

# Effiziente Näherungsverfahren 2

D. Rucker S. Major

Hochschule Zittau-Görlitz

21. Juni 2009

- ▶ DNA Computing
- ▶ evolutionäre Algorithmen
- ▶ Neuronale Netze

## DNA Computing

## erste Gedanken

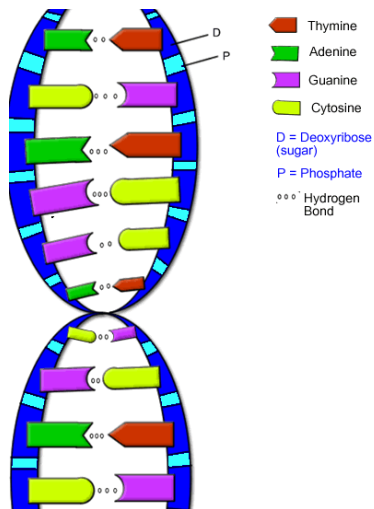
- ▶ NP - Probleme nur in exponentieller Zeit lösbar
- ▶ Idee, Berechnungen auf molekularer Ebene durchzuführen
- ▶ Kodierung der Daten in DNA
- ▶ auffassen des Reagenzglases als einen Computer mit zahlreichen parallel arbeitenden Prozessoren

## Grundidee

- ▶ Datenspeicherung durch DNA
- ▶ Operationen, welche auf alle DNA Stränge wirken
- ▶ man nutzt die Werkzeuge, welche durch Evolution entstanden sind
- ▶ Übertragung der Probleme auf DNA (Codierung)
- ▶ mathematische Berechnungen auf Basis von Interaktionen zwischen organischen Molekülen

## biologische Grundlagen

- ▶ DNA trägt die Erbinformation von Individuen
- ▶ Doppelhelixstruktur
- ▶ Zucker - Phosphat - Rückgrat(backbone)
- ▶ von diesem backbone stehen organische Basen ab (A,T,G,C)
- ▶ es stehen sich immer zwei Basenpaare gegenüber(A und T bzw. G und C)
- ▶ = komplementäre Basen



## Operationen

- ▶ Synthese: Anfertigung von Strängen
- ▶ Vervielfältigung
- ▶ schneiden durch Restriktionsenzyme
- ▶ Ligieren: verbinden zweier DNA - Ketten durch DNA - Ligase
- ▶ Sequenzierung: Bestimmung einer bestimmten Sequenz
- ▶ Rekombination: gezieltes einsetzen von DNA - Stücken an definierten Stellen in ein anderes DNA - Molekül

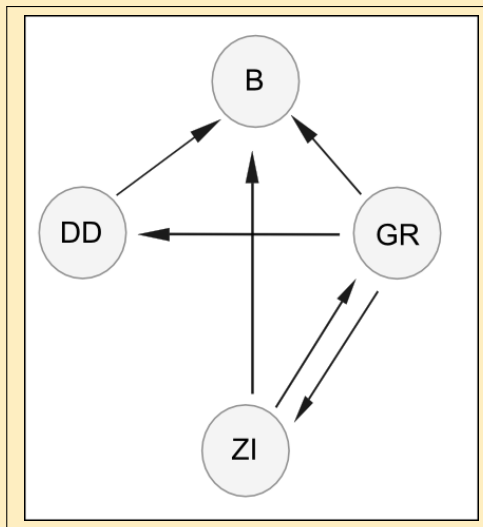
## Hamilton Pfad Problem

- ▶ beschäftigt sich mit dem finden eines Weges durch einen Graphen
- ▶ jeder Knoten genau einmal durchschritten bzw. es müssen alle Knoten durchgangen werden
- ▶ Adleman hat 1994 bewiesen, dass sich dieses Problem mit Hilfe von DNA Molekülen lösen lässt
- ▶ zählt zu den NP vollständigen Problemen
- ▶ Brute Force Methode
- ▶ Lösungen werden in einer chemischen Lösung in DNA codiert erzeugt
- ▶ zwei Phasen
  1. Initialisierungsphase: Sequenzen werden erzeugt
  2. Berechnungsphase: schlechte Kandidaten werden aussortiert



