# Informatik IV - Tutorium XVI & XVII

## Tut Nr. 5 – Lernen von NN & Hopfield-Netze

David Münch

Universität Karlsruhe (TH)
Fakultät für Informatik
IBDS Prautzsch

Dank an Yusuf für die Zusammenarbeit.

5. Juni 2009

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

# Inhaltsverzeichnis

# Inhaltsverzeichnis

# Inhaltsverzeichnis

# Inhaltsverzeichnis

# Organisatorisches

Email: muenchdavid@gmail.com

https://www.stud.uni-karlsruhe.de/~uhbro/

Tutorium 16: Freitags 8:00 Uhr - Raum -107
Tutorium 17: Freitags 9:45 Uhr - Raum -119

# Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009
- 02.07.2009
- 23.07.2009

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

# Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009
- 02.07.2009
- 23.07.2009

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

# Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009
- 02.07.2009
- 23.07.2009

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

Abgabe in Zweiergruppe erlaubt und ausdrücklich erwünscht!

# Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\sqrt{}$ $\varnothing 31$ Punkte
- 09.06.2009
- 02.07.2009
- 23.07.2009

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

Abgabe in Zweiergruppe erlaubt und ausdrücklich erwünscht!

Bitte folgendes Deckblatt verwenden:
`http://www.stud.uni-karlsruhe.de/~unbdh/deckblatt/index.php?course=8`

# Literatur

Boehm, Prautzsch: Numerical Methods. AK Peters 1993. ISBN 3-528-06350-5
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&fbt=7319953&nd=3204657

Ash: Information Theory. Dover 1990. ISBN 0-486-66521-6
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&nd=9866904

Goos: Vorlesungen über Informatik. Bd. 4, Springer 1998. ISBN 3-540-60650-5
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&fbt=9316367&nd=6568301

# Was wollen wir heute erreichen?

# Was wollen wir heute erreichen?

- Hausaufgabenblatt 5 besprechen

# Was wollen wir heute erreichen?

- Hausaufgabenblatt 5 besprechen
- Hopfiled-Netze

## Hausaufgabe 11

1. Zeigen Sie, dass jede Funktion $f : \{0,1\}^m \to \{0,1\}$ als boolscher Ausdruck mit $m$ Variablen darstellbar ist.

# Hausaufgabe 11

2. Geben Sie für die in der Tabelle aufgeführten Funktionen $f_i : \{0,1\}^3 \to \{0,1\}$, $i = 1,\ldots,5$, je einen boolschen Ausdruck an, der aus den Operatoren $\vee, \wedge, \neg, \Rightarrow, \Leftrightarrow$ aufgebaut ist.

| $x$ | $y$ | $z$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

## Hausaufgabe 11

3. Modellieren Sie mit 0/1-Neuronen je ein Netz für $f_2$ und $f_5$. Der Wert 1 soll dabei TRUE bedeuten, der Wert 0 den Wahrheitswert FALSE.

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation
**Gesucht**: Gewichte und Schwellwerte, dass das NN die Trainingsdaten richtig klassifiziert.

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation
**Gesucht**: Gewichte und Schwellwerte, dass das NN die Trainingsdaten richtig klassifiziert.
**Problem**: Die gewünschte Ausgabe der versteckten Neronen ist unbekannt.

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation
**Gesucht**: Gewichte und Schwellwerte, dass das NN die Trainingsdaten richtig klassifiziert.
**Problem**: Die gewünschte Ausgabe der versteckten Neronen ist unbekannt.
**Lösung**: Error-Backpropagation-Algorithmus

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation
**Gesucht**: Gewichte und Schwellwerte, dass das NN die Trainingsdaten richtig klassifiziert.
**Problem**: Die gewünschte Ausgabe der versteckten Neronen ist unbekannt.
**Lösung**: Error-Backpropagation-Algorithmus
**Idee**: Definiere eine von den Gewichten abhängige Fehlerfunktion und minimiere deren Wert mittels Gradientenabstieg.

