Boolesche Funktionen und Schaltkreise

Die *Oder-Funktion* (Disjunktion) $\vee$ und die *Und-Funktion* (Konjunktion) $\wedge$,

x	y	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

x	y	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

$\rightarrow$ (Implikationsfunktion), $\leftarrow$ (umgekehrte Implikation) und $\oplus$ (Xor-Funktion, exclusive or),

x	y	$\rightarrow$
0	0	1
0	1	1
1	0	0
1	1	1

x	y	$\leftarrow$
0	0	1
0	1	0
1	0	1
1	1	1

x	y	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

Berechnungen mit Konstanten

$$x \vee 0 = x, \quad x \vee 1 = 1, \quad x \wedge 0 = 0, \quad x \wedge 1 = x,$$
$$x \oplus 0 = x, \quad x \oplus 1 = \neg x$$

Assoziativität

$$x \vee (y \vee z) = (x \vee y) \vee z,$$
$$x \wedge (y \wedge z) = (x \wedge y) \wedge z,$$
$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$

Kommutativität

$$x \vee y = y \vee x, \quad x \wedge y = y \wedge x, \quad x \oplus y = y \oplus x$$

Distributivität

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z),$$
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z),$$
$$x \wedge (y \oplus z) = (x \wedge y) \oplus (x \wedge z)$$

Vereinfachungsgesetze

$$x \vee x = x, \quad x \vee \neg x = 1, \quad x \wedge x = x,$$
$$x \wedge \neg x = 0, \quad x \oplus x = 0, \quad x \oplus \neg x = 1$$

Absorptionsgesetze

$$x \vee (x \wedge y) = x, \quad x \wedge (x \vee y) = x$$

deMorgansche Gesetze, Negationsgesetze

$$\neg x \vee \neg y = \neg(x \wedge y), \quad \neg x \wedge \neg y = \neg(x \vee y),$$
$$\neg x \oplus \neg y = x \oplus y, \quad \neg\neg x = x$$

Ersetzen der Implikation und der Antivalenz

$$x \rightarrow y = \neg x \vee y, \quad x \leftarrow y = x \vee \neg y,$$

$$x \oplus y = (x \vee y) \wedge (\neg x \vee \neg y) = (\neg x \wedge y) \vee (x \wedge \neg y)$$

Satz Jede Booleschen Funktion läßt sich sowohl in disjunktiver Normalform (als Disjunktion von Mintermen) auch in konjunktiver Normalform (als Konjunktion von Maxtermen) darstellen.

Unformung von einer Formel F in eine äquivalente DNF Formel

1: F in eine Formel mit nur $\vee, \wedge$ und $\neg$ unformen.

2: Negationszeichen direkt vor den Variablen bringen.

3: Distributivgesetze anwenden.

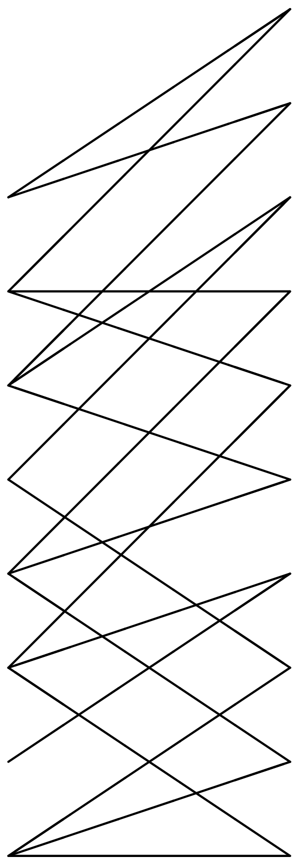
Minimierung von Normalformen, Karnaugh-Veitch-Diagramme

(für Formeln mit weniger als 4 oder 5 Variablen)

- Minterme sind im Diagram mit einem Punkt markiert.
- 8er, 4er, 2er und 1er Blöcke zusammenfassen (möglichst Große Blöcke).
- Die Blöcke dürfen sich überlappen und dürfen über die Kanten hinweg definiert werden.