# Anwendungsbeispiel

## Graph



# Anwendungsbeispiel

## Schritt 1: DNA Synthese

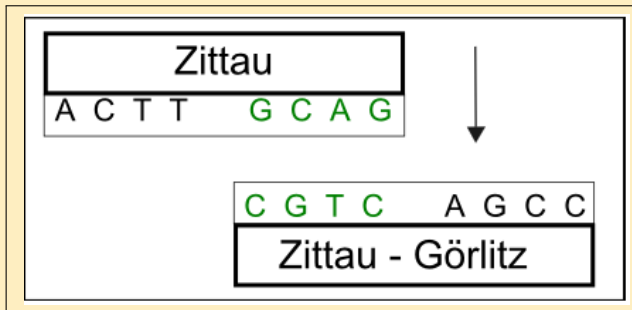
- ▶ jede Stadt wird durch eine Abfolge von 8 Basen codiert
- ▶ Städteverbindungen werden durch die komplementäre Abfolge der vier letzten Basen des Ausgangspunktes und der ersten vier Basen des Zielortes gebildet

## Schritt 2: Nasschemie im Reagenzglas

- ▶ durch Ligase werden DNA Stränge verknüpft
- ▶ Reaktionsschritte werden wiederholt
- ▶ Vielzahl von Reiseverbindungen im Reagenzglas
- ▶ um alle möglichen Lösungskandidaten zu erzeugen, müssen hinreichende Mengen von DNA Molekülen zusammengegeben werden

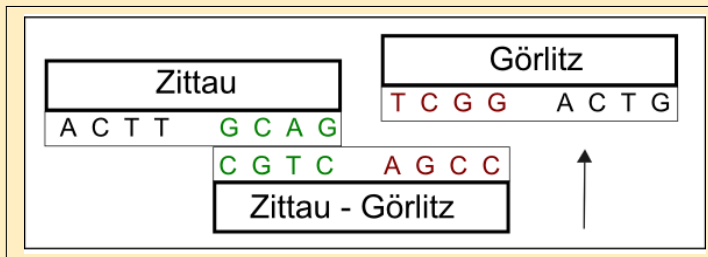
# Anwendungsbeispiel

## Schritt 2: Nasschemie



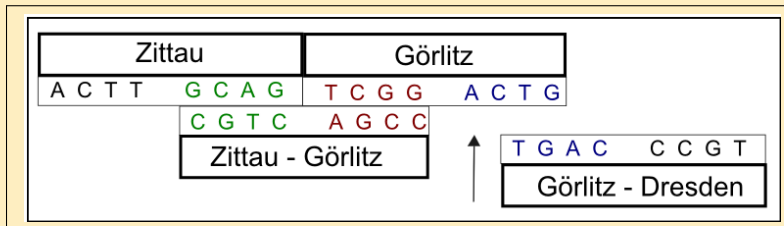
# Anwendungsbeispiel

## Schritt 2: Nasschemie



# Anwendungsbeispiel

## Schritt 2: Nasschemie



# Anwendungsbeispiel

## Schritt 3

- ▶ Stränge, welche den richtigen Start - und Zielort haben, werden vervielfältigt
- ▶ durch eine Gelelektrophorese werden diese Städte herausgefiltert, welche die richtig Stadtanzahl besitzen
- ▶ Affinitätsprüfung um festzustellen, ob jede Stadt tatsächlich nur einmal besucht wurde

## Schritt 4

- ▶ Sequenzanalyse des durch Aufarbeitung isolierten DNA Einzelstranges
- ▶ Bsp.: Zittau - Görlitz - Dresden - Berlin

## Algorithmus zum HPP

- ▶ Eingabe: gerichteter Graph  $G$  mit  $n$  Knoten und dem Startknoten  $V_{in}$  und Endknoten  $V_{out}$
- 1. Erzeuge Instanzen aller möglicher Pfade
- 2. Verwirf Pfade, die nicht mit  $V_{in}$  beginnen und/oder nicht in  $V_{out}$  enden
- 3. Verwirf Pfade, die nicht exakt  $n$  Knoten haben
- 4. Verwirf Pfade, die nicht jeden Knoten mindestens einmal enthalten
- ▶ Ausgabe: verbliebene Pfade sind Hamilton Pfade

