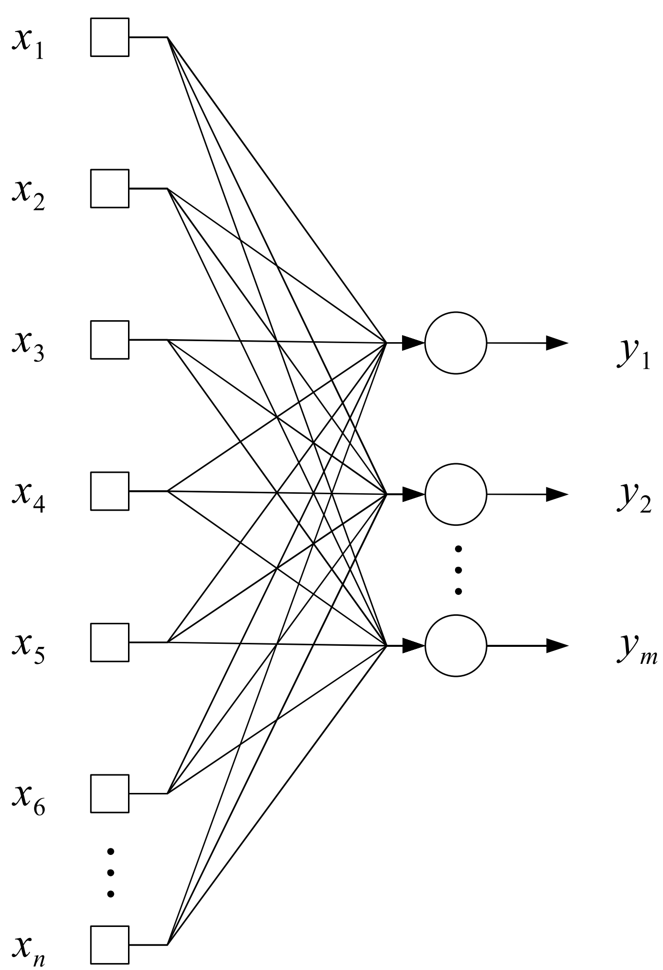
Multilayer-Perzeptrons (MLPs)
und
Backpropagation

Netzaufbau Perzeptron/Adaline



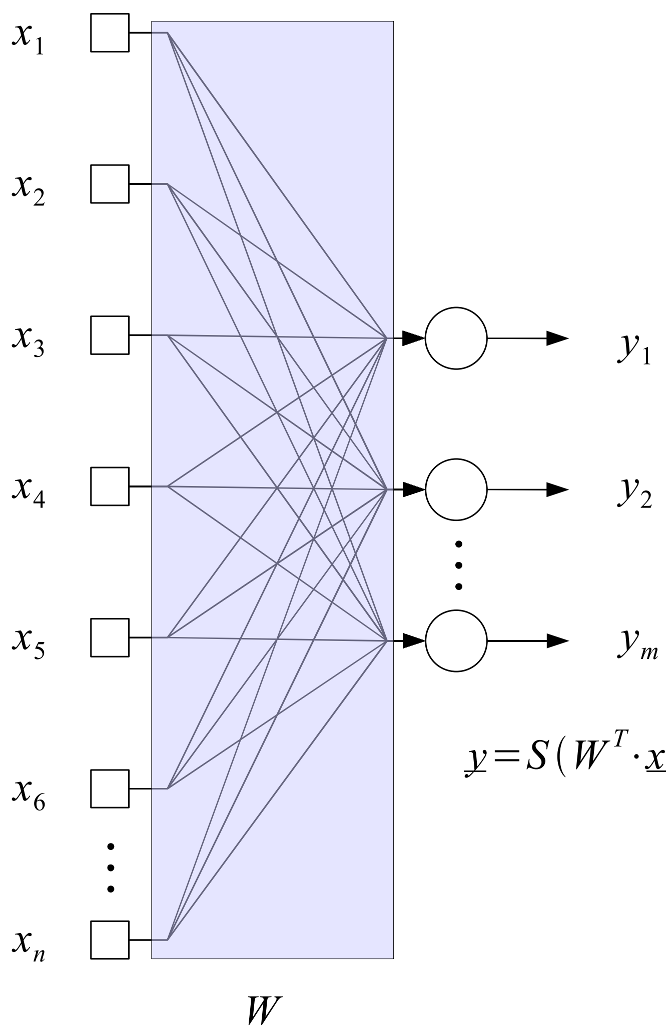
Perzeptron

$\underline{x} \in \{0,1\}^n$ $\quad L(\underline{x}) \in \{0,1\}^m$ $\quad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

Adaline

$\underline{x} \in \mathbb{R}^n$ $\quad L(\underline{x}) \in \mathbb{R}^m$ $\quad S(z)$ beliebig

Netzaufbau Perzeptron/Adaline



$$\underline{y} = S(W^T \cdot \underline{x})$$

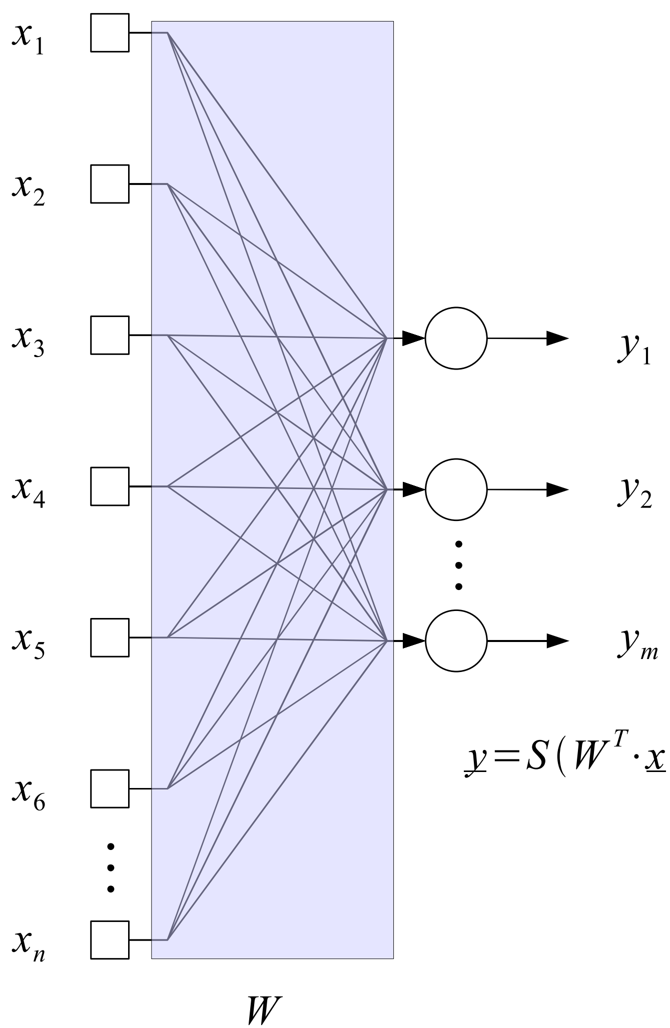
Perzeptron

$\underline{x} \in \{0,1\}^n \qquad L(\underline{x}) \in \{0,1\}^m \qquad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

Adaline

$\underline{x} \in \mathbb{R}^n \qquad L(\underline{x}) \in \mathbb{R}^m \qquad S(z)$ beliebig

Netzaufbau Perzeptron/Adaline



$\underline{y} = S(W^T \cdot \underline{x})$

Perzeptron

$\underline{x} \in \{0,1\}^n \qquad L(\underline{x}) \in \{0,1\}^m \qquad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

