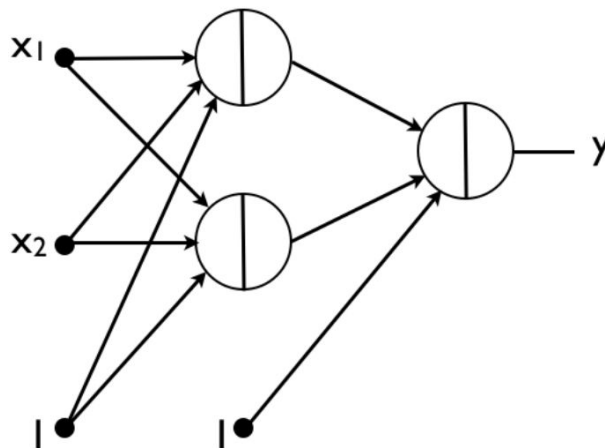
Implementieren Sie ein Neuronales Netz (siehe Abbildung) und lernen Sie mit dem Backpropagation-Algorithmus die **XOR-Funktion** (<http://de.wikipedia.org/wiki/Kontravalenz>). Verwenden Sie dabei die Matrix Form (Neural Networks, Kapitel 7.3: <http://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf>).

Verwenden Sie die batch/offline Variante für die Anpassung der Gewichte.

Verwenden Sie als Lernrate $\gamma = 10.0$.

Trainieren Sie das Netz bis die Summe der Fehler $E = \frac{1}{2} \sum (o_i^{(2)} - y_i)^2$ in einem Durchlauf kleiner als 0.00001 ist.

- (3 Punkte) Geben Sie die gelernten Gewichte an.
- (2 Punkte) Ordnen Sie jedem der 3 Knoten anhand der gelernten Gewichte eine logische Funktion zu (z.B. $(x_1 \wedge \neg x_2)$ oder $(o_1^{(1)} \vee o_2^{(1)})$).



2. Aufgabe (5 Punkte):

Laden Sie die Dateien **pendigits-testing.txt** und **pendigits-training.txt** aus dem Resources-Ordner der KVV Seite zur Vorlesung herunter. Jede Zeile dieser Dateien ist ein Datensatz für einen Linienzug einer Ziffer bestehend aus 17 Zahlen, die durch Leerzeichen getrennt sind. Die ersten 16 Zahlen sind 8 X/Y-Koordinatenpaare. Die letzte Zahl ist die Ziffer, die der Linienzug darstellen soll.

Trainieren Sie ein neuronales Netz für die Klassifizierung der Ziffern anhand der Daten in **pendigits-training.txt**. Verwenden Sie dabei den folgenden Aufbau:

16 Eingabeknoten, k Knoten der mittleren Schicht (Hidden Layer), sowie 10 Ausgabeknoten. (Für die Eingabeknoten und den Hidden Layer kommt jeweils noch ein konstanter Term dazu)

Verwenden Sie die online Variante für die Anpassung der Gewichte.

- a) (3 Punkte) Trainieren Sie das Netz mit jeweils $k = 2, 4$ und 16 Knoten in der mittleren Schicht. Verwenden Sie dabei als jeweils Lernrate $\gamma = 1.0, 0.1$ und 0.01 . (Es gibt also insgesamt 9 verschiedene Kombinationen der Parameter)

Plotten Sie für jede der 9 Kombinationen die Summe der Fehler $E = \frac{1}{2} \sum (o_i^{(3)} - y_i)^2$ für jeweils 100

Durchläufe als Graph. (Ein Durchlauf = alle Trainingsdaten einmal abgearbeitet)

- b) (2 Punkte) Klassifizieren Sie die Ziffern aus **pendigits-testing.txt** mit zwei verschiedenen von Ihnen auf den Daten aus **pendigits-training.txt** trainierten Netzen. Sie können für das Training die Parameter k und γ beliebig auswählen und beliebig lange trainieren. Geben Sie jeweils die Summe der Fehler nach dem Training, sowie die Confusion-Matrix und Erkennungsrate für die Klassifizierung aus.