# Vor - und Nachteile

## Vorteile

- ▶ sehr hohe Rechengeschwindigkeit
- ▶ Energieeffizienz
- ▶ enorm hohe Speicherkapazität
- ▶ Haltbarkeit der Daten wesentlich höher als bei digitalen Speichermedien

## Nachteile

- ▶ langwierige Aufarbeitung zur Ermittlung der Rechenergebnisse
- ▶ die Menge der benötigten DNA wächst exponentiell



## Visionen

- ▶ noch recht ungewiss
- ▶ man strebt an, die Operationen, welche mit der DNA durchführbar sind, als Standardoperationen zur Verfügung zu stellen
- ▶ Praktisch ist die Idee des DNA - Computing bis heute noch nicht anwendbar

evolutionäre Algorithmen

## erste Gedanken

- ▶ zur Optimierung von Problemen aus Forschung, Industrie und Wirtschaft
- ▶ orientieren sich an Vorbildern der natürlichen Evolution
- ▶ inspiriert durch die Biologie
- ▶ dadurch anderer Weg zur Beschreibung von Optimierungsproblemen

## Gen

- ▶ Abschnitt auf der DNA
- ▶ speichert Grundinformationen über Herstellung von RNA

## Chromosomen

- ▶ enthalten Gene und damit Erbinformationen

## Individuen

- ▶ räumlich und qualitativ einmaliges Einzelwesen

## Kreuzung

- ▶ gezielte Auswahl und Fortpflanzung zweier Individuen

## Selektion

- ▶ Auswahl der überlebenstüchtigeren Individuen

## Mutation

- ▶ Veränderung des Erbgutes

## Population

- ▶ Gruppe von Individuen einer Art
- ▶ bilden Fortpflanzungsgemeinschaft

## Traveling Salesman

- ▶ Handelsreisender soll viele Städte besuchen um seine Ware zu verkaufen
- ▶ er hat wenig Zeit und benötigt somit den kürzesten Weg durch alle Städte ohne eine zweimal zu besuchen
- ▶ Aufgabe: möglichst kurze Tour durch gegebene Städte

## Verbindung des Themas mit der Biologie

- ▶ die einzelnen Städte werden durch Gene verkörpert
- ▶ ein Individuum beschreibt eine Rundreise
- ▶ Individuen besitzen eine Fitness, welche sich aus der Gesamtlänge ergibt
- ▶ Population: Gesamtheit aller Rundreisen
- ▶ Kreuzung: wenn neuer Weg durch Kombination zweier Elternindividuen
- ▶ Selektion bezeichnet die Auswahl der Individuen zur Kreuzung



## Ablaufplan

- ▶ Erzeugung einer Anfangspopulation
- ▶ Bewertung der Individuen
- ▶ Erzeugen neuer Individuen durch Kreuzung und Mutation
- ▶ Erneute Bewertung
- ▶ Diese Vorgänge werden so oft wiederholt bis n-Generationen erzeugt wurden

## Neuronale Netze

## erste Gedanken

- ▶ Kategorien von Problemen, welche sich nicht in einen Algorithmus fassen lassen
- ▶ Unterschied zwischen Computer und Mensch ist der Aspekt der Lernfähigkeit
- ▶ Folge: man macht sich Eigenschaften der Biologie zu Nutze
- ▶ man übernimmt die sehr einfach aber massiv parallel arbeitende Neuronen und die Lernfähigkeit
- ▶ Wichtig: das neuronale Netz ist nur eine Art Inspiration, eine Abstraktion von der Biologie

## Roboter

- ▶ besitzt zwei Motoren und 8 Sensoren
- ▶ aus Sensoren werden Eingabedaten gewonnen(reeller Zahlenwert)
- ▶ Roboter soll einfach nur fahren und anhalten wenn ein Hindernis im Weg steht
- ▶ Output ist binär ,  $H=1$  für Stopp und  $H=0$  für weiterfahren
- ▶ der klassische Weg