# Lernen für mehrschichtige Perzeptronen

**Gegeben**: Menge von Trainingsdaten und zugehöriger Klassifikation
**Gesucht**: Gewichte und Schwellwerte, dass das NN die Trainingsdaten richtig klassifiziert.
**Problem**: Die gewünschte Ausgabe der versteckten Neronen ist unbekannt.
**Lösung**: Error-Backpropagation-Algorithmus
**Idee**: Definiere eine von den Gewichten abhängige Fehlerfunktion und minimiere deren Wert mittels Gradientenabstieg.
**Algorithmus**:
1. Forward-Pass (Schichtweises Durchreichen der Eingabe)
2. Bestimmen des Fehlers
3. Backpropagation (Rückrechnen des Fehlers)
4. Gewichte anpassen

**Algorithm 1** MUSTERLERNEN DURCH RÜCKKOPPLUNG

**input** : Perzeptron mit $m$ Schichten, Zustandsmenge $Q$, $n_0$ Eingabeneuronen, $n_m$ Ausgabeneuronen, Erregerfunktion $h(x)$, Gewichte $w_{ij}$, Schwellwerte $\sigma_k$, Lernmuster $(\vec{e}, \vec{a})$

**output**: Optimierte Gewichte $w_{ij}$

1. Vorwärtsrechnen

**for** $k = 1...m$ **do**

| berechne $\vec{q}_k, \vec{p}_k$ und $J_k = J(\vec{p}_k)$

**end**

2. Rückwärtsrechnen

$\vec{d}_{m+1} := (\vec{q}_m - \vec{a})$ und $W_{m+1} := E$

**for** *k=m..1* **do**

| berechne $\vec{d}_k^t = \vec{d}_{k+1}^t W_{k+1} J_k$ und $\partial E / \partial W_k$

**end**

Gib $W$ aus.

# Hausaufgabe 12

Wir benutzen die Bezeichnungen aus Kapitel 5.8 der Vorlesung.

Wir betrachten ein vorwärtsgerichtetes Perzeptron mit $m$ Schichten, Zustandsmenge $Q = \mathbb{R}$, $n_0$ Eingabeneuronen, $n_m$ Ausgabeneuronen, linearer Erregungsfunktion $h(x) = x$, Gewichten $\omega_{ij}^k$, Schwellwerten $\sigma_k$ und Lernmuster $(\mathbf{e}, \mathbf{a})$, wobei

$$n_0 = n_2 = 2, n_1 = 1, m = 2, \mathbf{e} = [1,\ 2]^t, \mathbf{a} = [2,\ 3]^t, \sigma_k = \mathbf{0}, \omega_{ij}^k = 1.$$

1. Führen Sie einen Schritt des Lernens mit Rückkopplung und mit Lernrate $\eta = 0{,}1$ durch.

2. Führen Sie einen Schritt des Quickprop-Lernalgorithmus mit den Anfangsgewichten $\omega_{ij}^k = 1$ für den Zeitpunkt $t = 0$ und den gelieferten Gewichten aus Teilaufgabe 1 für den Zeitpunkt $t = 1$ durch. Geben Sie die Gewichte und den Wert der Fehlerfunktion $E$ für den Zeitpunkt $t = 2$ an.

## Hausaufgabe 12

3. Wir benutzen in dieser Teilaufgabe die Bezeichnungen aus Kapitel 5.10 der Vorlesung. Lösen Sie für die Zeitpunkte $t = 1, 2$ die Gewichte $\omega_{ij}^{t=1}$, $\omega_{ij}^{t=2}$ und den Wert der Fehlerfunktion $E$ mithilfe von RPROP-Lernalgorithmus mit $h_0 = 0{,}1$, $h_{\max} = 50$, $\eta_- = 0{,}5$, $\eta_+ = 1{,}2$ und den Anfangsgewichten $\omega_{ij}^{t=0} = 1$ für den Zeitpunkt $t = 0$.

Bei der Berechnung können alle Zwischenergebnisse auf 4 Stellen nach dem Komma gerundet werden.

