Kurzeinführung in SNNS:

Der Stuttgarter Neuronale Netze Simulator ist relativ umfangreich, inzwischen leicht veraltet, aber sehr verbreitet, insbesondere wird er bei Suse mitgeliefert. Aktuell ist die Version 4.2, Version 4.1 unterscheidet sich kaum davon. Der SNNS sollte auf den Linux-Rechnern im Lehre-Netz installiert sein. Ein Online Manual und weitere Infos gibt's auf der Vorlesungshomepage. Ebenso liegen dort gezippte Versionen des Simulators und des Manuals, sowie ein link zur original-SNNS-Seite mit z.B. einer Windows-Version. Für einige Zusatzprogramme (analyze, mkhead, ...) ist es für Sie im Lehre-Netz nötig, folgenden Pfad zu setzen: /usr/lib/SNNS/tools/bin/i386-suse-linux

Start Die graphische Benutzeroberfläche wird mit snns gestartet. Es erscheint der SNNS-Manager.

manager Das Infofenster beseitigt man durch Draufklicken. Vom Manager aus wird das Programm gesteuert. Im Fenster selbst befindet sich außer den Icons für weitere Punkte gelegentlich ein Prompt ⟩ für's DISPLAY. SNNS verlassen mit QUIT.

FILE Hier können Netz-, Pattern- und Ergebnisdateien geladen oder gespeichert werden. Resultfiles werden bei drücken von SAVE nach Wunsch neu erzeugt oder verlängert.

HELP Online-Hilfe, mit TOPICS, Markieren des gewünschten Punktes und LOOK bekommt man direkt, was man will.

PRINT Erzeugt ps-Datei der aktuellen Ansicht im DISPLAY. PS-Drucker z.B. im Rechenzentrum, dazu evtl. telnet zu dosuni.

INFO Es werden die Werte einer aktuellen Verbindung, die im Display markiert wurde, angezeigt.

BIGNET Man kann große Netze erzeugen. Z.B. Feedforward bzw. General: x,y,z-Anzahl Eingabeneuronen eingeben (i.a.x=z=1), ENTER (hinten einfügen, INSERT wäre vorne), TYPE für hidden, dasselbe für jede hidden-Schicht, TYPE für output, dasselbe für die output-Schicht. FULL CONNECTION und CREATE erzeugt das Netz mit Default Gewichten (0) und Default Aktivierungsfunktion (sgd). SHORTCUT CONNECTION schafft zusätzlich direkte Verbindungen.

DISPLAY Zeigt das aktuelle Netz. Man kann unter setup einstellen, was wo angezeigt wird. Man kann hier auch Netztopologien modifizieren. Dazu die Maus im Display-Fenster lassen. Die Kommandos werden im Manager angezeigt. Etwa (... das brauchen Sie nicht wirklich wissen ;-)
u s f a (für unit set function activation – Aktivierungsfunktion setzen),
u i d (Einheit einfügen),
u d/m (Löschen/Verschieben markierter Units),
u c a/i/o/n (kopieren mit allen/input/output/keinen Verbindungen),
u s b (alle Biases werden auf den im INFO gesetzt),
l c i/o (alle Input-/Outputverbindungen zu einer Gruppe werden zum Neuron unter der Maus kopiert),
l d t/s (alle Verbindungen einer Gruppe vom/zum Neuron unter der Maus werden gelöscht),
l s (alle Links werden auf Wert im INFO gesetzt).
Man markiert Einheiten mit der linken Maustaste bzw. durch Aufziehen eines Fensters mit der linken Maustaste. Man markiert Links durch die daranliegenden Units. Einen einzelnen Link (für INFO) bekommt man durch Markieren der angrenzenden Units mit der mittleren Maustaste. Demarkieren mit shift und der linken Maustaste oder der rechten Maustaste. Man kann einzelne Werte auch direkt im INFO setzen, die zugehörigen Einheiten werden im DISPLAY markiert. Das ist etwas aufwendig, was aber nicht so schlimm ist, da man in der Regel nicht im Netz direkt herumeditiert, sondern lernt. Alternativ kann man auch immer die ASCII-Netzdatei direkt editieren.

GRAPH gibt während des Trainings den quadratischen Fehler (SSE), mittleren Fehler (MSE) oder Fehler durch die Anzahl Units geteilt (SSE/#u) aus, Skalieren der Achsen erfolgt mit den Pfeilen. Es werden bis zu 25 Lernkurven in ein Fenster gezeichnet. Bei Version 4.1 gibt's manchmal unmotivierte Zacken, diese sollte man ignorieren.

