

Aufgaben zum Tut am 21.05.2007

Thomas Pajor

19. Mai 2007

Aufgabe 1.

Betrachten Sie das Perzeptron-Modell für künstliche neuronale Netze.

- (a) Modellieren Sie folgende Funktionen durch ein Perzeptron mit minimaler Anzahl an Neuronen: AND, OR, XOR.
- (b) Lässt sich die XOR Funktion mit einem Neuron realisieren? Beweisen Sie ihre Behauptung.

Lösung.

- (a) Wir machen uns zunächst die Wahrheitstabeln für die Funktionen AND, OR und XOR klar:

x_1	x_2	$x_1 \wedge x_2$	x_1	x_2	$x_1 \vee x_2$
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

x_1	x_2	$x_1 \leftrightarrow x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Wir können damit die Netze aus Abbildung 1 herleiten.

Die Netze für AND und OR sind sehr einfach und sie liefern offensichtlich das korrekte Ergebnis. Für das XOR Netz ist das nicht mehr so einfach einzusehen, daher überprüfen wir die Funktionsweise zu Fuß. Zunächst fällt auf, dass das hintere Neuron ein AND Element darstellt. Das heißt, damit das gesamte Netz feuert, müssen beide der mittleren Neuronen gefeuert haben. Wir bezeichnen die mittleren Neuronen mit O_1 und O_2 (von oben nach unten). Also:

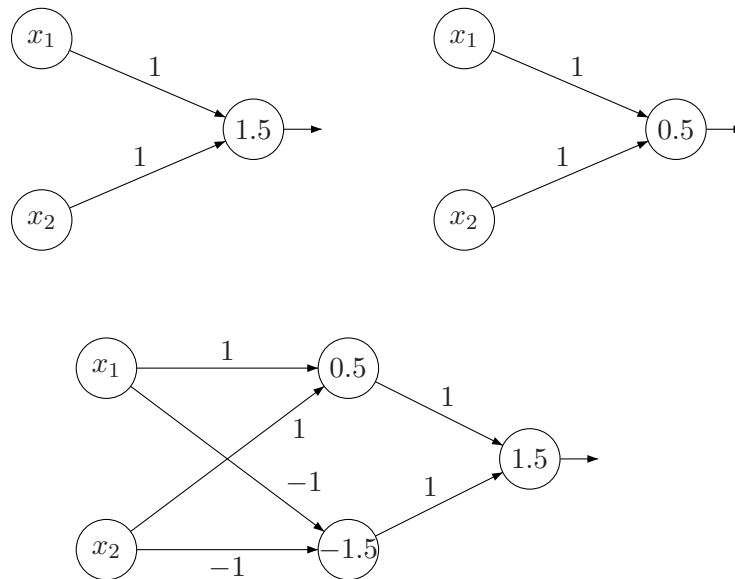


Abbildung 1: Neuronale Netze für AND, OR und XOR

- Sei $(x_1, x_2) = (0, 0)$. Dann feuert das Neuron O_1 nicht, das heißt das Ergebnis ist 0.
- Sei $(x_1, x_2) = (0, 1)$. Dann feuert das Neuron O_1 . Auch das Neuron O_2 feuert, da $-1 \cdot 0 - 1 \cdot 1 = -1 \geq -1.5$ erfüllt ist.
- Sei $(x_1, x_2) = (1, 0)$. Diese Situation ist symmetrisch zur Vorherigen, also feuern wieder beide Neuronen O_1 und O_2 .
- Sei $(x_1, x_2) = (1, 1)$. Es feuert zwar das Neuron O_1 , aber für O_2 gilt $-1 \cdot 1 - 1 \cdot 1 = -2 \not\geq -1.5$. Das Netz feuert also nicht.

Wir sehen dass das Netz genau unsere Anforderung, nämlich die XOR Funktion zu berechnen, erfüllt.

- (b) Die „kanonische“ Frage ist nun, ob sich das Netz auch mit weniger Neuronen implementieren lässt, wo sich doch andere logische Operationen wie AND und OR offensichtlich mit einem Neuron realisieren lassen. Wir beweisen, dass die XOR Funktion nicht mit einem einzigen Neuron realisiert werden kann.

Nehmen wir an, wir hätten ein Neuron mit den Gewichten w_1 und w_2 sowie dem Schwellwert σ , das die XOR Funktion realisiert. Die XOR Funktion stellt folgende Anforderungen (Gleichungen) an das Neuron:

- (1) $0 \cdot w_1 + 0 \cdot w_2 < \sigma$
- (2) $0 \cdot w_1 + 1 \cdot w_2 \geq \sigma$
- (3) $1 \cdot w_1 + 0 \cdot w_2 \geq \sigma$
- (4) $1 \cdot w_1 + 1 \cdot w_2 < \sigma$

Addiert man Gleichung (2) auf (3) so ergibt sich $w_1 + w_2 \geq 2\sigma$. Wegen Gleichung (4) können wir schreiben

$$\sigma > w_1 + w_2 \geq 2\sigma \quad \Rightarrow \quad \sigma > 2\sigma$$

Die Gleichung $\sigma > 2\sigma$ kann aber nur für $\sigma < 0$ erfüllt werden, und das ist ein Widerspruch zu Gleichung (1). \square

Aufgabe 2.