# Hopfield-Netze

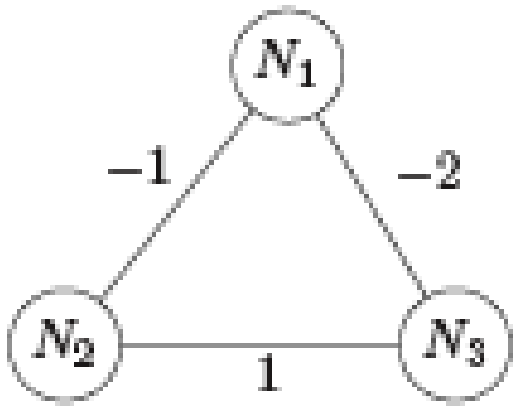
Ein Hopfield-Netz ist ein rekurrentes Neuronales Netz mit den folgenden Eigenschaften:

- Es besteht aus $n$ Neuronen die paarweise miteinander verbunden sind
- Es gibt keine negativen Schlingen
- Es gibt keine Eingabe- und Ausgabeneuronen
- Für die Gewichte gilt $w_{ij} = w_{ji}$

Ein Hopfield-Netz ist ein rekurrentes Neuronales Netz mit den folgenden Eigenschaften:

- Es besteht aus $n$ Neuronen die paarweise miteinander verbunden sind
- Es gibt keine negativen Schlingen
- Es gibt keine Eingabe- und Ausgabeneuronen
- Für die Gewichte gilt $w_{ij} = w_{ji}$

Beispiel:



Die Gewichte fassen wir als Matrix $W \in \mathbb{R}^{n \times n}$ und die Schwellwerte als Vektor $\boldsymbol{\sigma} \in \mathbb{R}^n$ auf. $W$ ist eine symmetrische Matrix mit Nullen auf der Hauptdiagonalen.
Die Erregungsfunktion ist wie beim Perzeptron:

$$p_{ij} := \sum_{j=1}^{n} w_{ij} q_j - \sigma_i = W_i \boldsymbol{q} - \sigma_i$$

Mit $\boldsymbol{p} = (p_1, ..., p_n)^t$ gilt dann: $\boldsymbol{p} = W\boldsymbol{q} - \boldsymbol{\sigma}$.

Die Gewichte fassen wir als Matrix $W \in \mathbb{R}^{n \times n}$ und die Schwellwerte als Vektor $\boldsymbol{\sigma} \in \mathbb{R}^n$ auf. $W$ ist eine symmetrische Matrix mit Nullen auf der Hauptdiagonalen.
Die Erregungsfunktion ist wie beim Perzeptron:

$$p_{ij} := \sum_{j=1}^{n} w_{ij} q_j - \sigma_i = W_i \boldsymbol{q} - \sigma_i$$

Mit $\boldsymbol{p} = (p_1, ..., p_n)^t$ gilt dann: $\boldsymbol{p} = W\boldsymbol{q} - \boldsymbol{\sigma}$.

Die Aktivierungsfunktion ist $q_i^+ = \begin{cases} 1 & \text{falls } p_i > 0 \\ -1 & \text{falls } p_i < 0 \\ q_i & \text{falls } p_i = 0 \end{cases}$

Das Netz ist stabil falls $q^+ = q$.

Varianten:

- synchron: Alle Neuronen schalten gleichzeitig
- asynchron: In einem Zeitpunkt schaltet ein zufälliges Neuron

Die Eingabe ist der Startzustand des Netzes.

Varianten:

- synchron: Alle Neuronen schalten gleichzeitig
- asynchron: In einem Zeitpunkt schaltet ein zufälliges Neuron

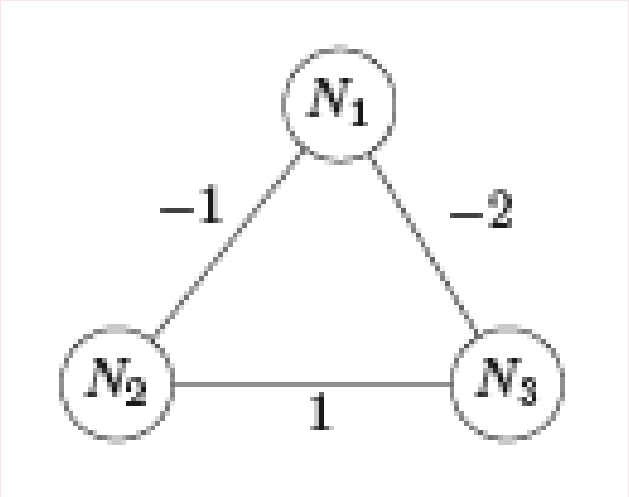
Die Eingabe ist der Startzustand des Netzes.