## der Weg des Lernens

- ▶ wir zeigen dem Roboter verschiedene Situationen in denen er sich befinden kann und er soll selbst lernen, was er zu tun hat
- ▶ dabei wird der Roboter einfach in die Landschaft platziert, damit er Sensorwerte für die verschiedenen Situationen liefern kann
- ▶ die Ausgabe geben wir selbst hinzu(H)
- ▶ neuronale Netz als Blackbox
- ▶ Angabe beispielhaften Input und dem dazugehörigen gewünschten Output = Trainingsbeispiel

## der Weg des Lernens

- ▶ Trainingsbeispiele lassen sich durch einfache Lernverfahren einen KNN beibringen
- ▶ Lernverfahren ist ein einfacher Algorithmus bzw. eine mathematische Formel
- ▶ generalisieren und eine allgemeingültige Vorschrift finden
- ▶ Ziel hinter der ganzen Sache ist nicht das Auswendiglernen der Beispiele, sondern das Erkennen eines Prinzips hinter der ganzen Sache
- ▶ ständiger Kreislauf

# Geschichtlicher Hintergrund

## Anfänge

- ▶ Beginn bereits in den frühen 1940er Jahren
- ▶ 1943: neurologische Netzwerke, Schwellwertschalter und arithmetische Funktionsberechnung
- ▶ 1949: klassische Hebb'sche Lernregel

## Blütezeit

- ▶ 1951: Neurocomputer Snark
- ▶ 1956: Dartmouth Summer Research Project
- ▶ 1958: erster erfolgreicher Neurocomputer (Perceptron)
- ▶ 1960: erstes kommerziell verbreitetes Neuronales Netz

## Lange Stille und langsamer Wiederaufbau

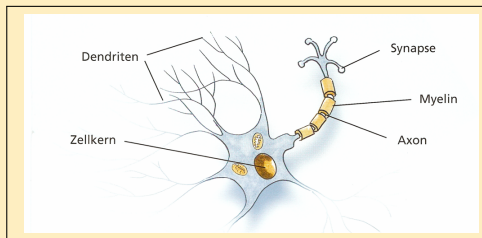
- ▶ 1972: Assoziativspeichermodell durch Teuvo Kohonen
- ▶ 1982: Kohonen- Netze und Hopfield- Netze

## Renaissance

- ▶ 1985: Lösung des Traveling Salesman Problem mit Hopfieldnetzen



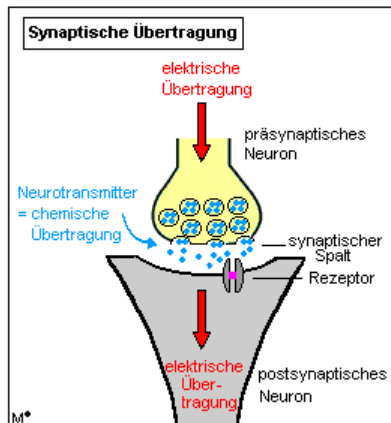
## Aufbau Neuronen



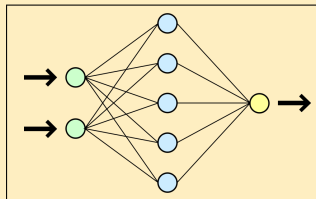
- ▶ Aufnahme, Verarbeitung und Weiterleitung von Reizen
- ▶ elektrische Signal läuft durch das Axon
- ▶ an den Synapsen wird das Signal chemisch weitergereicht
- ▶ Synapsen sind mit anderen Zellen verbunden
- ▶ Dendriten stellen Kontakt über Synapsen zu anderen Zellen her

## synaptische Übertragung

- ▶ Erregung geht durch die ganze Nervenzelle zur Synapse
- ▶ Überträgersubstanz wird frei
- ▶ strömt über den synaptischen Spalt zur gegenüberliegenden Membran(Dendrite)
- ▶ diese wird anschließend erregt



## Eigenschaften KNN



- ▶ Verknüpfung einzelner Prozessoren, welche einfache Rechnungen ausführen können

