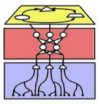

Universität Hamburg
Department Informatik, ABe NATS, WTM & TAMS
Wolfgang Menzel, Stefan Wermter, Jianwei Zhang

Kontakt: Cornelius Weber
Raum F-233
Tel 42883-2537

Übungen zum Modul: Algorithmisches Lernen

SS 2012 Blatt 5

Bearbeitung bis: 9.05.2012

Hinweis: Die Aufgaben im AL Teil II sind für den Python Code des "Machine Learning" Buchs von Stephen Marsland optimiert. Sie können jedoch mit beliebiger Software gelöst werden, etwa:

- PyBrain: http://pybrain.org
- Emergent: http://grey.colorado.edu/emergent/index.php/Main_Page
- Encog: http://www.heatonresearch.com/encog
- LENS: http://tedlab.mit.edu/~dr/Lens

Diese haben wir jedoch nicht alle getestet. Achten Sie bei der Wahl des Tools darauf, dass mindestens folgende Netztypen unterstützt werden: Multi-Layer Perceptron (MLP), Elman Netz (simple recurrent network), Self-Organizing Map (SOM, auch als Kohonen Netz bekannt).

Aufgabe 5.1

Für die beiden größten Gehirnteile, Kortex und Hippocampus, in der Maus gelten folgende Zahlen:

- Volumen von Kortex und Hippocampus: 2×90 Kubikmillimeter
- Zelldichte: 90.000 Zellen pro Kubikmillimeter
- Synapsendichte: 700.000.000 pro Kubikmillimeter
- Axonlänge pro Neuron: 40 Millimeter
- Reichweite der Axone: 1 Millimeter
- Dendritenlänge pro Neuron: 4 Millimeter
- Reichweite der Dendriten: 0.2 Millimeter

Berechnen Sie daraus:

- Zahl der Zellen
- Gesamt-Axonlänge pro Kubikmillimeter
- Abstand der Zellen voneinander
- Synapsen pro Neuron
- Synapsen pro Axonlänge
- Synapsen pro Dendritenlänge
- Schätzen Sie die Wahrscheinlichkeit für Synapsen zwischen benachbarten Zellen

Die Anzahl der sensorischen Eingangsfasern in diese Gehirnteile ist kleiner als 1 Million. Was lässt sich daraus über die Funktionsweise von Kortex und Hippocampus aussagen?

— bitte wenden —

Wolfgang Menzel
(menzel@informatik...)
Stefan Wermter
(wermter@informatik...)
Jianwei Zhang
(zhang@informatik...)

Vorlesung:
Mi 8:15-11:45, F-009
Übungen:
Mi 12:15-13:45, F-009

Dienstgebäude
Universität Hamburg
Vogt-Kölln-Str. 30
22527 Hamburg

Öffentliche Verkehrsmittel
Buslinien 181, 281
Haltestelle: Informatikum

Aufgabe 5.2 Perzeptron und XOR-Klassifizierung

Um die XOR Logik mit einem Perzeptron abzubilden, versehen Sie den XOR Datensatz mit einer dritten Eingabedimension. Deren Wert x_3 berechne sich aus den gegebenen Werten x_1 und x_2 wie folgt:

1. $x_3 = x_1 \cdot x_2$
2. $x_3 = x_1 + x_2$
3. $x_3 = x_1 - x_2$
4. $x_3 = e^{-(x_1 - x_2)^2}$
5. $x_3 = random(0.0, 1.0)$

In welchen Fällen kann das Perzeptron nun XOR richtig klassifizieren? Wenn nein, warum nicht? Warum ist Fall 5. ungünstig?

Gibt es weitere Möglichkeiten, wie das Perzeptron die XOR Daten klassifizieren kann?

Aufgabe 5.3 Perzeptron und AND-Klassifizierung

Trainieren Sie ein Perzeptron, das die logische AND Funktion abbildet.

Berachten Sie den Code des Buchs von Stephen Marsland "Machine Learning - An Algorithmic Perspective" von der Webseite: http://seat.massey.ac.nz/personal/s.r.marsland/MLBook.html

Die Klasse pcn, die das Perzeptron implementiert, ist in der Datei pcn_logic_eg.py (unter "Chapter 2").

Was macht darin die Methode pcnfwd? Charakterisieren Sie die Matrizen inputs, outputs, weights.

Folgender ist der Code (im Buch!), der das Perzeptron aus dieser Klasse mit den AND Daten trainiert:

```
from numpy import *
import pcn_logic_eg
inputs = array([[0,0],[0,1],[1,0],[1,1]])        # the input data
targets = array([[0],[1],[1],[1]])               # the outputs for AND
p = pcn_logic_eg.pcn(inputs,targets)
p.pcntrain(inputs,targets,0.25,6)
```

Lassen Sie das Programm (oder ein anderes Perzeptron Ihrer Wahl) laufen. Hat das Neuron alle AND Daten gelernt?

Wieviele Parameter hat der Programmierer im Programm gesetzt, wieviele Parameter werden gelernt?

Geben Sie die Confusion Matrix aus (entsprechende Methode der pcn Klasse benutzen)!

Lesen Sie die Gewichte und Schwelle und berechnen Sie die Gerade, die die Entscheidungsgrenze bildet!

Kann dieses Problem mit negativen Gewichten, oder negativem Bias, gelöst werden?