

Neuronale Netze (SS 2002)
7. Übungsblatt

Abgabe: Donnerstag, 6.6.02, 12⁰⁰ Uhr, Briefkasten ‚Neuronale Netze‘ im 4. Stock des AVZ

1. (5 Punkte) Es seien f_1, \dots, f_k Funktionen von \mathbb{R}^n nach \mathbb{R} ; z.B. Netze, die auf die Funktion f trainiert wurden. Es sei $(x, f(x))$ ein Muster in $\mathbb{R}^n \times \mathbb{R}$ für die Funktion f . α_i seien nichtnegative Werte mit $\sum \alpha_i = 1$; \bar{f} sei die Mittelung der f_i , also $\bar{f}(x) = \sum_{i=1}^k \alpha_i f_i(x)$.

Diese Mittelung verbessert evtl. den Fehler! Rechnen Sie die Formel

$$(\bar{f}(x) - f(x))^2 = \sum_{i=1}^k \alpha_i (f_i(x) - f(x))^2 - \sum_{i=1}^k \alpha_i (f_i(x) - \bar{f}(x))^2$$

nach (direkt oder mithilfe Bias/Varianz). Was bedeutet diese Formel?

2. (5 Punkte) Trainieren Sie ein einfaches sigmoides Neuron auf die Daten aus linie.pat (Datei im Web). Was ergibt sich als Lösung? Trainieren Sie ein einfaches Perzeptron mit dem Perzeptronalgorithmus auf linie.pat. Was ergibt sich dann? Wie ist das zu erklären? Gibt es insbesondere Vorteile/Nachteile der beiden ‚Lösungen‘?
3. (5 Punkte) Trainieren Sie verschieden große Netze auf die Daten glass.pat und geben Sie je den Generalisierungsfehler auf glass_test.pat mit aus (die beiden Mengen entstammen derselben Verteilung: einer Klassifikation von Glassplittern). Wie zeigt sich hier das Bias-Varianz-Dilemma? Sie können sich dabei den Fehlerverlauf der Menge glass_test.pat während des Trainings anzeigen lassen (VALID).
4. (5 Punkte) Versuchen Sie, die Zeitreihe aus punkte.pat mit einem Netz vorherzusagen. Was erhalten Sie für Prognoseergebnisse für die Daten test1.pat bzw. test2.pat? Wie gut sind diese Ergebnisse? Kann man sie verbessern?

[VALID] Neben dem Knopf VALID kann man unter USE eine sog. Validierungsmenge bezeichnen, d.h. eine Menge, auf der schon während des Trainings der Fehler berechnet, die aber selbst nicht zum Training benutzt wird. Dazu muß im Fenster neben VALID eingetragen werden, nach wievielen Schritten der Fehler auf der Validierungsmenge jeweils berechnet werden soll (0 entspricht nie). Im Graphen werden dann zwei Kurven, der Trainings- und der Validierungsfehler (= Generalisierungsgüte) angezeigt (unter SNNS 4.1. ein bug: völlig überflüssige Zacken :-)

[Variable Pattern] Etwa eine Zeitreihe liegt als Reihe vor, die Trainingspattern für ein Neuronales Netz mit k Folgengliedern als Eingabe ergeben sich dann jeweils aus unmittelbar aufeinanderfolgenden k Werten als Eingabe und dem $k + 1$ -ten als Ausgabe. Da das so häufig kommt, gibt es eine Möglichkeit, Zeitfenster über eine Zeitreihe laufen zu lassen und die zugehörigen Pattern entsprechend zu generieren.

In der Patterndatei entspricht Eingabe- und Ausgabeunitanzahl der fixen Anzahl, die dann etwa von verschiedenen Zeitreihen bedient werden; die Anzahl der variablen Dimensionen in der Ein- und Ausgabe wird angegeben, für eine Zeitreihe ist das 1, wenn man Teile eines Bildes betrachtet, kann das 2 sein; es wird die maximale Größe je Dimension aus Speicherplatzzwecken angegeben; vor jedem Pattern steht die aktuelle Größe, d.h. das Pattern hat an tatsächlichen Werten Eingabeunitanzahl multipliziert mit aktueller Größe viele. (Siehe z.B. bsp.pat.)