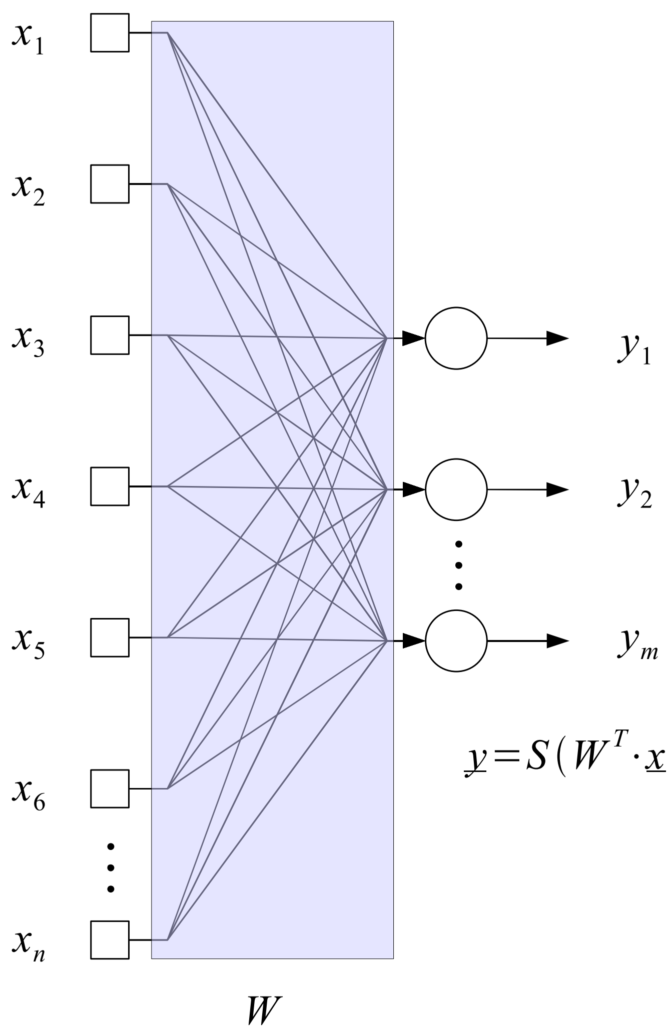
Lernregel: $W \leftarrow W - \gamma \cdot \underline{x} \cdot (\underline{y} - L(\underline{x}))^T$

Adaline

$\underline{x} \in \mathbb{R}^n \qquad L(\underline{x}) \in \mathbb{R}^m \qquad S(z)$ beliebig

Lernregel: $W \leftarrow W - \gamma \cdot \frac{1}{\underline{x}^T \underline{x}} \cdot \underline{x} \cdot (W^T \underline{x} - L(\underline{x}))^T$

Netzaufbau Perzeptron/Adaline



$\underline{y} = S(W^T \cdot \underline{x})$

Perzeptron

$\underline{x} \in \{0,1\}^n \qquad L(\underline{x}) \in \{0,1\}^m \qquad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

Lernregel: $W \leftarrow W - \gamma \cdot \underline{x} \cdot \underline{\delta}^T \qquad \underline{\delta} := \underline{y} - L(\underline{x})$

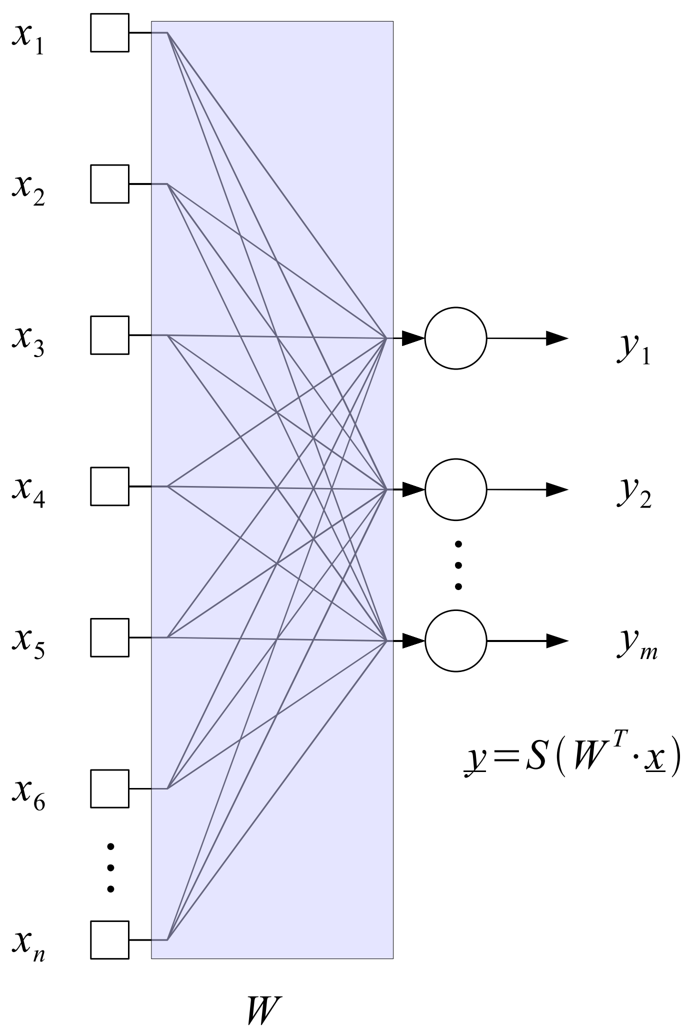
Adaline

$\underline{x} \in \mathbb{R}^n \qquad L(\underline{x}) \in \mathbb{R}^m \qquad S(z)$ beliebig

Lernregel: $W \leftarrow W - \gamma \cdot \frac{1}{\underline{x}^T \underline{x}} \cdot \underline{x} \cdot \underline{\delta}^T \qquad \delta := W^T \underline{x} - L(\underline{x})$

Wiederholung Singlelayer Netzwerke

Netzaufbau Perzeptron/Adaline



Perzeptron

$\underline{x} \in \{0,1\}^n \qquad L(\underline{x}) \in \{0,1\}^m \qquad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

Lernregel: $W \leftarrow W - \gamma \cdot \underline{x} \cdot \underline{\delta}^T \qquad \underline{\delta} := \underline{y} - L(\underline{x})$

Lernziel: Linear separierbare Mengen

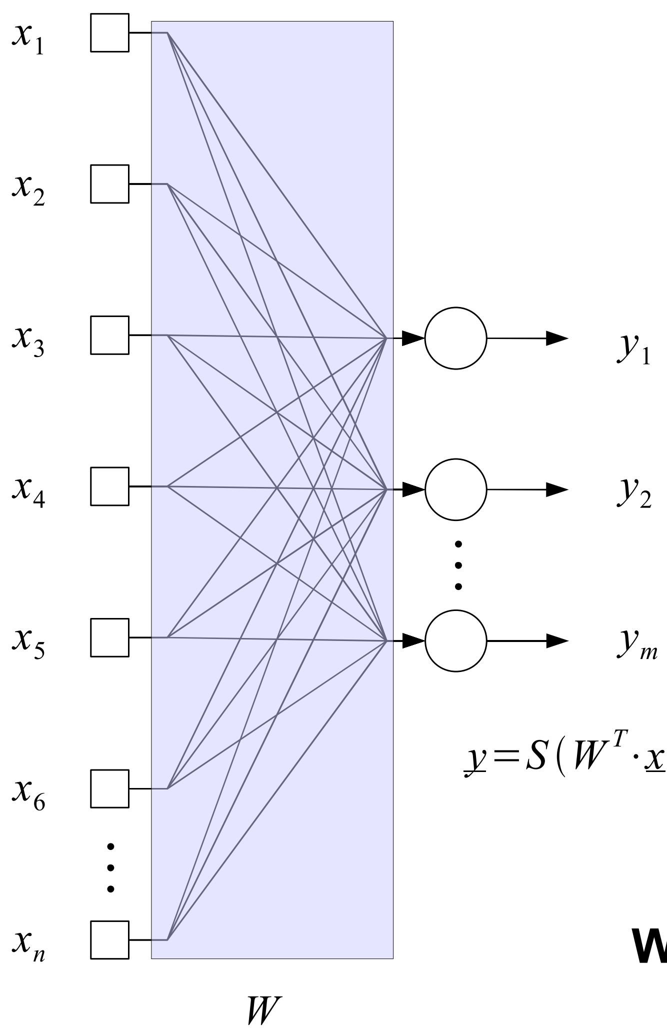
Adaline

$\underline{x} \in \mathbb{R}^n \qquad L(\underline{x}) \in \mathbb{R}^m \qquad S(z)$ beliebig

