

12. Übungsblatt

Die folgenden Übungsaufgaben werden im Labor G29-035 durchgeführt. Sie können sich gern darauf vorbereiten. Benötigtes Material wird in der Übung verteilt. Machen Sie sich mit dem *Stuttgart Neural Network Simulator (SNNS)* vertraut. Er ist für verschiedene Betriebssysteme unter <http://www.ra.cs.uni-tuebingen.de/SNNS/> erhältlich. Hinweise zum SNNS finden Sie auch auf der Website dieser Lehrveranstaltung.

Aufgabe 44 SNNS: Biimplikation

- a) Legen Sie eine SNNS-Musterdatei für die Biimplikation an. (Orientieren Sie sich dabei an der Datei `examples/xor.pat`.) Legen Sie mit dem SNNS ein neuronales Netz mit zwei Neuronen in der Eingabeschicht, zwei Neuronen in der versteckten Schicht und einem Neuron in der Ausgabeschicht an. Initialisieren Sie das Netz mit Zufallswerten (Schalter `init` im `snns-control`-Fenster) und testen Sie es im Einzelschritt (mit dem Schalter `test` im `snns-control`-Fenster). Lassen Sie sich die Verbindungsgewichte und den Schwellenwert anzeigen und deuten Sie diese geometrisch. (Beachten Sie, dass die Neuronen statt einer scharfen Schwellenwertfunktion eine logistische Aktivierungsfunktion verwenden. Sie können die Position der Neuronen im `display`-Fenster verändern, wenn Sie aus den Menüs, die sich bei Drücken der linken Maustaste zusammen mit der Control-Taste öffnen, zuerst `Units` und dann `Move` auswählen.)
- b) Wählen Sie als Lernfunktion (Schalter `sel.func.`) `BackpropMomentum`, trainieren Sie das Netz bis sich der Fehler nicht mehr bzw. kaum noch verringert (Fenster `graph`, es kann einige tausend Epochen dauern), lassen Sie sich wieder die Verbindungsgewichte und den Schwellenwert anzeigen und fertigen Sie eine neue graphische Darstellung an. Wiederholen Sie den Lernvorgang mehrmals. Wird immer das gleiche Ergebnis erreicht? Wenn nein, warum nicht?
- c) In der Zeile `learn` des `snns-control`-Fensters können Sie Parameter des Lernverfahrens eingeben. Der erste Parameter ist die Lernrate, der zweite ein Momentterm (Genaueres zu diesem Term in der Vorlesung). Experimentieren Sie mit verschiedenen Werten für diese beiden Parameter! Welche Wirkung lässt sich erzielen?

Aufgabe 45 SNNS: MLP zur Analyse der Irisdaten

Die von R. A. Fisher gesammelten Irisdaten sind wahrscheinlich der bekannteste Datensatz im Bereich der Mustererkennung. Er enthält zu jeder der drei Irisarten Iris Setosa, Iris Versicolor und Iris Virginica Beschreibungen von 50 Beispielen, insgesamt also 150 Beispiele. Für jedes Beispiel sind die Länge und Breite der Kelchblätter (sepal length/width, in cm), die Länge und Breite der Blütenblätter (petal length/width, in cm) und die Irisart angegeben. (Passende Eingabedatei namens `iris.train.pat` `iris.test.pat` erhalten Sie während der Übung.)

- a) Versuchen Sie, ein neuronales Netz als Klassifikator zu trainieren. D.h. das neuronale Netz soll aus den Eingabedaten die Irisart vorhersagen können.

- b) Laden Sie die Trainingsmuster `iris.train` und die Testmuster `iris.test` und trainieren und testen Sie mit diesen das erzeugte neuronale Netz.
- c) Experimentieren Sie mit der Zahl der Einheiten in der versteckten Schicht. Welche Fehler-raten lassen sich mit welcher Zahl versteckter Einheiten erzielen?
- d) Experimentieren Sie mit dem Weglassen von Eingabewerten. Wählen Sie dazu beim Anlegen des Netzwerks *nicht full connection*, sondern legen Sie die Verbindungen per Hand an und lassen Sie dabei einige Eingabeeinheiten einfach unverbunden. (Eine Anleitung finden Sie im SNNS-Handbuch unter “Link Editor”.) Auf welche Eingabewerte kann man am leichtesten verzichten?

Aufgabe 46 SNNS: RBF-Netz zur Analyse Irisdaten

- a) In Aufgabe 45 haben wir mit dem SNNS ein mehrschichtiges Perzeptron zur Klassifikation der Irisdaten trainiert. Trainieren Sie nun, ein Radiale-Basisfunktionen-Netz für die gleiche Lernaufgabe! Verwenden Sie allerdings eine veränderte Musterdatei, die Sie in der Übung erhalten werden.
- b) Lassen sie sich die gelernten radialen Basisfunktionen und die Ausgaben des Netzes im `projection`-Fenster anzeigen. (Sie können auch das gelernte Netz speichern und die radialen Basisfunktionen anhand der Netzparameter skizzieren).

Hinweise: Sie müssen die Aktivierungsfunktionen der Neuronen, die Initialisierungsmethode für die Gewichte und das Lernverfahren ändern. Wählen Sie als Aktivierungsfunktionen für die Neuronen der versteckten Schicht `Act_RBF_Gaussian` und für die Ausgabeneuronen `Act_IdentityPlusBias`. Wählen Sie als Initialisierungsmethode `RBF_Weights` und als Lernverfahren `RadialBasisLearning`. Beachten Sie, dass Sie einige Parameter setzen müssen, z.B. den Biaswert $b = \frac{1}{2\sigma^2}$ der Basisfunktionen im vierten Feld der Zeile `INIT` des `snns-control`-Fensters, sowie die Lernparameter in der Zeile `LEARN` dieses Fensters. Weitere Informationen finden Sie im HTML-Handbuch zum SNNS.

Aufgabe 47 SNNS: SOM zur Analyse der Irisdaten

In dieser Aufgabe wenden wir uns noch einmal den Irisdaten zu, für die wir diesmal mit dem SNNS eine selbstorganisierende Karte trainieren wollen. (Eine passende Eingabedatei erhalten Sie während der Übung.)

- a) Legen Sie eine selbstorganisierende Karte mit ca. 10×10 Neuronen in der Ausgabeschicht an (`BIGNET > Kohonen`)! Trainieren Sie diese Karte mit den Irisdaten für einige hundert Epochen (`cycles`)! Wählen Sie dabei `shuffle` im `snns-control`-Fenster, um die Trainingsmuster zufällig zu mischen. Beachten Sie, dass Sie einige Lernparameter setzen müssen (siehe Abschnitt “Parameters of the Learning Functions” im SNNS-Handbuch). Setzen Sie außerdem als Aktivierungsfunktion der Ausgabeneuronen `Act_Euclid`.
- b) Testen Sie die trainierte Karte durch Drücken des Knopfes `test` im `snns-control`-Fenster für die einzelnen Trainingsmuster. (Dazu ist es günstig, die Gitterweite der Anzeige im `snns-display`-Fenster auf 16 Pixel zu setzen und die Beschriftungen abzuschalten.) Wie werden die Eingabemuster den verschiedenen Ausgabeneuronen zugeordnet? (Die ersten

50 Trainingsmuster gehören zur Klasse *Iris Setosa*, die nächsten 50 zur Klasse *Iris Versicolor* und die restlichen 50 zur Klasse *Iris Virginica*.)

- c) Initialisieren und trainieren Sie die Karte erneut! Erhalten Sie wieder die gleichen Zuordnungen?