Jun.-Prof. Dr. Christian Mendl Abgabe: 5. Dezember 2018

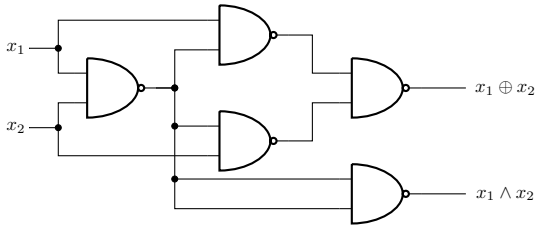
Aufgabe 7 (Perzeptron und NAND-Gate)
Das Perzeptron ist ein von Frank Rosenblatt 1958 vorgeschlagenes vereinfachtes Modell eines künstlichen neuronalen Netzwerks (ANN) und gilt als Vorläufer "moderner" ANNs. Für einen Eingabevektor $x \in \{0,1\}^d$ lautet die Ausgabe eines einzelnen Neurons mit Gewichten $w \in \mathbb{R}^d$ und Bias-Wert $b \in \mathbb{R}$

$$\text{output} = \begin{cases} 0, & w \cdot x + b \leq 0 \\ 1, & w \cdot x + b > 0 \end{cases}$$

wobei $w \cdot x$ das Skalarprodukt zwischen w und x bezeichnet.

(a) Rechnen Sie nach, dass für $d = 2$ und $w_1 = w_2 = -2$, $b = 3$ das logische NAND-Gate realisiert wird. [2 Punkte]

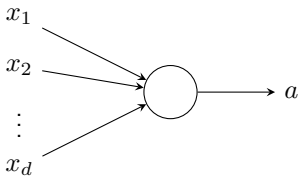
(b) Verifizieren Sie (z.B. durch Einsetzen aller 4 Kombinationen von $x_1, x_2 \in \{0,1\}$), dass nebenstehende Schaltung aus NAND-Gates tatsächlich $x_1 \oplus x_2$ und $x_1 \wedge x_2$ ausgibt, d.h. einen binären "Addierer" implementiert (mit $x_1 \wedge x_2$ der Übertrag der Addition). [2 Punkte]



(c) Wie können w und b für allgemeines d gewählt werden, so dass die Perzeptron-Zelle die verallgemeinerte NAND-Operation $\neg(x_1 \wedge x_2 \wedge \cdots \wedge x_d)$ berechnet? [2 Punkte]

Aufgabe 8 (Trainieren eines künstlichen Neurons)
Wir betrachten ein einzelnes künstliches Neuron beschrieben durch

$$a = \sigma(z), \quad z = w \cdot x + b$$



wobei $x \in \mathbb{R}^d$ die Eingabe bezeichnet, $w \in \mathbb{R}^d$ die Gewichte und $b \in \mathbb{R}$ den Bias-Wert. Die (nichtlineare, monoton steigende) differenzierbare Abbildung $\sigma : \mathbb{R} \to \mathbb{R}$ ist die sogenannte Aktivierungsfunktion.
Das Neuron soll anhand einer vorgegebenen Sequenz von Tupeln $(x^{(j)}, y^{(j)})$, $j = 1, 2, \ldots$ "trainiert" werden (d.h. die Parameter w und b optimiert werden), wobei $x^{(j)} \in \mathbb{R}^d$ die jeweilige Eingabe und $y^{(j)} \in \mathbb{R}$ die gewünschte Ausgabe ist. Für ein Tupel (x, y) quantifizieren wir die Abweichung der tatsächlichen Ausgabe von der gewünschten mittels der quadratischen Kostenfunktion

$$C = \frac{1}{2}(a - y)^2.$$

Die Kostenfunktion soll also minimiert werden.

(a) Bestimmen Sie (symbolisch) den Gradienten von C bezüglich der Parameter w und b, also $\partial C/\partial w_i$, $i = 1, \ldots, d$ und $\partial C/\partial b$. [2 Punkte]

(b) Eine gebräuchliche Aktivierungsfunktion ist die Sigmoid-Funktion $\sigma(z) = 1/(1 + \mathrm{e}^{-z})$. Verifizieren Sie folgende Relation für deren Ableitung:

$$\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big).$$

[1 Punkt]

(c) Das Neuron besitze (anfänglich) die Parameter $w = (1, -1)$, $b = \frac{1}{2}$ und verwende die Sigmoid-Funktion. Bestimmen Sie für jedes Tupel des Trainingsdatensatzes

$x^{(j)}$	$(1,0)$	$(0,1)$	$(\frac{1}{4}, -1)$	$(\frac{1}{2}, \frac{3}{4})$
$y^{(j)}$	$\frac{3}{8}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{16}{25}$

jeweils die numerischen Werte der Gradienten $\partial C/\partial w_i$ und $\partial C/\partial b$, und bilden Sie den Mittelwert der Gradienten bezüglich der vier Tupel. (Sie können für diese Teilaufgabe ein kurzes Python-Programm schreiben, um langwierige manuelle Rechnungen zu umgehen.) Entspricht der mittlere Gradient den Erwartungen, wenn Sie bereits wissen, dass die optimierten Parameter näherungsweise $w = (-\frac{1}{2}, 1)$ und $b = 0$ lauten? [3 Punkte]

Tutoraufgabe 4 (Universalität künstlicher neuronaler Feedforward-Netze)
Künstliche neuronale Feedforward-Netze sind *universell*, d.h. zu jeder kompakten Teilmenge $K \subset \mathbb{R}^n$ und stetigen Funktion $f : K \to \mathbb{R}$ sowie Fehlerschranke $\epsilon > 0$ existiert ein Feedforward-Netz $\mathcal{N}$, so dass $|\mathcal{N}(x) - f(x)| < \epsilon$ für alle $x \in K$ gilt, wobei $\mathcal{N}(x)$ die Ausgabe des Netzes bezeichnet (und die Sigmoid-Aktivierungsfunktion in der Ausgabeschicht weggelassen wird, um z.B. auch negative Ausgabewerte zu ermöglichen).[1] Ziel dieser Aufgabe ist die Skizze eines konstruktiven Beweises der Universalität.

(a) Es sei zunächst $n = 1$. Zeigen Sie durch geeignete Wahl von $w, b \in \mathbb{R}$, dass eine einzelne Nervenzelle mit Ausgabe $\sigma(wx + b)$ eine Stufenfunktion $x \mapsto \Theta(x - s)$ nachbilden kann. Wie erhält man durch Linearkombination von Stufenfunktionen eine Rechteckfunktion bzw. beliebige Treppenfunktion? Konstruieren Sie hieraus ein Netzwerk zur Approximation einer stetigen Funktion $f : K \to \mathbb{R}$, wobei o.B.d.A. $K = [0, 1]$ angenommen werden kann.

(b) Zur Verallgemeinerung auf beliebiges $n \geq 1$ konstruieren Sie mittels (a) zunächst eine Summe von Rechteckfunktionen in jeder Koordinatenrichtung, und unter Verwendung eines Schwellenwerts eine charakteristische Funktion auf einem beliebigen Hyperrechteck. Approximieren Sie nun eine Funktion $f : K \to \mathbb{R}$ durch eine Linearkombination charakteristischer Funktionen.

[1] siehe z.B. G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems* 2, 303–314 (1989) [http://www.dartmouth.edu/~gvc/Cybenko_MCSS.pdf] und K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.* 2, 359–366 (1989).