Lernregel: $W \leftarrow W - \gamma \cdot \frac{1}{\underline{x}^T \underline{x}} \cdot \underline{x} \cdot \underline{\delta}^T \qquad \delta := W^T \underline{x} - L(\underline{x})$

Lernziel: Beliebige Lineare Transformationen
$(S(z) = z)$

Netzaufbau Perzeptron/Adaline



$$\underline{y} = S(W^T \cdot \underline{x})$$

Perzeptron

$\underline{x} \in \{0,1\}^n \qquad L(\underline{x}) \in \{0,1\}^m \qquad S(z) = \begin{cases} 1, z > 0 \\ 0, z \leq 0 \end{cases}$

Lernregel: $W \leftarrow W - \gamma \cdot \underline{x} \cdot \underline{\delta}^T \qquad \underline{\delta} := \underline{y} - L(\underline{x})$

Lernziel: Linear separierbare Mengen

Adaline

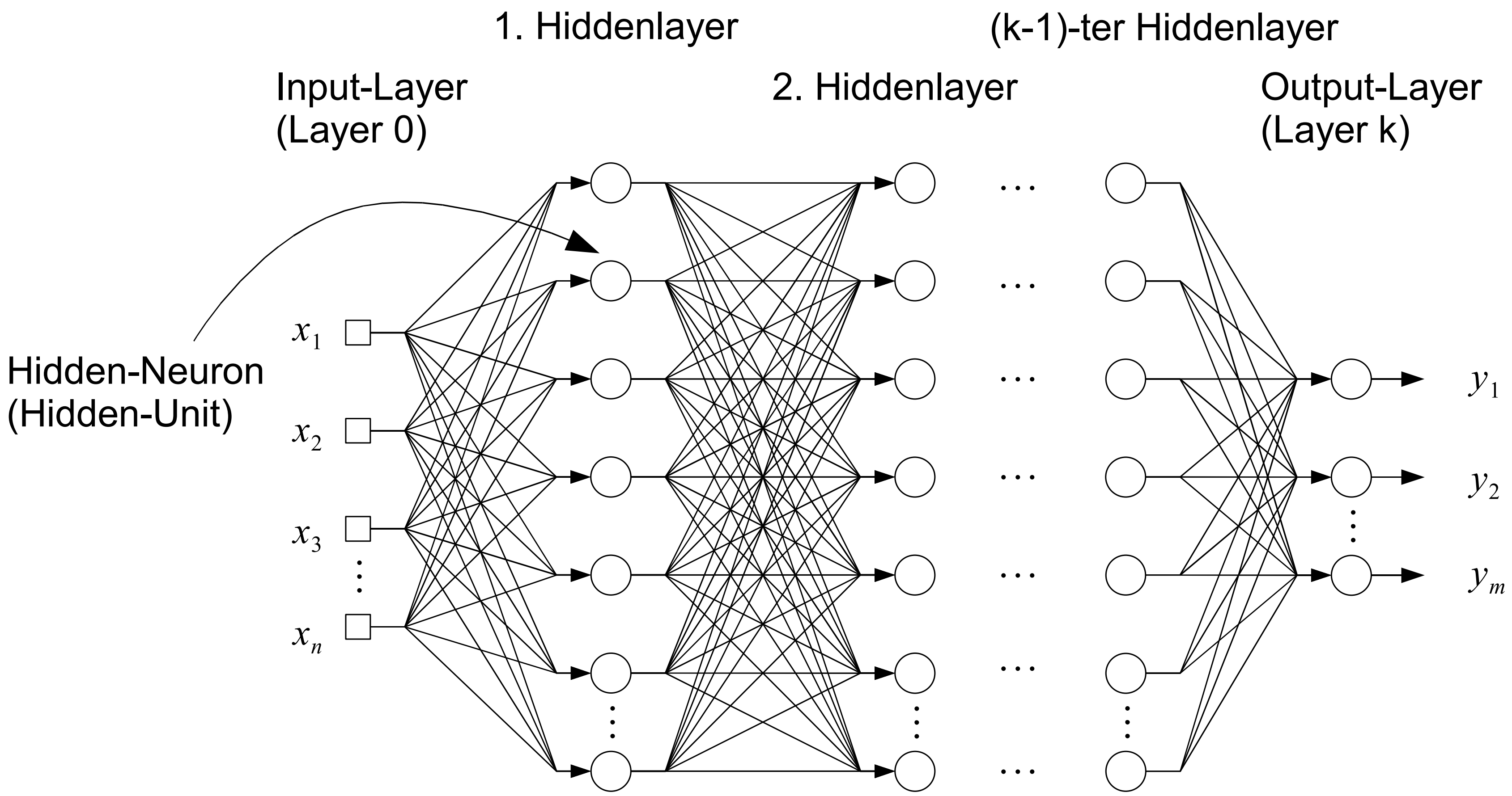
$\underline{x} \in \mathbb{R}^n \qquad L(\underline{x}) \in \mathbb{R}^m \qquad S(z)$ beliebig

Lernregel: $W \leftarrow W - \gamma \cdot \frac{1}{\underline{x}^T \underline{x}} \cdot \underline{x} \cdot \underline{\delta}^T \qquad \delta := W^T \underline{x} - L(\underline{x})$

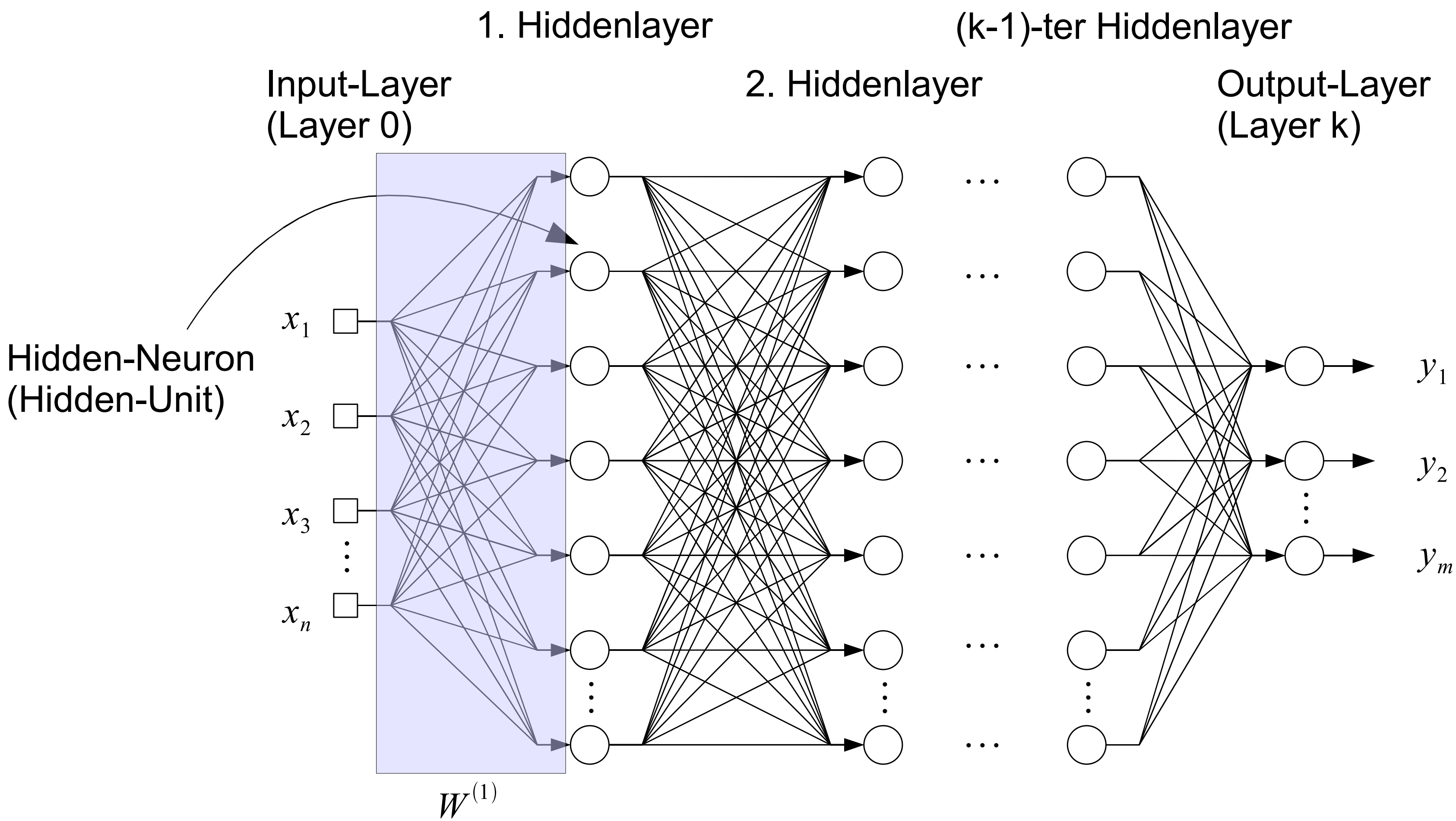
Lernziel: Beliebige Lineare Transformationen
$(S(z) = z)$

