Hans U. Simon
Malte Darnstädt

Bochum, den 07.04.2013
Abgabe bis zum 15.05.2013

Übungen zur Vorlesung
Theorie des Maschinellen Lernens
Sommer 2013
Übungsblatt 04

Aufgabe 4.1
Sei $p_n : \{0,1\}^n \to \{0,1\}$ die n-Parity-Funktion, d.h.

$$p_n(x) = \left(\sum_{i=1}^{n} x_i\right) \mod 2$$

Zeige, dass p_n nicht durch eine lineare Thresholdfunction (ein Perceptron) darstellbar ist, aber dass ein Netzwerk von $O(n)$ vielen linearen Thresholdfunctions dazu ausreicht.

Hinweis: Der Einfachheit halber darf man annehmen, dass n eine Zweierpotenz ist.

Aufgabe 4.2
Sei H eine Klasse von Funktionen von $\{0,1\}^r$ nach $\{0,1\}$ mit VC-Dimension d. Analog zu den Netzwerken von Thresholdfunctions betrachten wir Netzwerke von Funktionen aus H (beachte, dass Kantengewichte in diesen Netzwerken nicht benötigt werden; statt dessen ist die Reihenfolge der Eingänge in einen Knoten relevant).

Zeige, dass die VC-Dimension solcher Netzwerke mit n Eingangs- und k Berechnungsknoten $O(dk\log(k))$ ist. Zeige dafür zunächst $\Pi_{Netz}(m) \leq \Pi_H(m)^k$.

Aufgabe 4.3
Reduziere Decision-Trees von Rang k auf k-Decision-Lists.

Aufgabe 4.4
Reduziere 1-Decision-Lists auf LTF.

In den Aufgaben 3 und 4 zeigen wir, wieviel “Power” in der Klasse der linearen Thresholdfunctions steckt. Denn wie wir aus der Übung wissen, sind Reduktionen transitiv und k-Decision-Lists reduzierbar auf 1-Decision-Lists, so dass insgesamt folgt: “Decision-Trees von konstantem Rang sind reduzierbar auf LTF”

Beachte zu den Aufgaben 3 und 4 die Definitionen auf der Rückseite (auch bekannt aus der Übung).

Definition k-Decision-List
Eine k-Decision-List über $X_n = \{0,1\}^n$ ist eine Funktionenklasse, in der jede Funktion durch eine Liste $((c_1, b_2), \dots, (c_l, b_l))$ und einem Bit b gegeben ist, wobei die c_i Konjunktionen von maximal k Literalen aus $x_1, \overline{x}_1, \dots, x_n, \overline{x}_n$ sind und die b_i ebenfalls Bits sind.
Bei der Berechnung von $f(x)$ wird die Liste von links nach rechts abgearbeitet und sobald eine Konjunktion c_i von x erfüllt wird, ist die Ausgabe von $f(x)$ das Bit b_i. Sollte keine Konjunktion erfüllt werden, gibt $f(x)$ das Bit b aus.

Beispiel: Sei $n = 3$, $k = 2$ und f gegeben durch die Liste $((x_1, 0), (\overline{x}_2 \wedge x_3, 1), (\overline{x}_2, 0))$ und das Bit $b = 1$. Dann ist $f(001) = 1$, $f(101) = 0$ und $f(011) = b = 1$.

Definition Decision-Tree
Ein Decision-Tree über $X_n = \{0,1\}^n$ ist eine Funktionenklasse, in der jede Funktion durch einen binären Baum (mit einer ausgezeichneten Wurzel) gegeben ist, wobei jeder innere Knoten mit einem Literal aus $x_1, \overline{x}_1, \dots, x_n, \overline{x}_n$ und jedes Blatt mit einem Bit beschriftet ist.
Bei der Berechnung von $f(x)$ starten wir an der Wurzel des Baumes und gehen nach rechts, falls das Literal an der Wurzel von x erfüllt wird, ansonsten gehen wir nach links. In dieser Art durchlaufen wir den Baum bis wir bei einem Blatt angekommen sind. Das zu dem Blatt gehörende Bit ist die Ausgabe von $f(x)$.

Definition Rang eines Baumes
Sei T ein binärer Baum und T_L, T_R der linke bzw rechte Teilbaum von der Wurzel aus. Dann ist der Rang von T folgendermaßen definiert:

$$\operatorname{rang}(T) := \begin{cases} 0 & \text{falls } T \text{ nur einen Knoten enthält} \\ \operatorname{rang}(T_L) + 1 & \text{falls } \operatorname{rang}(T_L) = \operatorname{rang}(T_R) \\ \max\{\operatorname{rang}(T_L), \operatorname{rang}(T_R)\} & \text{sonst} \end{cases}$$