1. $x_1\,x_2\,x_3\,x_4$
2. $x_1\,x_2\,x_3\,\overline{x_4}$
3. $x_1\,x_2\,\overline{x_3}\,x_4$
4. $\overline{x_1}\,x_2\,\overline{x_3}\,x_4$
5. $x_1\,x_2\,\overline{x_3}\,\overline{x_4}$
6. $\overline{x_1}\,x_2\,x_3\,\overline{x_4}$
7. $\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}$
8. $\overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4}$

1. $x_1\,x_2\,x_3\,x_4$
2. $x_1\,x_2\,x_3\,\overline{x_4}$
3. $x_1\,x_2\,\overline{x_3}\,x_4$
4. $\overline{x_1}\,x_2\,\overline{x_3}\,x_4$
5. $x_1\,x_2\,\overline{x_3}\,\overline{x_4}$
6. $\overline{x_1}\,x_2\,x_3\,\overline{x_4}$
7. $\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}$
8. $\overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4}$



$x_1\,x_2\,x_3$

$x_1\,x_2\,x_4$

$x_2\,\overline{x_3}\,x_4$

$x_1\,x_2\,\overline{x_4}$

$x_2\,x_3\,\overline{x_4}$

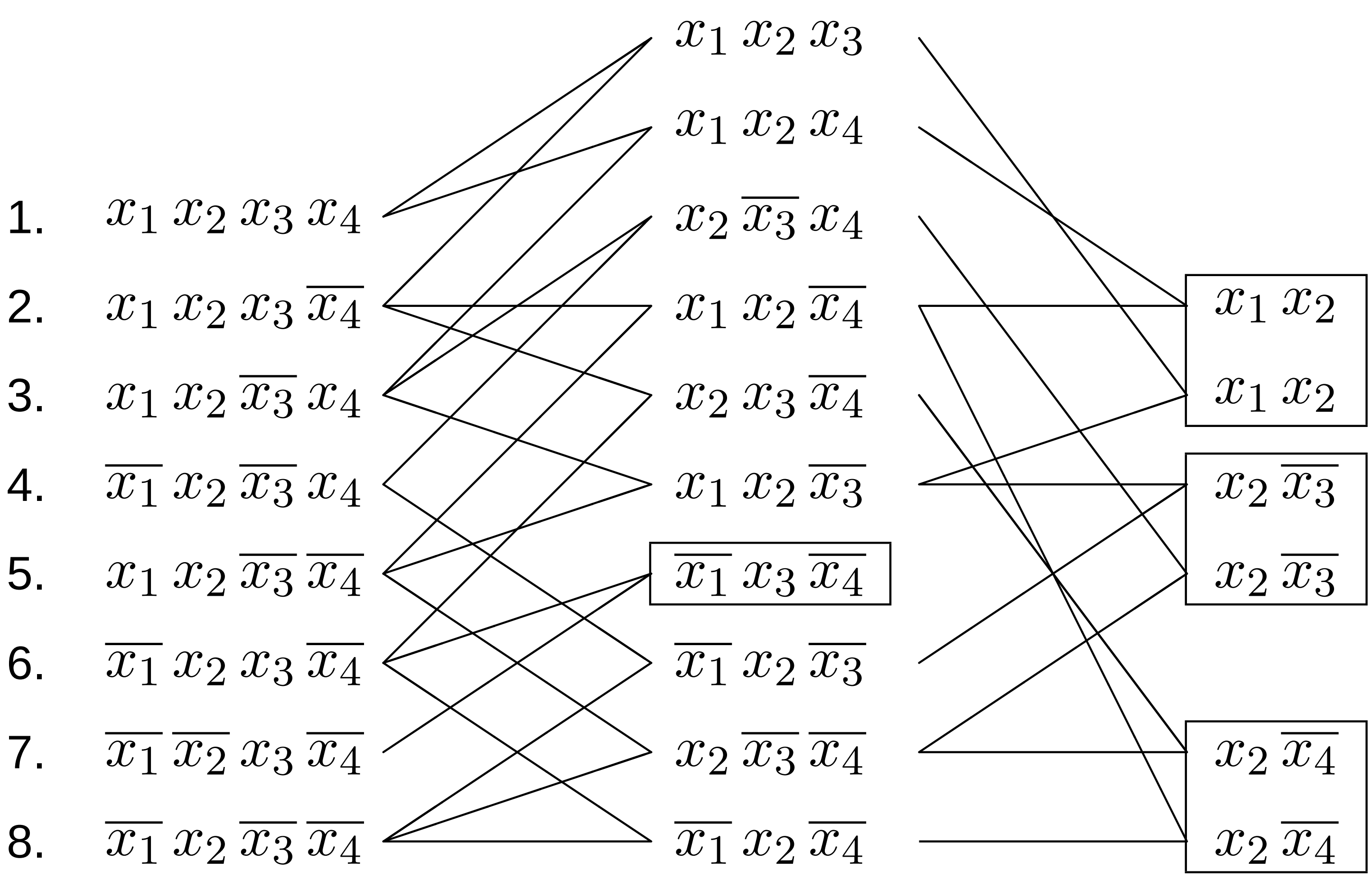
$x_1\,x_2\,\overline{x_3}$

$\overline{x_1}\,x_3\,\overline{x_4}$

$\overline{x_1}\,x_2\,\overline{x_3}$

$x_2\,\overline{x_3}\,\overline{x_4}$

$\overline{x_1}\,x_2\,\overline{x_4}$


1. $x_1\, x_2\, x_3\, x_4$
2. $x_1\, x_2\, x_3\, \overline{x_4}$
3. $x_1\, x_2\, \overline{x_3}\, x_4$
4. $\overline{x_1}\, x_2\, \overline{x_3}\, x_4$
5. $x_1\, x_2\, \overline{x_3}\, \overline{x_4}$
6. $\overline{x_1}\, x_2\, x_3\, \overline{x_4}$
7. $\overline{x_1}\, \overline{x_2}\, x_3\, \overline{x_4}$
8. $\overline{x_1}\, x_2\, \overline{x_3}\, \overline{x_4}$
$x_1\, x_2\, x_3$
$x_1\, x_2\, x_4$
$x_2\, \overline{x_3}\, x_4$
$x_1\, x_2\, \overline{x_4}$