## Gewichte

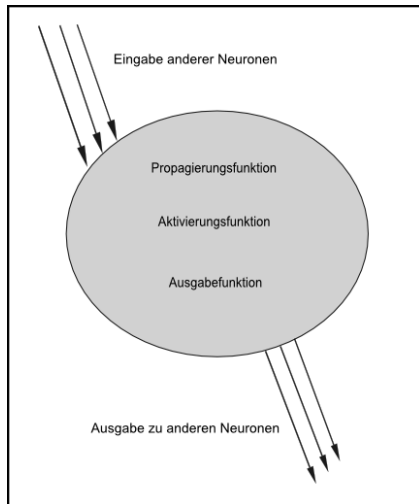
- ▶ Verbindung wird durch ein Gewicht bewertet

## Gewichte

- ▶ Gewicht = Stärke der Verbindung zwischen zwei Neuronen
- ▶ Modifizierung der Gewichte durch wiederholte Eingabe Trainingsbeispiele
- ▶ Wissen der KNN wird sozusagen in den Gewichten gespeichert
- ▶ durch das Lernen werden die Gewichte verändert

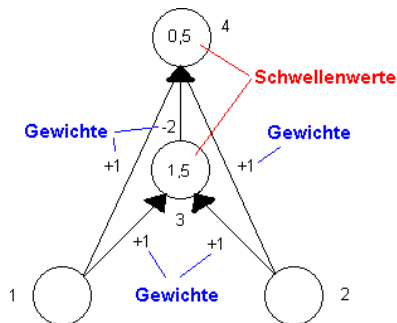
## Bestandteile

- ▶ Neuronen
- ▶ Verbindungen
- ▶ Propagierungsfunktion
- ▶ Netzeingabe
- ▶ Aktivierung
- ▶ Schwellenwert
- ▶ Aktivierungsfunktion
- ▶ Ausgabefunktion
- ▶ Lernregeln
- ▶ Umgebung



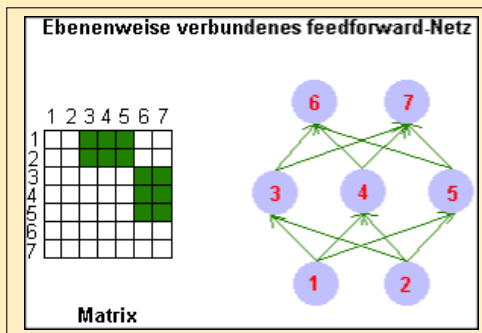
## XOR

- ▶ besteht aus 4 Zellen
- ▶ für die Aktivierung werden die binären Werte 0 und 1 verwendet
- ▶ Netzeingabe wird berechnet durch die PGF
- ▶ Aktivierungsfunktion stellt in diesem Fall die binäre Schwellenwertfunktion dar
- ▶ Ausgabe ist in diesem Fall die Identität



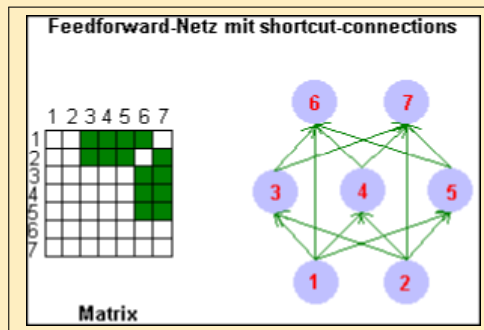
## FeedForward

- ▶ besteht aus einer Eingabeschicht, n versteckten Verarbeitungsschichten sowie einer Ausgabeschicht
- ▶ Verbindungen nur zu Neuronen in der nächsten Schicht erlaubt