Wie können komplexere Probleme gelöst werden?

Multilayer-Perzeptrons

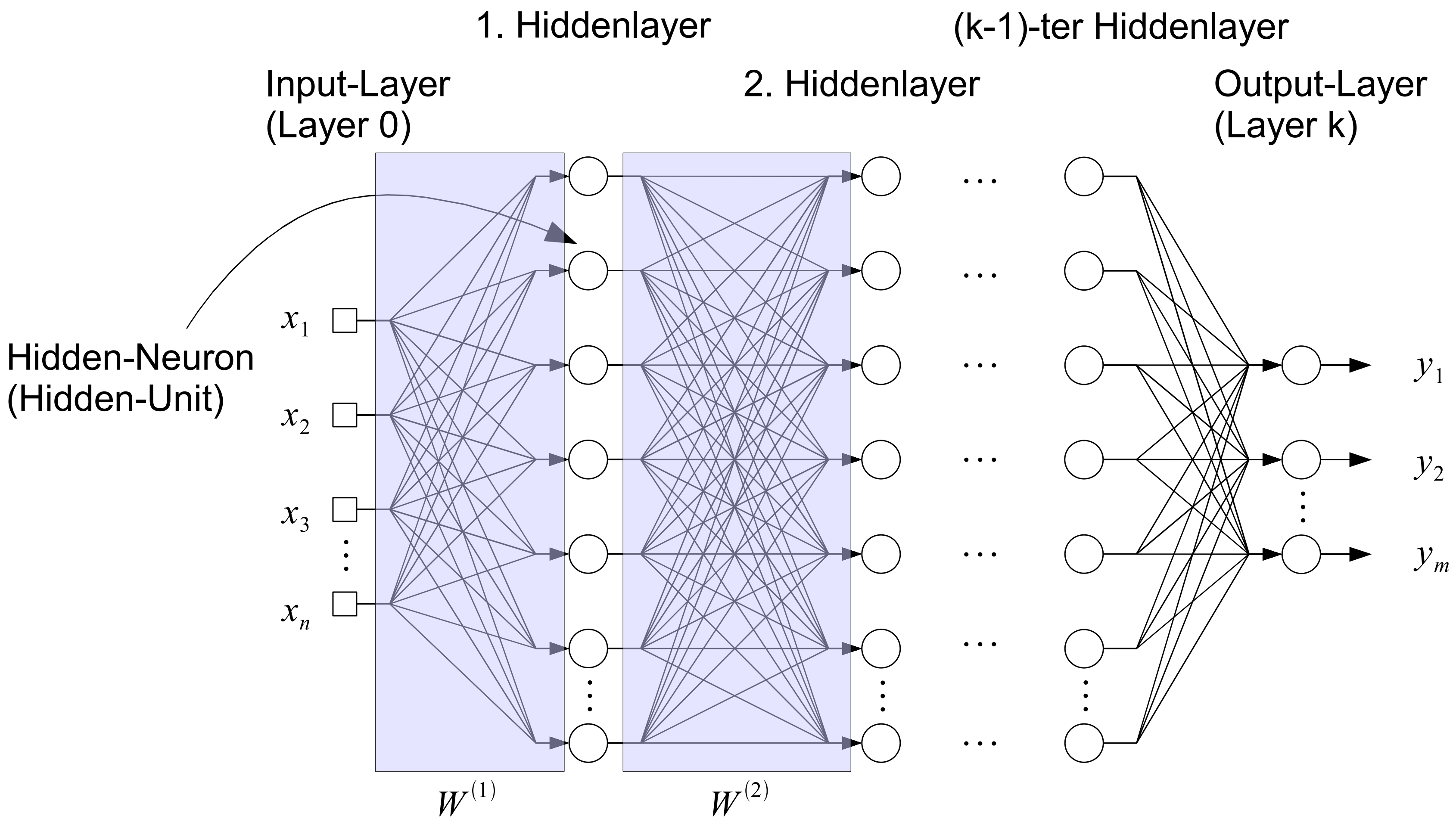


Multilayer-Perzeptrons



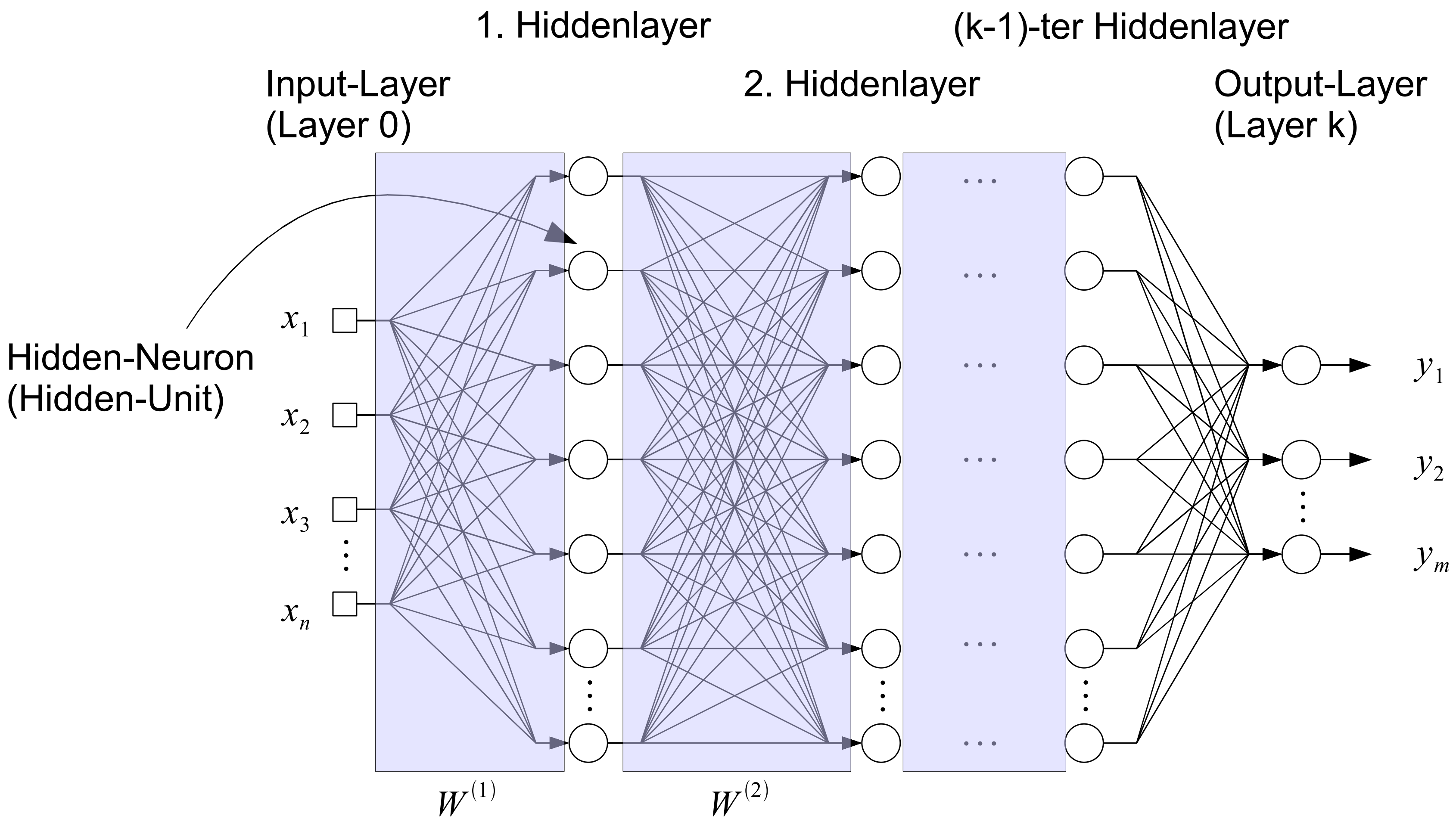
$$\underline{y}^{(1)} = S_1(W^{(1)T} \cdot \underline{x})$$