$x_2\, x_3\, \overline{x_4}$
$x_1\, x_2\, \overline{x_3}$
$\boxed{\overline{x_1}\, x_3\, \overline{x_4}}$
$\overline{x_1}\, x_2\, \overline{x_3}$
$x_2\, \overline{x_3}\, \overline{x_4}$
$\overline{x_1}\, x_2\, \overline{x_4}$
$x_1\, x_2$
$x_1\, x_2$
$x_2\, \overline{x_3}$
$x_2\, \overline{x_3}$
$x_2\, \overline{x_4}$
$x_2\, \overline{x_4}$

	1	2	3	4	5	6	7	8
$\overline{x_1}\, x_3\, \overline{x_4}$								
$x_1\, x_2$								
$x_2\, \overline{x_3}$								
$x_2\, \overline{x_4}$								

	1	2	3	4	5	6	7	8
$\overline{x_1}\, x_3\, \overline{x_4}$						$+$	$+$	
$x_1\, x_2$	$+$	$+$	$+$		$+$			
$x_2\, \overline{x_3}$			$+$	$+$	$+$			$+$
$x_2\, \overline{x_4}$		$+$			$+$	$+$		$+$

Ringsummennormalform

Für jede Boolesche Funktion $f : B^n \longrightarrow B$ gibt es genau einen 0-1-Vektor $a = (a_{T_1}, a_{T_2}, \ldots, a_{T_{2^n}})$ mit $T_i \subseteq \{1, \ldots, n\}$ und $a_{T_i} \in \{0, 1\}$ so dass

$$f(x_1, \ldots, x_n) = \bigoplus_{T \subseteq \{1,\ldots,n\}} \left(a_T \wedge \bigwedge_{j \in T} x_j \right)$$

Schaltkreise

Definition: Ein Schaltkreis über einer Basis Ω von Grundfunktionen (zum Beispiel $\Omega = \{\vee, \wedge, \neg\}$) ist ein gerichteter, azyklischer Graph. Die Eingangsknoten sind mit Variablennamen oder den Konstanten 0,1 beschriftet, während die inneren Knoten mit Elementen aus Ω beschriftet sind.

Funktion $res_g(x_1, \ldots, x_n)$ die von einem Gatter g berechnet wird

- Falls g ein Eingangsknoten ist, der mit der Variablen 0 (bzw. 1) beschriftet ist, so ist $res_g(x_1, \ldots, x_n) = 0$ (bzw. $= 1$).

