



(für SOM Gruppen)

**Übungen zum Modul: Algorithmisches Lernen**  
**SS 2012 Blatt 7 (für SOM Gruppen)**  
**Bearbeitung bis: 23.05.2012**

Gegeben sind auf dem MinCommSy eine Self-Organizing Map (Kohonenkarte) in Python: SOM.py. Ferner einige Datensätze mit zumeist je 10000 Daten. In dieser Übung gilt es, sich mit der SOM anzufreunden und Erfahrungen mit den Daten, sowie der Parametrisierung des Netzes zu machen. Das Programm data\_create.py, mit dem die Daten kreiert wurden, ist ebenfalls erhältlich, so dass Sie leicht auch die Daten in einem anderen Format für eine andere SOM Software generieren können.

Die Visualisierung erfolgt, indem in den (2D oder 3D) Datenraum die Gewichte der Kartenneurone als Punkt dargestellt werden, und die Punkte benachbarter Neurone als Linien verbunden werden. Dazu wird das Programm Gnuplot verwendet. Das Programm SOM.py exportiert nach dem Training zwei Dateien: out.dat enthält die Punkte der Neurone im Datenraum; out.gnu enthält Gnuplot Skriptbefehle, um die Linien zu setzen.

**Aufgabe 7.1 Selbstorganisierende Karte auf einer 2D Mannigfaltigkeit im 3D Raum**

Trainieren Sie das SOM auf den Daten data\_sphere.dat. Geeignet ist etwa eine Karte der Topologie eines 10x10-Gitters, d.h. im Programm ist dim\_a = 10 und dim\_b = 10.

Stellen Sie die Daten als Punkte im 3D Raum und gleichzeitig die Gewichtsvektoren als Kreuze dar. Das erfolgt in der Kommandozeile von Gnuplot durch:

```
plot "data_sphere.dat" with dots, "out.dat" with points
```

Anschließend plotten Sie die Verbindungslinien zwischen den Gewichten benachbarter Neuronen durch:  
load "out.gnu"

Spielen Sie mit den Parametern, wie der Kartendimensionen, der Lernschrittweite sowie dem Reduktionsfaktor, mit dem die Nachbarschaftswechselwirkungsbreite sigma in jedem Trainingsschritt reduziert wird. Charakterisieren Sie die Daten. Gibt es Overfitting?

**Aufgabe 7.2 Faltung der Karte im flachen Raum**

Beobachten Sie, wie die Karte sich im Raum faltet. Trainieren Sie ein Netz mit einer Topologie von 1x50 Neuronen auf den Daten data\_snake.dat, die im 2D liegen. Ebenso, ein Netz mit einer 15x25 Topologie auf den Daten data\_waves.dat im 3D Raum.

Was können Sie über die Daten allein aus dem Verhalten des Netzes sagen?

Hinweis: Für Daten in 2D benutzen Sie in Gnuplot plot anstatt splot. Stellen Sie zur Kontrolle alle Daten dar.

**Aufgabe 7.3 Travelling Salesperson Problem**

In data\_capitals.dat befinden sich die 2D Koordinaten von 44 europäischen Hauptstädten. Das Problem des Handlungsreisenden ist, alle Städte auf möglichst kurzem Weg (Fluglinie) zu besuchen. Ein naheliegender Lösungsweg ist, ein SOM mit 1x44 Neuronen auf diesen Daten zu trainieren. Tun Sie dies! Hinweis: Im Programm erhöhen Sie die Variable batch, um länger zu lernen.

Um die Reise zu einer Rundreise zu machen, modifizieren Sie den Code, so dass das Netz zyklisch ist, d.h. das erste Neuron Nachbar des letzten ist. Für das TSP reicht eine 1D-Topologie der Karte aus (geschlossene Kette), anstatt eine in zwei Dimensionen geschlossene Mannigfaltigkeit (Torus).

Ist das Problem perfekt gelöst? Falls nicht, welche Verbesserungsmöglichkeiten gibt es?



### **Aufgabe 7.4 k-Means**

Verwandeln Sie die SOM in den k-means Algorithmus (als on-line, nicht batch, Version). Eine sinnvolle Anwendung ist ein Netz mit  $k=4$  Neuronen auf den Daten `data_clusters.dat`. Werden die Daten bei mehrmaligem Durchlauf immer richtig geclustert? Falls nicht, was geht schief? Wird das Problem durch Verwendung einer SOM gelöst?

### **Hinweis für Benutzer des Betriebssystems Windows der Firma Microsoft**

Die Homepage von Gnuplot ist [www.gnuplot.info](http://www.gnuplot.info). Dort erhalten Sie über SourceForge die Datei `gp460win32.zip`. Entpacken Sie diese, und im Unterverzeichnis `gnuplot\binary` befindet sich das ausführbare Programm `wgnuplot.exe`. Dieses starten Sie, und Sie können gleich loslegen. In Gnuplot können Sie den Befehl `pwd` benutzen und etwa auf folgende Weise in das Verzeichnis Ihrer Daten wechseln:

```
cd "C:/Users/weber/Desktop/AL/SOM_data/"
```