

NN.Perzeptron.01

1. Trennebene und +1-Bereich

Gegebenes Perzeptron:

$$w = (2, 1, 1)$$

Entscheidungsgrenze:

$$2 + x_1 + x_2 = 0$$

$$\Leftrightarrow x_2 = -x_1 - 2$$

Dies ist eine Gerade mit Steigung -1 und Achsenabschnitt -2 .

Der Bereich mit Klassifikation $+1$ ist:

$$2 + x_1 + x_2 > 0$$

$$\Leftrightarrow x_2 > -x_1 - 2$$

→ also die Halbebene oberhalb der Geraden.

2. Vergleich der anderen Perzeptrons

Zwei Perzeptrons haben dieselbe Trennebene, wenn ihr Gewichtungsvektor ein skalares Vielfaches des Originals ist ($k \neq 0$).

Sie haben dieselbe Klassifikation, wenn $k > 0$.

Bei $k < 0$ kehren sich die Klassen um.

Vergleich:

$$(1, 0.5, 0.5)$$

$k = 0.5 \rightarrow$ gleiche Trennebene, gleiche Klassifikation

$$(200, 100, 100)$$

$k = 100 \rightarrow$ gleiche Trennebene, gleiche Klassifikation

$$(2, 1, 1)$$

$k = 1 \rightarrow$ gleiche Trennebene, gleiche Klassifikation

$$(-2, -1, -1)$$

$k = -1 \rightarrow$ gleiche Trennebene, aber invertierte Klassifikation

NN.Perzeptron.02

1. Logische Funktionen mit einem Perzeptron (1,5 P)

Wir verwenden binäre Eingaben $x_1, x_2 \in \{0, 1\}$ und die Entscheidungsregel:

$$y = 1, \text{ falls } w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 \geq 0$$

$$y = 0, \text{ sonst}$$

Alle gezeigten Funktionen (AND, OR, NOT) sind linear separierbar.

1.1 UND (AND)

Gewichte:

$$w_0 = -1,5$$

$$w_1 = 1$$

$$w_2 = 1$$

Entscheidung:

$$y = 1, \text{ falls } -1,5 + x_1 + x_2 \geq 0$$

Wahrheitstabelle (AND):

x1	x2	Summe	y	AND
0	0	-1,5	0	0
0	1	-0,5	0	0
1	0	-0,5	0	0
1	1	0,5	1	1

1.2 ODER (OR)

Gewichte:

$$w_0 = -0,5$$

$$w_1 = 1$$

$$w_2 = 1$$

Entscheidung:

$$y = 1, \text{ falls } -0,5 + x_1 + x_2 \geq 0$$

Wahrheitstabelle (OR):

x1	x2	Summe	Y	OR
0	0	-0,5	0	0
0	1	0,5	1	1
1	0	0,5	1	1
1	1	1,5	1	1

1.3 KOMPLEMENT (NOT)

Unäre Funktion, deshalb wird x2 nicht verwendet.

Gewichte:

$$w_0 = 0,5$$

$$w_1 = -1$$

Entscheidung:

$$y = 1, \text{ falls } 0,5 - x_1 \geq 0$$

Wahrheitstabelle (NOT):

x1	Summe	y	NOT
0	0,5	1	1
1	-0,5	0	0

2. XOR – Warum nicht mit einem Perzeptron? (0,5 P)

XOR ist nicht linear separierbar, d. h. es gibt keine einzige Gerade (bzw. Hyperebene), die die Klassen (Ausgabe 0 und 1) für alle vier Eingabekombinationen korrekt trennen kann.

XOR-Tabelle:

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

NN.Perzeptron.03

1. Datensatz-Konstruktion (1P)

Ziel:

Einen Datensatz D mit zufälligen Punkten und einer zufälligen linearen Zielfunktion f konstruieren.