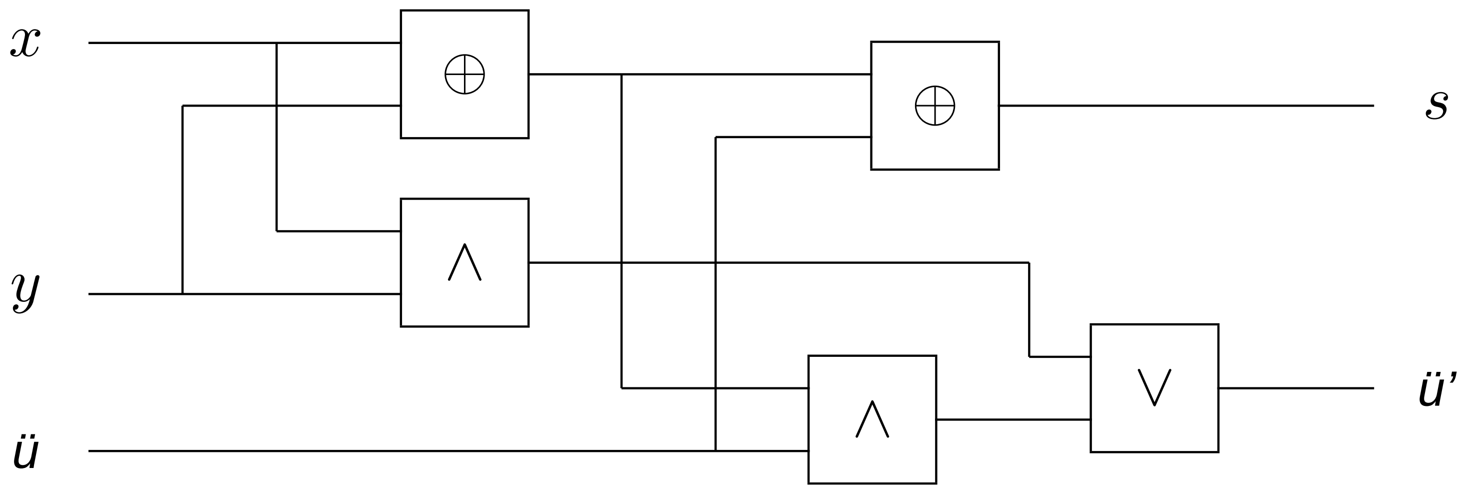
- Falls g ein Eingangsknoten ist, der mit der Variablen x_i beschriftet ist, so ist $res_g(x_1, \ldots, x_n) = x_i$.

- Falls g ein innerer Knoten ist, der mit $f \in \Omega$ beschriftet ist, und falls k die Stelligkeit von f ist, so seien $f_1, \ldots, f_k$ die an den Vorgängerknoten von g berechneten Funktionen. Dann ist

$$res_g(x_1, \ldots, x_n) = f(f_1(x_1, \ldots, x_n), \ldots, f_k(x_1, \ldots, x_n))$$

Wir sagen, dass ein Schaltkreis S eine Boolesche Funktion f *berechnet*, falls es einen Knoten g in S gibt mit $f = res_g$.

Volladdierer



Schaltkreiskomplexität

Die **Größe** eines Schaltkreises S, $C(S)$,ist die Anzahl seiner Gatter.

Die **Tiefe** eines Schaltkreises S, $D(S)$ ist die Länge eines längsten Pfades im Schaltkreis.

Für eine Boolesche Funktion f und eine Basis Ω

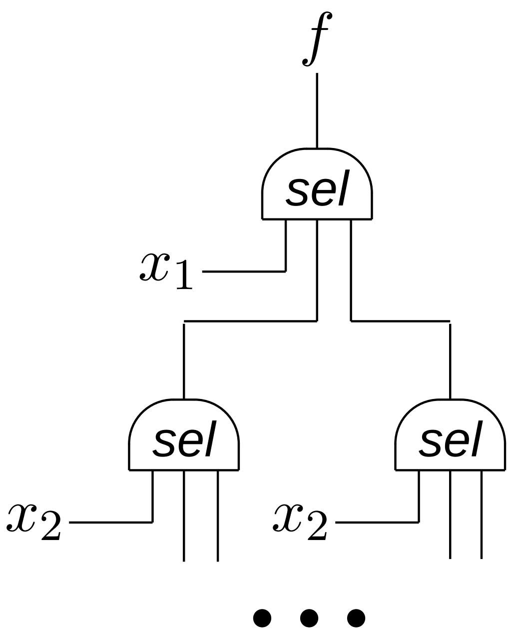
$$
\begin{aligned}
C_\Omega(f) &= \min\{C(S) \mid S \text{ ist ein } \Omega\text{-Schaltkreis, der } f \text{ berechnet}\} \\
D_\Omega(f) &= \min\{D(S) \mid S \text{ ist ein } \Omega\text{-Schaltkreis, der } f \text{ berechnet}\}
\end{aligned}
$$

Satz Seien Ω und Ω' zwei vollständige Basen. Sei $c = \max\{C_\Omega(g) \mid g \in \Omega'\}$ und sei $d = \max\{C_{\Omega'}(g) \mid g \in \Omega\}$. Dann ist für alle Booleschen Funktionen f, $C_\Omega(f) \leq c \cdot C_{\Omega'}(f)$ und $C_{\Omega'}(f) \leq d \cdot C_\Omega(f)$.