**Satz**

Ein Hopfield-Netz erreicht für jede Eingabe im asynchronen Betrieb nach endlich vielen Schritten einen stabilen Zustand.

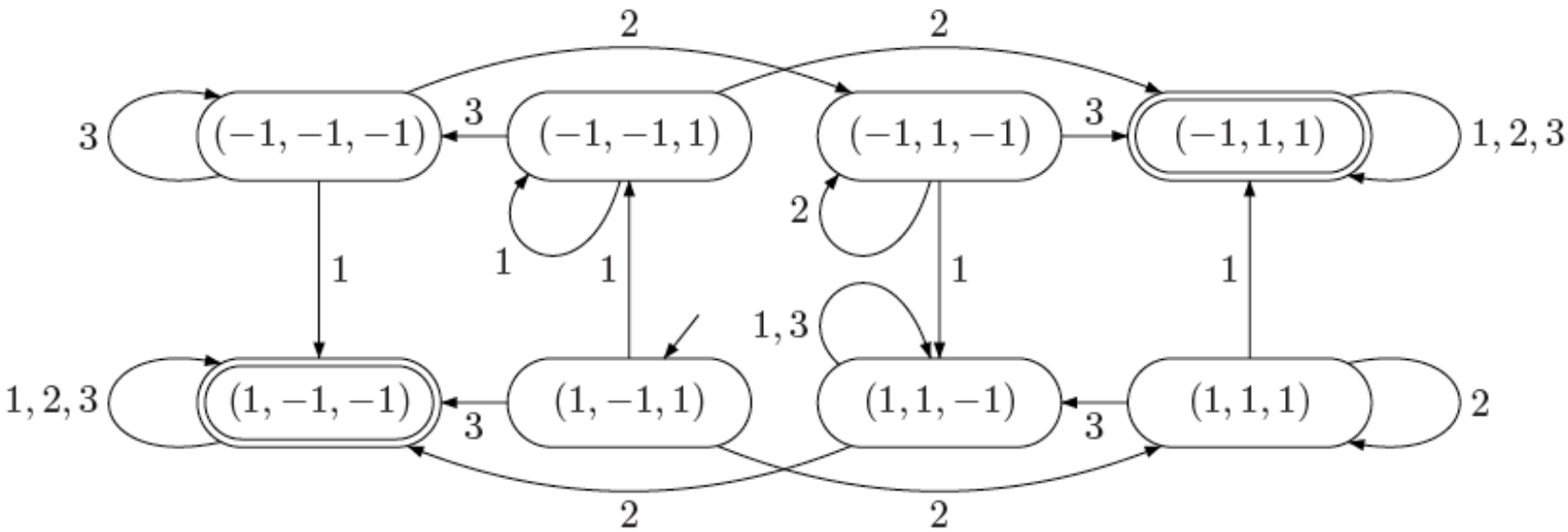
## Aufgabe

Betrachte das hier abgebildete sehr einfache Hopfield-Netz




mit $\boldsymbol{\sigma} := (0, -1, 2)^t$.
Bestimme für dieses Netz ausgehend vom Anfangszustand $(q_1, q_2, q_3)^t = (1, -1, 1)^t$ den, bzw. die Endzustände, wenn das Netz asynchron arbeitet.
Hinweis: benutze einen Zustandsgraphen als Hilfe.

# Lösung



# Quellen

Pajor - Informatik 4 Tutorium SS2007
Prautzsch - Skript Informatik 4 SS2008
Goos, Vorlesungen über Informatik Band 4

# Reflexion

Was haben wir heute gelernt?

# Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 5 besprochen

# Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 5 besprochen
- Lernverfahren verstanden

# Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 5 besprochen
- Lernverfahren verstanden
- Hopfield-Netze

# Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 5 besprochen
- Lernverfahren verstanden
- Hopfield-Netze
- Bis einschliesslich Kapitel 6 abgeschlossen

Noch Fragen?

# Vorschau

# Vorschau

- Informationsmass

# Vorschau

- Informationsmass
- Codes

# Bis zum nächsten Mal