Betrachten Sie [Abbildung 2](#).

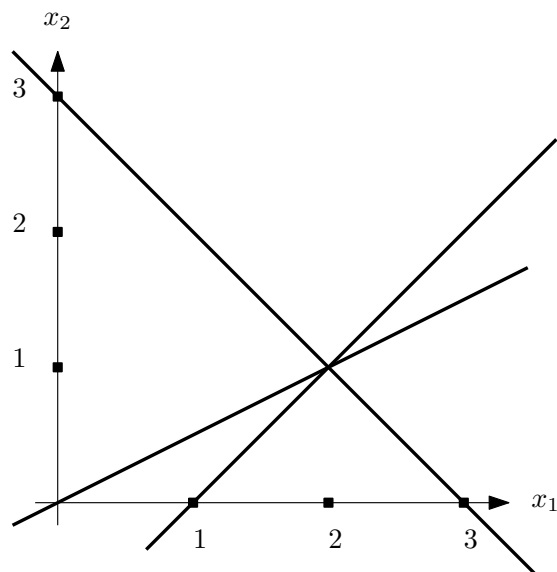


Abbildung 2: Ein Geradenbüschel.

Konstruieren Sie ein neuronales Netz mit den Neuron-Zuständen $Q = \{0, 1\}$, das genau dann eine eins ausgibt, wenn die Eingabe ein Punkt $(x_1, x_2)^T$ auf einer der Geraden ist.

Lösung.

Wir wissen, dass ein Neuron mit n Eingabewerten einem Halbraum im \mathbb{R}^n entspricht. Da wir hier den \mathbb{R}^2 vorliegen haben, haben wir zwei Eingaben (x_1, x_2) . Offensichtlich müssen Punkte auf Geraden klassifiziert werden, das heißt Punkte die Gleichungen der Form $w_1x_1 + w_2x_2 = \sigma$ ¹ erfüllen. Unsere Neuronen sind allerdings nur für „ \geq “ definiert. Aber das macht nichts, denn wir können eine Gleichung der Form $w_1x_1 + w_2x_2 = \sigma$ durch zwei Gleichungen ausdrücken, die beide gleichzeitig erfüllt sein müssen, nämlich $w_1x_1 + w_2x_2 \geq \sigma$ und $-w_1x_1 - w_2x_2 \geq -\sigma$ ². Die beiden Gleichungen müssen wir in unserem Netz mit einem „UND-Neuron“ verknüpfen, um zu erzwingen, dass beide erfüllt sein sollen.

Da wir eine eins ausgehen sollen, wenn ein Punkt auf *mindestens einer* der Geraden ist, werden wir die Ergebnisse, von unseren Geradengleichungen durch ein OR Neuron verknüpfen, und erhalten damit das gewünschte Verhalten.

Die Geradengleichungen ergeben sich durch:

- Gerade 1: $x_2 = -x_1 + 3 \Leftrightarrow x_1 + x_2 = 3$
- Gerade 2: $x_2 = \frac{1}{2}x_1 \Leftrightarrow -\frac{1}{2}x_1 + x_2 = 0$
- Gerade 3: $x_2 = x_1 - 1 \Leftrightarrow -x_1 + x_2 = -1$

Die Gewichte sowie die Schwellwerte für die Neuronen, die die Geraden modellieren, können wir jetzt einfach ablesen und erhalten als Gesamtes das Netz aus Abbildung 3.

¹Koordinatendarstellung einer Hyperebene, in unserem Fall also einer Geraden

²Das haben wir ja schon oft gesehen. Vgl. Lineare Programme.

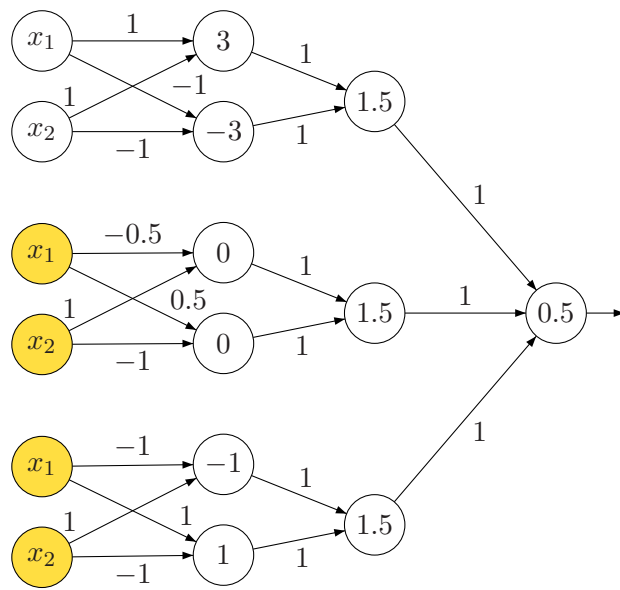


Abbildung 3: Neuronales Netz zur Erkennung der Geraden. Die gelben Eingabeneuronen sind nur der Übersicht wegen eingezeichnet. Eigentlich sind die Neuronen für x_1 und x_2 nur einmal vorhanden!