Vorgehen:

1. Wähle m Punkte $x(j) = (x_1(j), x_2(j))$ gleichverteilt aus dem Quadrat $X = [-1, 1] \times [-1, 1]$.
2. Wähle zwei weitere zufällige Punkte P1 und P2 aus $[-1, 1] \times [-1, 1]$.
Die Gerade durch P1 und P2 definiert die Entscheidungsgrenze der Zielfunktion f.
3. Bestimme eine Geradengleichung durch P1 und P2 der Form $a \cdot x_1 + b \cdot x_2 + c = 0$
Zum Beispiel über den Normalenvektor:
 $a = p2_y - p1_y$
 $b = p1_x - p2_x$
 $c = p2_x \cdot p1_y - p1_x \cdot p2_y$
4. Beschrifte jeden Datenpunkt $x(j)$ durch die Zielfunktion:
 $y(j) = \text{sgn}(a \cdot x_1(j) + b \cdot x_2(j) + c)$
Wertebereich: $\{-1, +1\}$.
Punkte mit positivem Ausdruck erhalten Label +1, die anderen -1.

Damit entsteht ein linear separierbarer Datensatz.

Für $m = 10$ erhältst du so den geforderten Datensatz D.

2. Perzeptron-Lernalgorithmus (Training, 3P)

Wir nutzen die erweiterte Eingabedarstellung:

Eingabepunkt:

$$x_tilde(i) = (1, x_1(i), x_2(i))$$

Gewichtsvektor:

$$w = (w_0, w_1, w_2)$$

Hypothese:

$$h(x) = \text{sgn}(w \cdot x_tilde)$$

Die gegebene Update-Regel lautet:

$$w := w + \alpha \cdot (y(i) - h(x(i))) \cdot x_tilde(i)$$

mit Lernrate $\alpha = 1$.

Ablauf eines Trainingsdurchlaufs:

1. Initialisiere die Gewichte:
 $w = (0, 0, 0)$
2. Wiederhole, bis alle Punkte korrekt klassifiziert sind:
 - Berechne $h(x(i))$ für alle Punkte
 - Bestimme die Menge der falsch klassifizierten Punkte:
 $M = \{ i \mid h(x(i)) \neq y(i) \}$
 - Falls M leer ist \rightarrow Algorithmus konvergiert
 - Sonst wähle zufällig ein i aus M
 - Update:
 $w := w + (y(i) - h(x(i))) \cdot x_tilde(i)$
 - Erhöhe Zähler der Updates um 1
- 3.

Gesuchte Größe:

Die Anzahl der Updates bis zur Konvergenz.

Die Aufgabe verlangt:

Führe den Algorithmus für $m = 10$ insgesamt 1000-mal aus.

Dabei kann man

- jedes Mal denselben Datensatz verwenden oder
- für jeden Lauf einen neuen Datensatz erzeugen (beides interpretierbar).

Typische empirische Werte:

Bei $m = 10$ und $\alpha = 1$ liegt die mittlere Anzahl benötigter Schritte ungefähr im Bereich 15 bis 30.

3. Experimente (2P)

Es werden größere Datensätze getestet:

$m = 100$ und $m = 1000$

jeweils mit Lernraten $\alpha = 1$ und $\alpha = 0,1$.

Für jede Kombination (m , α):

1. Erzeuge einen neuen Datensatz nach dem Verfahren aus Teil 1.
2. Führe den Perzeptronalgorithmus viele Male aus (z. B. 100 bis 1000 Durchläufe).
3. Miss die Schrittzahl bis zur Konvergenz.
4. Berechne den Mittelwert.

Typische Größenordnungen (empirisch):

- $m = 10$, $\alpha = 1$
Schritte $\approx 10^1$ (z. B. 15–30)
- $m = 100$, $\alpha = 1$
Schritte im Bereich 10^2 bis 10^3
- $m = 100$, $\alpha = 0,1$
deutlich mehr Schritte, ca. 10^3
- $m = 1000$, $\alpha = 1$
ca. 10^3 bis 10^4
- $m = 1000$, $\alpha = 0,1$
ca. 10^4 bis 10^5

Grund für die Unterschiede:

- Größeres $m \rightarrow$ mehr Punkte \rightarrow mehr mögliche Fehlklassifikationen \rightarrow mehr Updates.
- Kleinere Lernrate $\alpha \rightarrow$ kleinere Updates \rightarrow langsamerer Fortschritt.