Multilayer-Perzeptrons



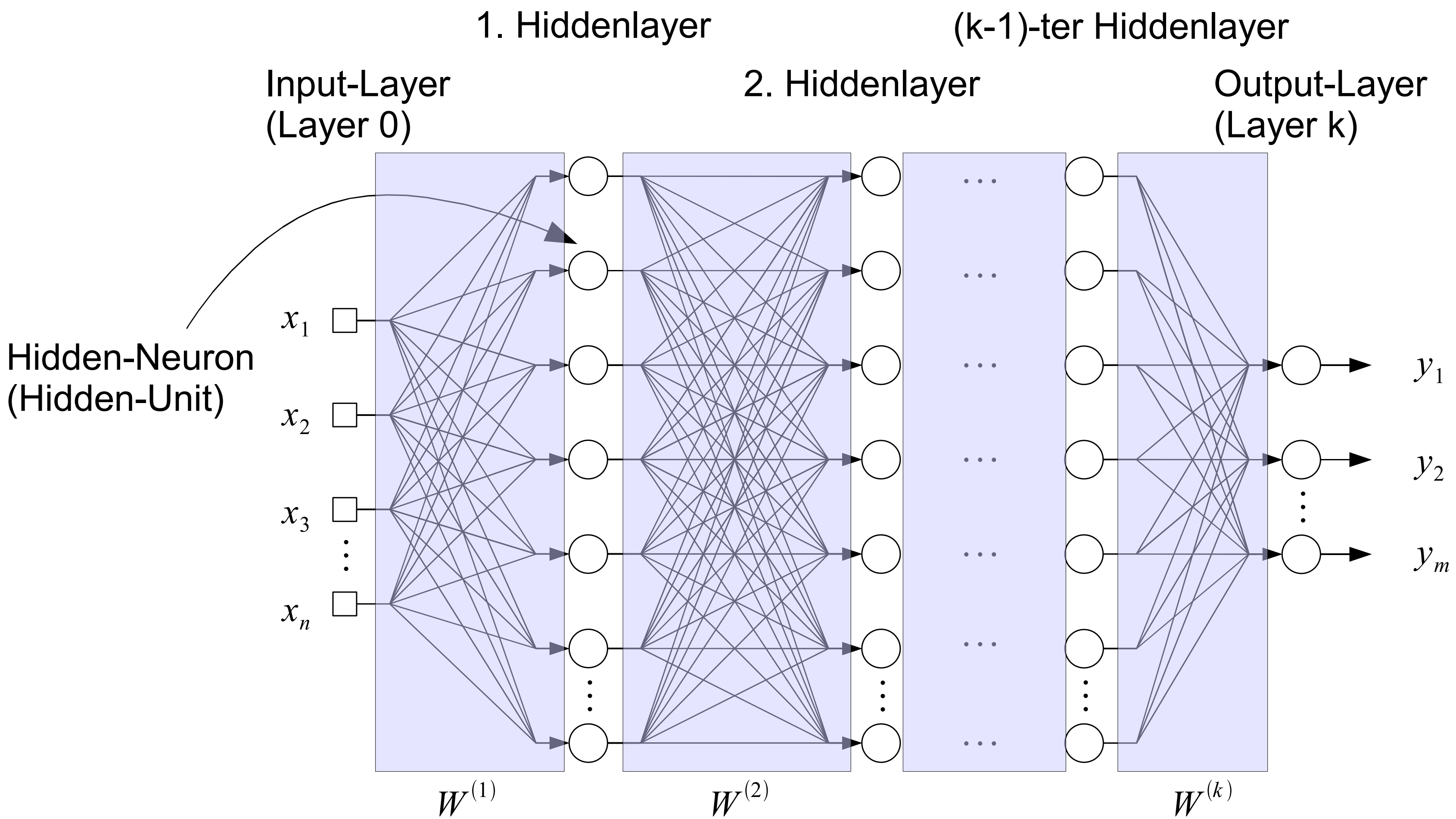
$$\underline{y}^{(1)} = S_1(W^{(1)T} \cdot \underline{x}) \qquad \underline{y}^{(2)} = S_2(W^{(2)T} \cdot \underline{y}^{(1)})$$