Obere Schranken für $f : B^n \to B$

Mit der DNF, Größe $O(n2^n)$ und Tiefe $n + \log n$.

Mit der sel Funktion, Größe $O(2^n)$ und Tiefe n.



Obere Schranken II

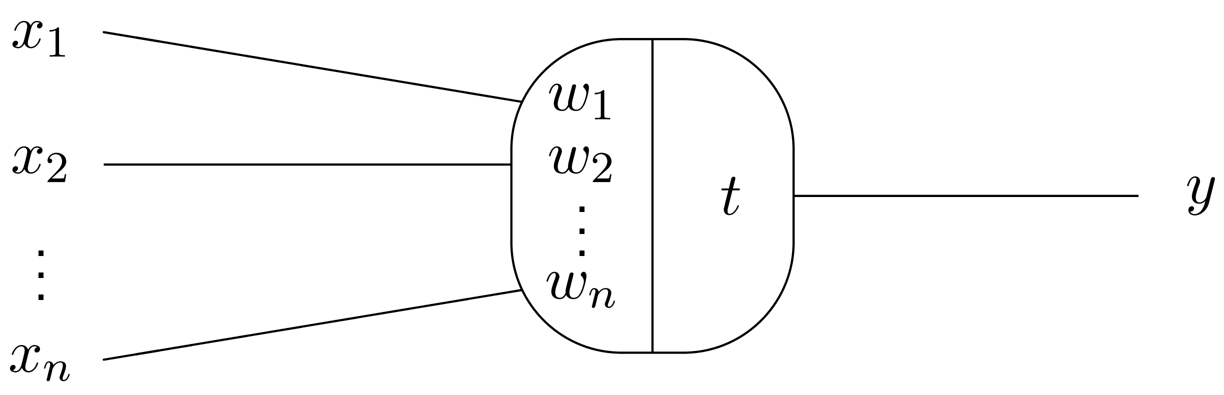
Satz: (Lupanov, 1958) Es gibt Konstanten c und n_0, so dass für alle n-stelligen Booleschen Funktionen f und $n \geq n_0$ gilt $C(f) \ \leq \ c \cdot 2^n/n$.

Untere Schranken

Satz: (Shannon, 1949) Für jedes n gibt es n-stellige Boolesche Funktionen f, die für ihre Realisierung (mittels Nand-Gattern) mindestens $\frac{1}{2} \cdot 2^n/n$ viele Gatter benötigen. Also $C_{\{nand\}}(f) \geq \frac{1}{2} \cdot 2^n/n$.

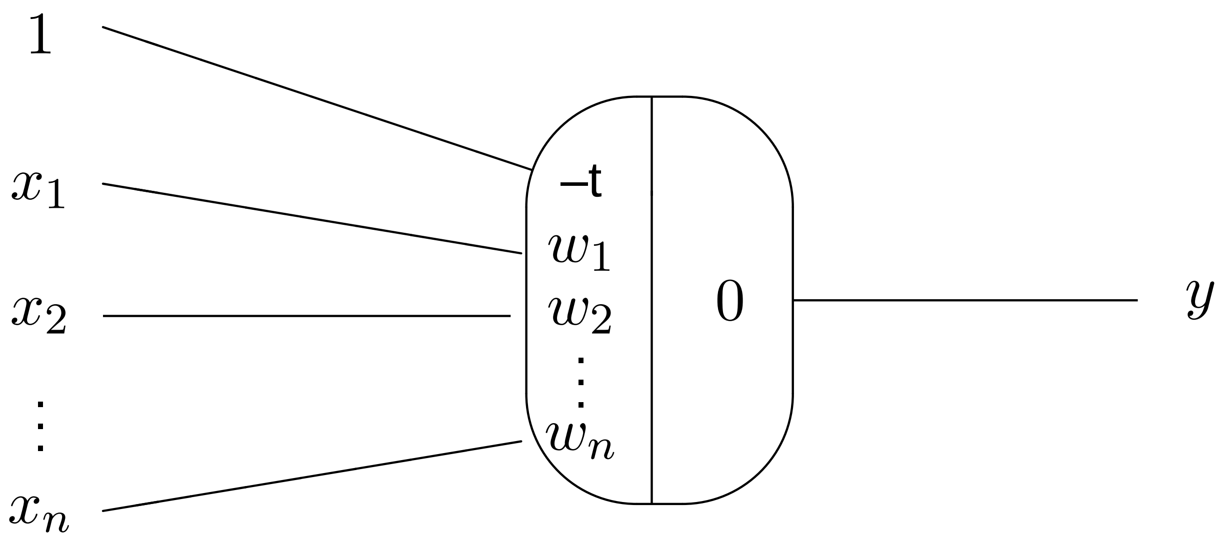
Folgerung: Es gibt Konstanten c und n_0, so dass es für jedes $n \geq n_0$ eine n-stellige Boolesche Funktion f gibt mit $C(f) \ \geq \ c \cdot 2^n/n$.