WEIGHTS zeigt die Änderung der Gewichte graphisch (Hintondiagramm) an.

CONTROL Hauptsächlich zum Lernen. Man kann aber auch unter GOTO bzw. den Pfeilen die (gewünschten) Ausgabepattern im DISPLAY anzeigen lassen und mit TEST die tatsächliche Ausgabe berechnen. ERROR zeigt u.a. den quadratischen Fehler SSE an.
Initialisieren: INIT initialisiert das Netz, voreingestellt: RANDOMIZE_WEIGHTS, kann man auch unter dem dritten Knopf SEL_FUNC wählen. Es werden die Gewichte mit kleinen Zufallszahlen zwischen den links unten in zwei Fenstern eingetragenen Zahlen initialisiert.
Trainieren: Auswählen der zu trainierenden Patterndatei beim oberen USE. Die Pattern müssen eingelesen worden sein! Auswählen der Funktion (voreingestellt ist die Lernfunktion STD_BACKPROPAGATION) beim obersten SEL_FUNC. Die zur Funktion gehörenden Parameter werden in Fenstern neben LEARN gesetzt.
Unter dem zweiten SEL_FUNC wählt man die Update-Reihenfolge. Voreingestellt und bei Feedforward Netzen immer richtig ist TOPOLOGICAL_ORDER. STEPS kann dann auf 1 stehen.
Unter CYCLES wählt man die Zahl der Zyklen d.h. Durchläufe durch alle Pattern. Mit ALL trainiert man alle Pattern. Sofern SHUFFLE gedrückt ist, werden diese dem Netz in einer zufälligen Reihenfolge präsentiert. SINGLE lernt nur ein Pattern. STOP hält an. Es wird dabei der SSE (quadratischer Fehler), der MSE (SSE/Anzahl Pattern) und der SSE/Anzahl Ausgabeeinheiten ausgegeben.
Mit DEL SET kann man geladene Patterndateien wieder löschen. Es sind maximal 5 möglich.
Lernfunktionen und Parameter: (was das alles heißt kommt später in der Vorlesung ...)
STD_BACKPROPAGATION mit Lernrate $(0.1 - 1.0)$, d_{max} $(0.1 - 0.2)$, d.h. Differenzen unterhalb d_{max} in einer Ausgabeeinheit werden nicht berücksichtigt.
BACKPROPBATCH (Änderungen werden für alle Pattern aufaddiert und dann erst vorgenommen) mit gleichen Parametern.
BACKPROPMOMENTUM: Lernrate $(0.2 - 1)$, Momentum Term $(0 - 1)$, flat spot Elimination $(0 - 0.25$, wird zu sgd$'$ addiert), d_{max} $(0 - 0.3)$
BACKPROPWEIGHTDECAY: Lernrate, Decay $(0.005 - 0.3)$ (Gewichtsbetrag wird in jedem Schritt entsprechend verkleinert), d_{min} (unterhalb werden Gewichte 0 gesetzt), d_{max}
QUICKPROP: Lernrate $(0.1 - 0.3)$, maximales Wachstum der Schrittweite durch den Differenzenquotienten relativ zu 1 $(1.75 - 2.25)$, weight-decay (0.0001 oder kleiner), d_{max}
RPROP: Startwert für Änderungen (0.1), Maximale Änderung (50), Decay (4) (als $1 : 10^x$ notiert)
Skalierte konjugierte Gradienten SCG: Line search mit Verfahren zweiter Ordung, Parameter ($\sigma_1 \in]0, 10^{-4}]$, $\lambda_1 \in]0, 10^{-4}]$ beides zur Berechnung der zweiten Ableitungen, $\triangle_{max}$ =max. Schrittweite, ϵ_1 = Maschinengenauigkeit).

pattern Pattern files haben die Extension .pat. Sie bestehen aus einem Kopf, der Daten zu den Pattern enthält, und den sukzessive als $x_1\ y_1\ x_2\ y_2\ \dots$ geschriebenen Pattern, zwischen den Komponenten und den einzelnen Pattern steht je ein Trenner, was ein Leerzeichen oder eine neue Zeile sein kann. Ein Beispiel ist in bsp.pat. Den Kopf kann man mit dem Aufruf mkhead <Anzahl Pattern> <Eingabedimension> <Ausgabedimension> erzeugen. (Mit >> <name> in die Datei name umlenken.) Achtung 4.2 Benutzer: Evtl. im Kopf Vx.y durch V4.1 bzw. V4.2 ersetzen!