Barbara Hammer, FB Mathematik/Informatik, Universität Osnabrück

**Neuronale Netze (SS 2002)**
**11.Übungsblatt**

Es gibt keine Punkte mehr für diesen Zettel; Musterlösungen finden Sie im Semesterapparat; bei Fragen kommen Sie bitte einfach vorbei oder fragen in der Übung nach. Dieses Blatt ist nicht klausurrelevant. Die Fragen sind teilweise nicht so einfach.

---

Was ist richtig:

ja naja nein

1. □ □ □ Für binäre Ausgaben $y \in \{-1, 1\}$ (d.h. $-1$ statt $0$) könnte die Änderung im Perzeptronalgorithmus als $\boldsymbol{w} := \boldsymbol{w} - y\boldsymbol{x}$, falls $\boldsymbol{x}$ falsch ist, geschrieben werden.

2. □ □ □ Die Pattern $(0, 0, 0; 1)$, $(1, 0, 0; 0)$, $(0, 1, 0; 0)$, $(0, 0, 1; 0)$, $(1, 0, 1; 1)$, $(0, 1, 1; 1)$, $(1, 1, 1; 0)$ können mit einem $(3, 2, 1)$ Netz mit der Aktivierungsfunktion $H$ korrekt abgebildet werden.

3. □ □ □ Wenn ein Perzeptronnetz eine endliche Trainingsmenge korrekt abbildet, dann kann dasselbe auch mit ganzzahligen Gewichten und Biases geschehen.

4. □ □ □ Zu entscheiden, ob eine Menge nicht linear trennbar ist, ist NP-vollständig.

5. □ □ □ Die Funktion, die genau die Muster, die mindestens ein Pixel am Rand auf $1$ haben, nach $1$ abbildet, ist mit einem Rosenblatt Perzeptron mit Durchmesser beschränkten Neuronen erkennbar.

6. □ □ □ Egal in welcher Reihenfolge die Muster präsentiert werden, der Tower Algorithmus konvergiert nach endlich vielen Schritten, sofern man je Nachtraining mit einem Gewichtsvektor stoppt, der eine minimale Anzahl Fehler macht.

7. □ □ □ Der upstart-Algorithmus generalisiert immer sehr gut.

8. □ □ □ Mit einem Ensemble der Form $H(\sum_{i=1}^{m} f_i(x) - m/2)$, $m$ beliebig, $f_i$ Perzeptronen, kann man jede Boolesche Funktion darstellen.

9. □ □ □ Für ein feedforward Netz mit der Aktivierungsfunktion $\tanh$ würden sich die Fehlersignale $\delta_j = \partial E / \partial \text{net}_j$ berechnen als $(y_j - o_j)(1 + o_j^2)$, $j$ ist Ausgabe, $\sum_{k \to j} w_{kj}\delta_k(1 + o_j^2)$, sonst.

10. □ □ □ Für ein feedforward Netz mit der Fehlerfunktion $\sum_i (o_i - y_i)^6$ würden sich die Fehlersignale $\delta_j = \partial E / \partial \text{net}_j$ berechnen als $6(o_j - y_j)^5 o_j(1 - o_j)$, $j$ ist Ausgabe, $\sum_{j \to k} w_{jk}\delta_k o_j(1 - o_j)$, sonst.

11. □ □ □ RProp wird üblicherweise als online-Verfahren, d.h. Änderung nach jedem Pattern, trainiert.

12. □ □ □ Bei Gradientenabstieg mit festem Momentum auf einer linearen Fehlerfläche kann die Schrittweite gegenüber der normalen Schrittweite bei einfachem Gradientenabstrieg um einen beliebig großen Faktor wachsen.

13. □ □ □ Man sollte beim Training mit Backprop immer versuchen, den Trainingsfehler so klein wie möglich zu bekommen, um eine gute Generalisierung zu gewährleisten.

14. □ □ □ Statt Pruning könnte man auch gleich mit kleinen Netzen starten, das macht keinen Unterschied im Trainingsergebnis.

15. □ □ □ Die SVM generalisiert ganz schlecht, da sie einem Perzeptron in einem hochdimensionalen Raum entspricht.

16. □ □ □ Wenn man Funktionen $f_i$ zu $f = \frac{1}{m}\sum_{i=1}^{m} f_i$ mittelt, dann wird der lineare Fehler gegenüber $g$, d.h. $\frac{1}{p}\sum_{j=1}^{p} |f(x_j) - g(x_j)|$, gegenüber dem Mittel der Einzelfehler $\frac{1}{m}\sum_i \frac{1}{p}\sum_{j=1}^{p} |f_i(x_j) - g(x_j)|$ höchstens kleiner.

17. □ □ □ Verteilungsunabhängige PAC Lernbarkeit ist äquivalent dazu, daß man für jede Verteilung $P$ und jedes $\epsilon > 0$ eine von diesen beiden Größen abhängige Schranke für die sich ergebende Überdeckungszahl finden kann.

18. □ □ □ Aus verteilungsunabhängig PAC-lernbar folgt die verteilungsunabhängige UCED Eigenschaft und aus der Eigenschaft PAC-lernbar folgt die UCED Eigenschaft.

19. □ □ □ Die VC-Dimension folgender Funktionenklasse ist 42: $\mathcal{F} = \{f : \{1, 2, 3, \ldots, 42\} \rightarrow \{0, 1\} \mid f(1) = 1\}$.

20. □ □ □ Die lineare SVM könnte für die Punkte (Ausgaben $-1/1$-wertig)
$(1, -1; 1)$, $(1, 1; 1)$, $(2, -1; 1)$ $(-1, -1; -1)$, $(-1, 1; -1)$, $(-2, -1; -1)$ die Gerade durch $0$ mit den Gewichten $(1, -0.5)$ ausgeben.

21. □ □ □ Die lineare SVM könnte für die Punkte (Ausgaben $-1/1$-wertig)
$(1, -1; 1)$, $(1, 1; 1)$, $(2, -1; 1)$ $(-1, -1; -1)$, $(-1, 1; -1)$, $(-2, -1; -1)$ die Werte $\alpha = (0, 0, 1, 0, 0, 1)$ berechnen.

22. □ □ □ Feedforward Netze mit zwei verborgenen Schichten sind approximationsuniversell.

23. □ □ □ Bei der Zeitreihenprognose besteht die einzige Vorverarbeitung in der Auswahl des Zeitfensters.

24. □ □ □ Symbolische Daten muß man immer unär kodieren.

25. □ □ □ Neuronen sind gelb.

Und alles Gute :-)