Perzeptrone



$w_i, t \in R$

$$y = \begin{cases} 1, & \sum_{i=1}^{n} x_i w_i \geq t, \\ 0, & \text{sonst} \end{cases}$$


1
x_1
x_2
$\vdots$
x_n
$-t$
w_1
w_2
$\vdots$
w_n
0
y

Definition: $f : B^n \longrightarrow B$ ist linear separierbar falls es eine Hyperebene der Dimension $n-1$ gibt, die die 0'n von den 1'n von f trennt.

Die durch ein Perzeptron berechenbaren Funktionen sind genau die linear separierbaren Funktionen.

Lernen von Booleschen Funktionen

Y_0 und Y_1 disjunkte, und linear separierbare, Mengen von “Beispielen”.

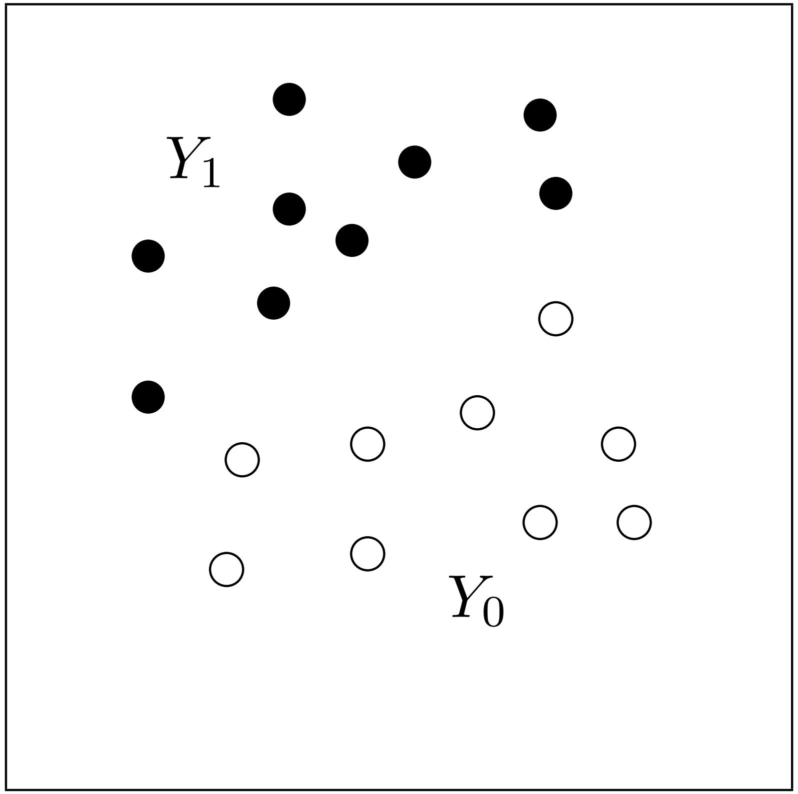
Wir wollen einen Gewichtsvektor w finden mit:

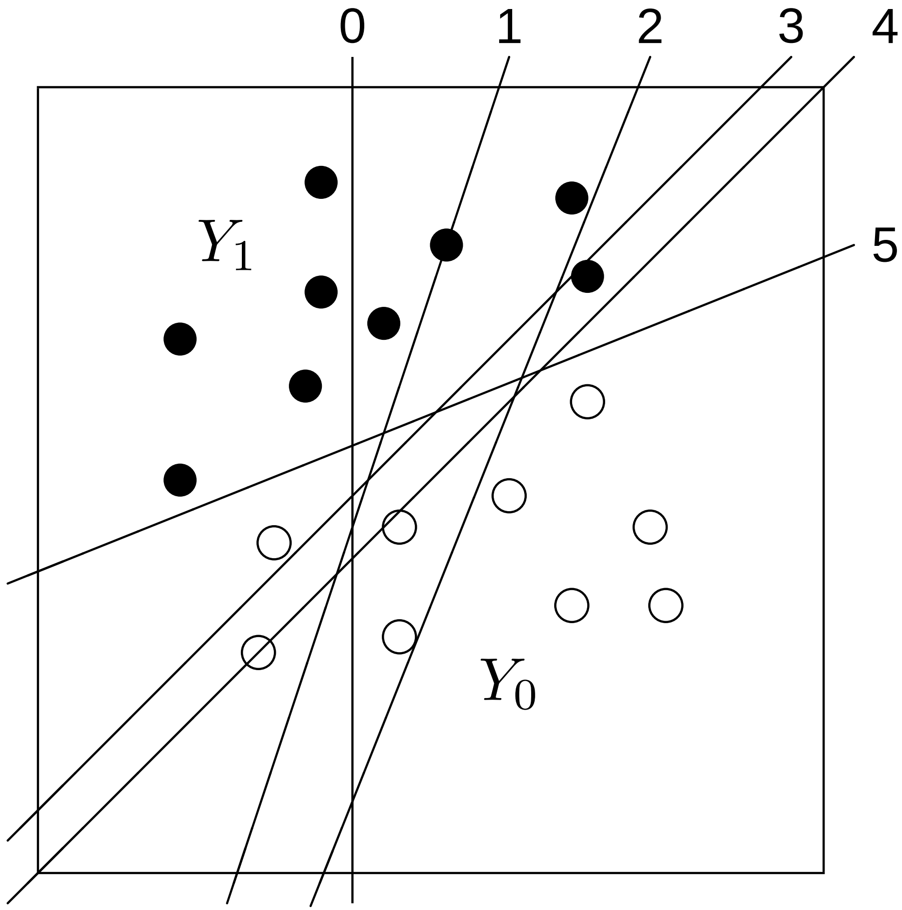
$$
\begin{aligned}
y \in Y_1 \quad &\Rightarrow \quad yw > 0 \\
y \in Y_0 \quad &\Rightarrow \quad yw < 0
\end{aligned}
$$

$w^0 := (0, 0, \ldots, 0)$

Wird dem Perzeptron ein Beispiel y vorgelegt, und dieses wird falsch klassifiziert, so wird der Gewichtsvektor im nächsten Schritt wie folgt von w^t zu w^{t+1} geändert:

$$w^{t+1} = \begin{cases} w^t + y, & \text{falls } y \in Y_1 \text{ und } yw \leq 0 \\ w^t - y, & \text{falls } y \in Y_0 \text{ und } yw \geq 0 \\ w^t, & \text{sonst} \end{cases}$$


Y_1
Y_0


0
1
2
3
4
5
Y_1
Y_0

Perzeptron-Konvergenztheorem

Satz: Gegeben seien zwei linear separierbare Mengen von Punkten Y_0, Y_1 und eine unendliche Trainingsfolge, mit der Eigenschaft, dass jedes Element aus $Y_0 \cup Y_1$ in der Treiningsfolge unendlich oft auftrtt. Dann erreicht das Perzeptron nach endlich vielen Lernschritten (gemäß der oben beschriebenen Lernregel) einen Gewichtsvektor, so dass das Perzeptron allen Punkten in Y_0 den Wert 0, und allen Punkten in Y_1 den Wert 1 zuweist.

Boolesche Funktionen, Zusammenfassung

Beschreibungsmöglichkeiten: Wahrheitstafel, DNF, KNF, RSNF, Boolesche Schaltkreise über eine vollständige Basis, Perzeptrone...

Wie Groß sind solche Beschreibungen? Sind sie eindeutig? Können sie minimiert werden?

Wie kann man die Modelle gegenseitig simulieren? Was sind die Vorteile der verschiedenen Modelle.