Multilayer-Perzeptrons



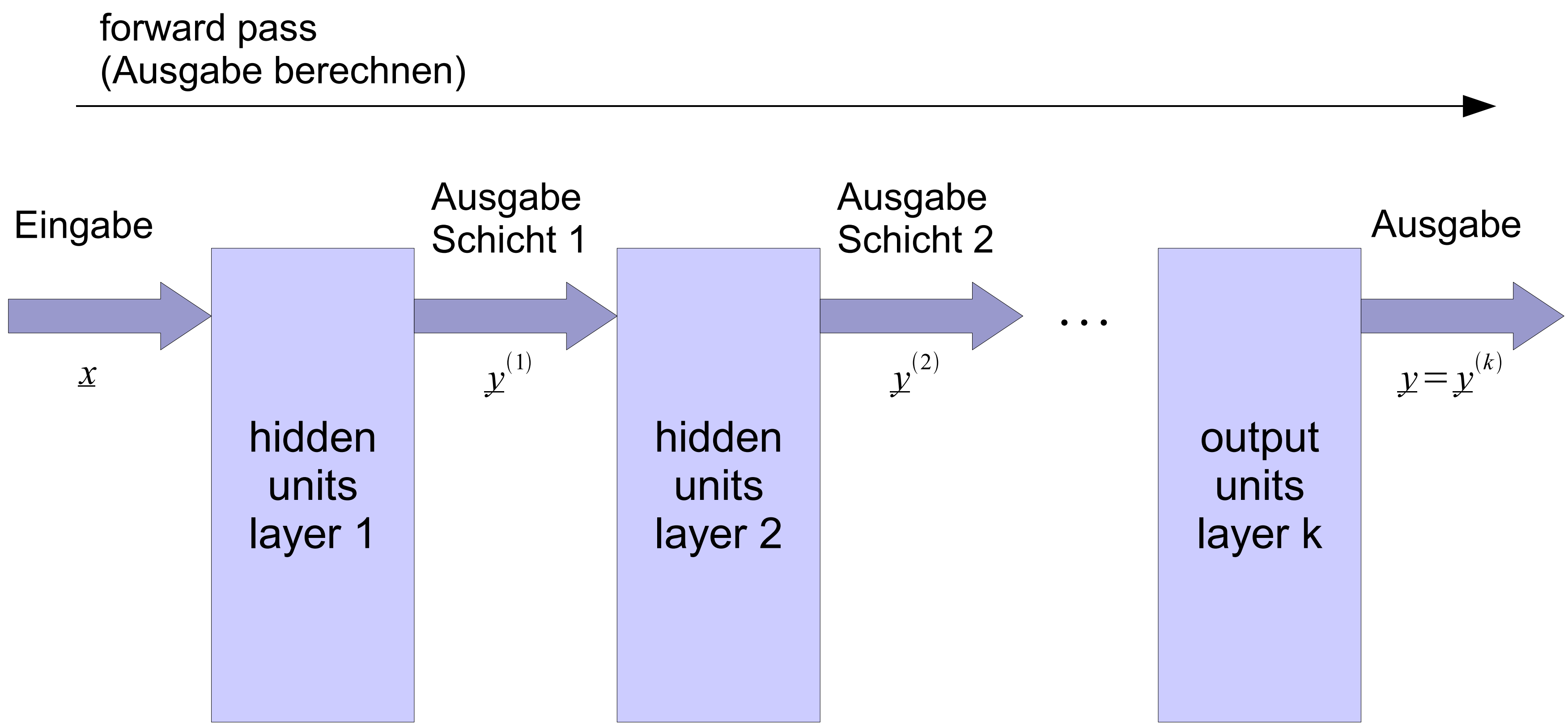
$$\underline{y}^{(1)} = S_1(W^{(1)T} \cdot \underline{x}) \qquad \underline{y}^{(2)} = S_2(W^{(2)T} \cdot \underline{y}^{(1)})$$

Multilayer-Perzeptrons

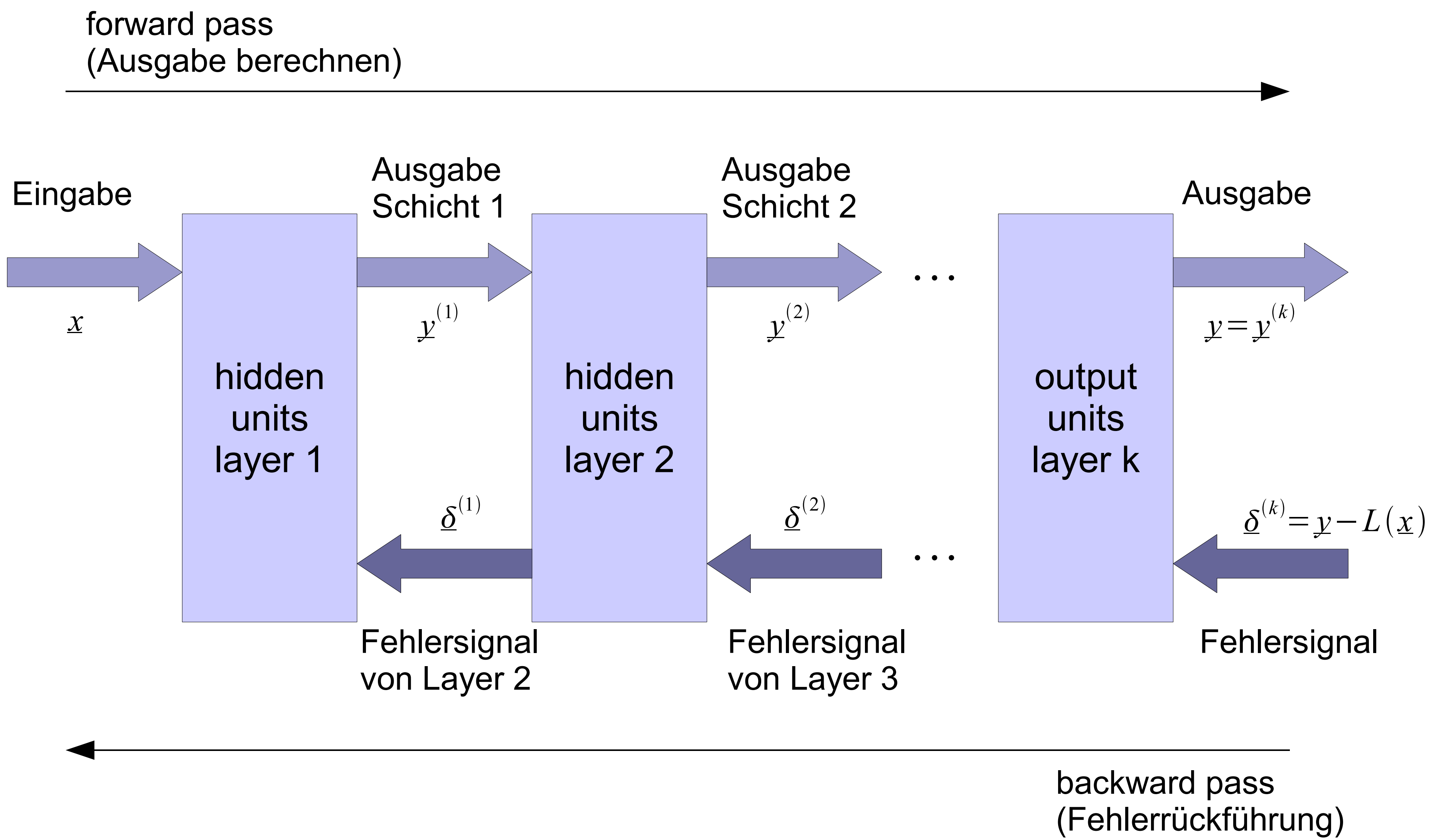


$\underline{y}^{(1)} = S_1(W^{(1)T} \cdot \underline{x})$ $\qquad$ $\underline{y}^{(2)} = S_2(W^{(2)T} \cdot \underline{y}^{(1)})$ $\qquad$ $\underline{y} = \underline{y}^{(k)} = S_k(W^{(k)T} \cdot \underline{y}^{(k-1)})$