Wenn man mit so einer Patterndatei trainieren will, lädt man sie normal ein, definiert auch das Netz normal, muß aber, bevor man irgendetwas anzeigt oder trainiert, unter dem Knopf DEF

SUB im control-panel die tatsächliche Größe der Pattern fixieren. Je Dimension kann angegeben werden, wieviele (variable) Eingaben gemacht werden (Size). Die Anzahl der Eingabeneuronen sollte diese Zahl multipliziert mit der Anzahl Eingabeneuronen im Pattern-file sein. Es kann weiter angegeben werden, wie groß die Schritte (Step) sind. Sind diese gleich der Size, dann überlappen die Pattern nicht, sind sie kleiner, überlappen die Pattern.

Beispielsweise werden bei der Reihe x_1, x_2, x_3, x_4 als Eingabepattern und x_3, x_4, x_5 als Ausgabepattern durch Input: Size= 2, Step= 1, Output: Size= 1, Step= 1 nacheinander die Pattern $(x_1, x_2 \rightarrow x_3); (x_2, x_3 \rightarrow x_4); (x_3, x_4 \rightarrow x_5)$ zum Training erzeugt.

Es wird durch das Inputpattern bestimmt, wieviele Pattern extrahiert werden, ist die Ausgabe zu lang, wird der Rest ignoriert, ist sie zu kurz, wird beginnend vom Anfang wieder gelesen. Sie können daher, falls eine Reihe prognostiziert werden soll, i.A. nicht den Startpunkt einer Reihe für Eingabe und Ausgabe gleich wählen.

Im SUB PAT Panel ist ein Knopf für das Shufflen der Teilpattern, sowie Knöpfe, um sich die Pattern im Display anzeigen zu lassen.

[analyze] ist ein Kommandozeilentool, das das Auswerten von Klassifikationen erleichtert. Dazu kann man bei einem fertig trainierten Netz die Ergebnisse als result-file abspeichern. Man sollte für analyze sowohl die gewünschten Ausgaben (teaching-output) als auch die tatsächlichen Ausgaben bereithaben. analyze zählt dann, wieviele Muster korrekt/falsch/gar nicht klassifiziert sind. Parameter:

Eingabedatei in -i <result-Datei>, Ausgabe der Nummer falsch/richtig/gar nicht/alles drei klassifizierter Muster (-w/-r/-u/-a), bei der Option -s wird das in eine Statistik zusammengefaßt, in -o <Ausgabe> oder auf die Kommandozeile. Es gibt verschiedene Klassifizierungsfunktionen, die mit -e eingegeben werden, die jeweiligen Parameter mit -l und -h. Funktionen sind 402040, WTA und band.

402040: (unäre Ausgaben) Genau eine Ausgabe ist $\geq h$, diese soll auf 1 klassifiziert werden, alle anderen Ausgaben sind $\leq l$, dann ist das richtig. Genau eine Ausgabe ist $\geq h$, diese soll nicht auf 1 klassifiziert werden, alle anderen sind $\leq l$, dann ist das falsch. Ansonsten: unbekannt.

WTA: (unäre Ausgaben) Die maximale Ausgabe soll auch die maximale Ausgabe des Trainingspattern sein. diese ist mindestens h , alle anderen sind mindestens um l kleiner, dann ist das richtig. Dieselbe Situation, aber die maximale Ausgabe ist nicht die maximale Soll-Ausgabe. Dann ist das falsch. Sonst: Nicht klassifiziert.

band: Genau, wenn alle Ausgaben in einem Bereich von soll $-l$ und soll $+h$ liegen, wird das als richtig angenommen, falls alle außerhalb liegen, ist falsch klassifiziert, ansonsten unbekannt.

Beispiel: Testet Abweichung um 0.2: analyze -i name.res -o name.out -a -s -e band -l 0.2 -h 0.2

Dieses tool liegt im selben Verzeichnis wie mkhead! (Pfad setzen.)