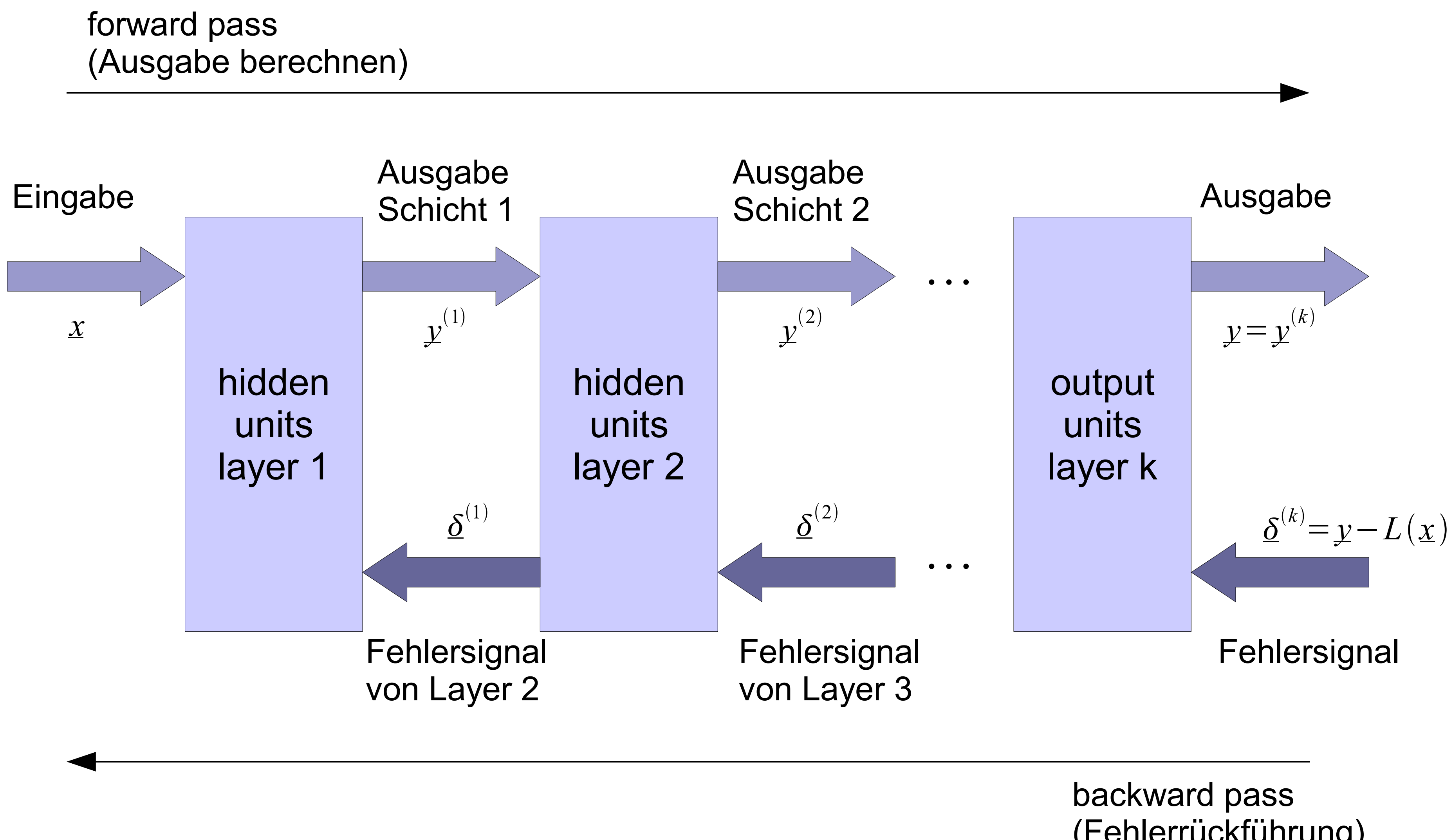
Backpropagation



Backpropagation



Backpropagation



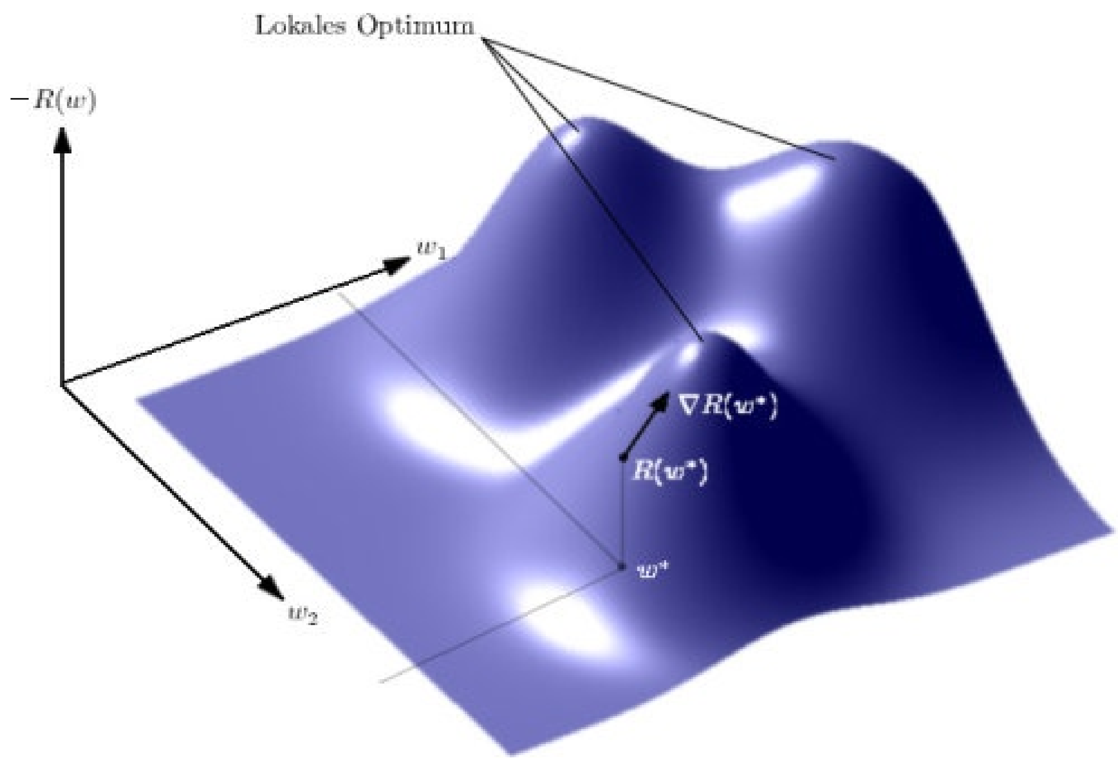
Gewichtsanpassung durch: $\underline{w}_i^{(j)} \leftarrow \underline{w}_i^{(j)} - \gamma \cdot \underline{y}^{(j-1)} \cdot \delta_i^{(j)} \cdot S'_j(z_i^{(j)})$

Interpretation als Gradientenverfahren

Ziel: Minimierung der Fehlerfunktion

$$R(W^{(1)}, W^{(2)}, \ldots, W^{(k)}) = \left\langle \left\| \underline{\delta}^{(k)} \right\|_2^2 \right\rangle_{\underline{x}}$$

(Erwartetes Längenquadrat des Fehlervektors)



Offline(Batch-) Variante

$$\underline{w}_i^{(j)} \leftarrow \underline{w}_i^{(j)} - \gamma \cdot \langle \underline{y}^{(j-1)} \cdot \delta_i^{(j)} \cdot S'_j(z_i^{(j)}) \rangle_{\underline{x}}$$

Online-Variante

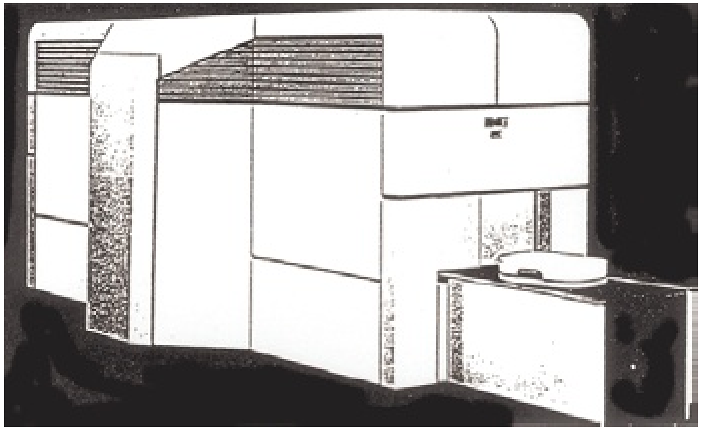
$$\underline{w}_i^{(j)} \leftarrow \underline{w}_i^{(j)} - \gamma \cdot \underline{y}^{(j-1)} \cdot \delta_i^{(j)} \cdot S'_j(z_i^{(j)})$$

Snoope (**S**ystem for **N**uclear **O**nline **O**bservation of **P**otential **E**xplosives, 1989)

Entdeckung von Plastiksprengstoff im Fluggepäck

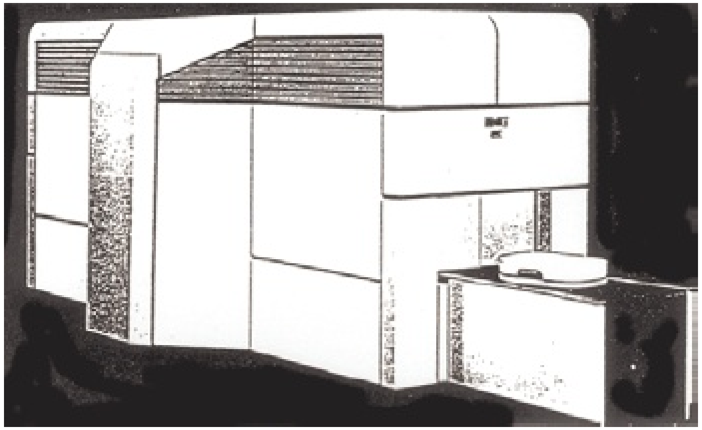
Eingabe: Gepäckstück

Ausgabe: Diagnose „gefährlich“ oder „ungefährlich“



Training und Test eines Backpropagation Netzwerks mit Gepäckstücken

Snoope (**S**ystem for **N**uclear **O**nline **O**bservation of **P**otential **E**xplosives, 1989)



Entdeckung von Plastiksprengstoff im Fluggepäck

Eingabe: Gepäckstück

Ausgabe: Diagnose „gefährlich“ oder „ungefährlich“

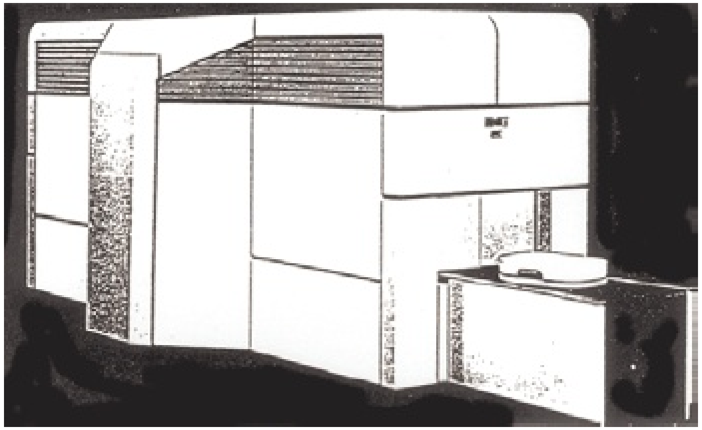
Training und Test eines Backpropagation Netzwerks mit Gepäckstücken

Nettalk (Sejnowsky, Rosenberg, 1986)

Umwandlung von Text in Sprache

16 CPU-Stunden Backpropagation-Training für 98% Genauigkeit

Snoope (**S**ystem for **N**uclear **O**nline **O**bservation of **P**otential **E**xplosives, 1989)



Entdeckung von Plastiksprengstoff im Fluggepäck

Eingabe: Gepäckstück

Ausgabe: Diagnose „gefährlich" oder „ungefährlich"

Training und Test eines Backpropagation Netzwerks mit Gepäckstücken

Nettalk (Sejnowsky, Rosenberg, 1986)

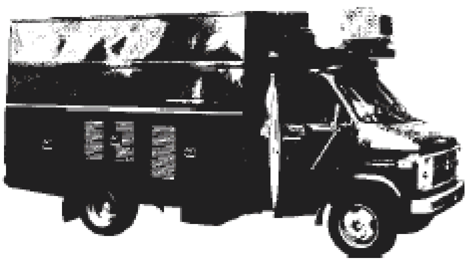
Umwandlung von Text in Sprache

16 CPU-Stunden Backpropagation-Training für 98% Genauigkeit

Älteres System: DECTalk

Ausgabe Text->Sprache der Firma Digital Eq. (DEC)

Aufwand 20 PJ für 95% Genauigkeit



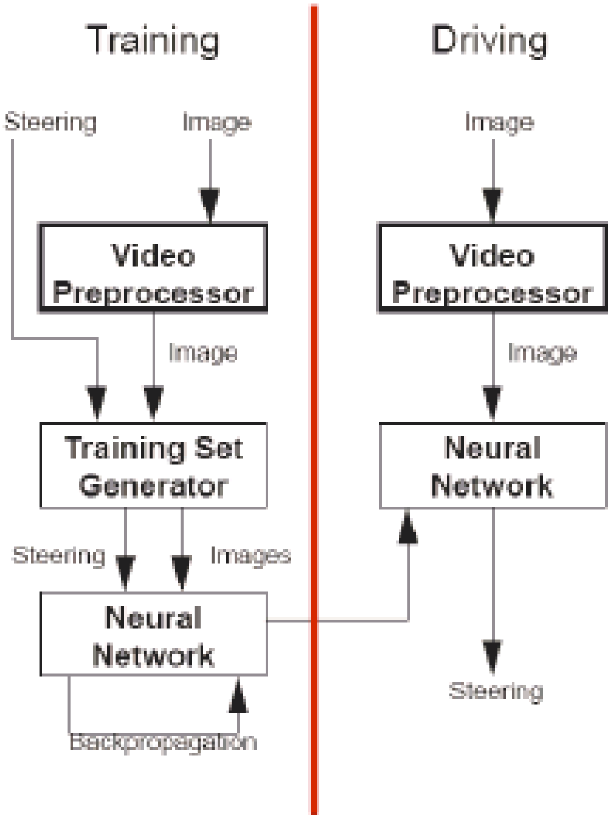
ALVINN (**A**utonomous **L**and **V**ehicle **i**n a **N**eural **N**etwork)

Team: Carnegie-Mellon University, Pittsburgh USA

Methode: 2-Schicht Backpropagation

Training auf stationärem Supercomputer (100 MFlops)
mit Aufzeichnungen von 1200 simulierten Straßenbildern
(40x präsentiert)

Resultat: Automatisches Fahren auf Uni-Gelände
mit ca. 5 km/h, ausgelegt bis max. ca. 100 km/h



Was können Multilayer-Perzeptrons?

Folgendes ist bekannt:

Was können Multilayer-Perzeptrons?

Folgendes ist bekannt:

→ Jede boolesche Funktion kann exakt durch ein Multilayer-Perzeptron mit 1 Hiddenlayer dargestellt werden.

Was können Multilayer-Perzeptrons?

Folgendes ist bekannt:

→ Jede boolesche Funktion kann exakt durch ein Multilayer-Perzeptron mit 1 Hiddenlayer dargestellt werden.

→ Jede beschränkte kontinuierliche Funktion kann beliebig genau durch ein MLP mit 1 Hiddenlayer und linearer Ausgabefunktionen im Ausgabelayer approximiert werden.

Was können Multilayer-Perzeptrons?

Folgendes ist bekannt:

→ Jede boolesche Funktion kann exakt durch ein Multilayer-Perzeptron mit 1 Hiddenlayer dargestellt werden.

→ Jede beschränkte kontinuierliche Funktion kann beliebig genau durch ein MLP mit 1 Hiddenlayer und linearer Ausgabefunktionen im Ausgabelayer approximiert werden.

→ Jede beliebige Funktion kann beliebig genau durch ein MLP mit 2 Hiddenlayern und linearer Ausgabefunktion im Ausgabelayer approximiert werden.

Was können Multilayer-Perzeptrons?

Folgendes ist bekannt:

→ Jede boolesche Funktion kann exakt durch ein Multilayer-Perzeptron mit 1 Hiddenlayer dargestellt werden.

→ Jede beschränkte kontinuierliche Funktion kann beliebig genau durch ein MLP mit 1 Hiddenlayer und linearer Ausgabefunktionen im Ausgabelayer approximiert werden.

→ Jede beliebige Funktion kann beliebig genau durch ein MLP mit 2 Hiddenlayern und linearer Ausgabefunktion im Ausgabelayer approximiert werden.

→ Fazit: 1 Hiddenlayer in den meisten Fällen ausreichend