## FeedForward mit ShortCutConnection

- ▶ Verbindungen erlaubt, welche eine Schicht überspringen





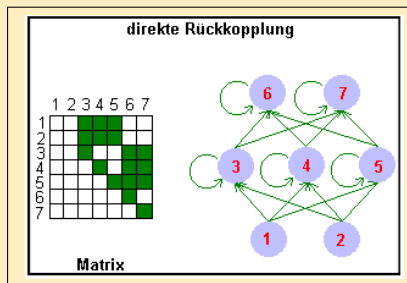
# Netztopologien

## FeedBack

- ▶ Neuron kann sich selbst beeinflussen

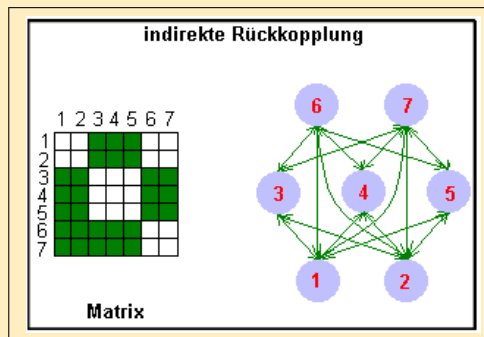
## direkte Rückkopplung

- ▶ Verbindungen von einem Neuron zu sich selbst



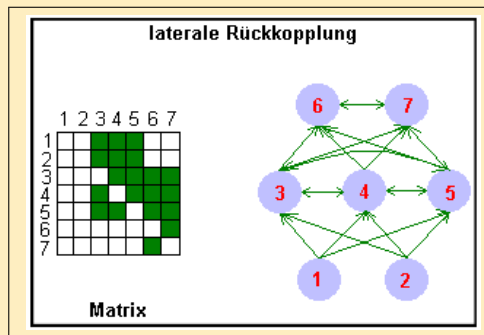
## indirekte Rückkopplung

- ▶ Verbindungen in Richtung der Eingabeschicht erlaubt
- ▶ Neuron kann sich durch einen Umweg selbst beeinflussen



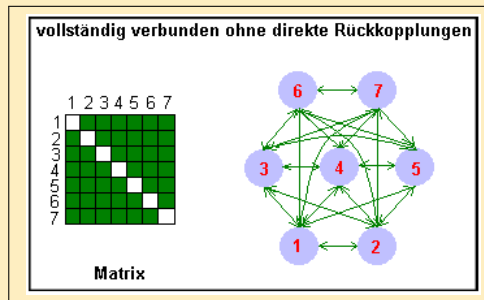
## lateral

- ▶ Verbindungen innerhalb einer Schicht bzw. Ebene erlaubt



## vollständig verbunden

- ▶ Verbindungen zwischen allen Neuronen erlaubt, außer der direkten Rückkopplung



## Möglichkeiten

- ▶ neue Verbindungen entwickeln
- ▶ vorhandene Verbindungen löschen
- ▶ Verbindungsgewichte ändern
- ▶ Schwellwerte von Neuronen ändern
- ▶ Funktionen abwandeln (PGF, AWF, AF)
- ▶ neue Neuronen entwickeln
- ▶ vorhandene Neuronen löschen
- ▶ Das verändern der Verbindungsgewichte stellt die meist benutzte Lernmethode dar

# Arten des Lernens

## unüberwacht

- ▶ biologisch plausibelste Lösung
- ▶ gegeben ist nur das Eingabemuster und das Netz versucht ähnliche Muster zu erkennen und in ähnliche Kategorien zu klassifizieren

## bestärkt

- ▶ nach erfolgreichem Durchlauf wird ein Wahrheitswert oder ein reeller Wert geliefert
- ▶ definiert ob das gelieferte Ergebnis mit dem gewünschten Ergebnis überein stimmt

## überwacht

- ▶ es existiert eine Trainingsmenge von Eingabemustern inklusive deren korrekte Ergebnisse in Form der genauen Aktivierung sämtlicher Ausgabeneuronen
- ▶ Ausgabe kann direkt mit der korrekten Lösung verglichen werden
- ▶ anhand der Differenz kann die Netzgewichtung geändert werden
- ▶ Ziel: Netz kann nach diesem Training von allein assoziieren

## Hebbsche - Lernregel

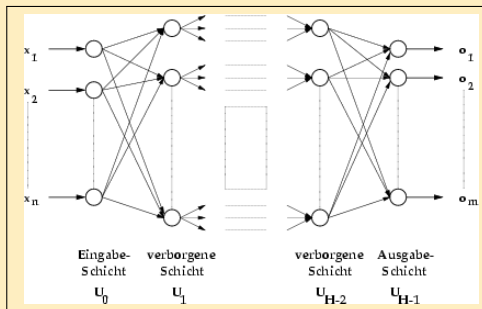
- ▶ stellt in allgemeiner Form fast alle neuronalen Lernverfahren dar
- ▶ je häufiger ein Neuron A gleichzeitig mit einem Neuron B aktiv ist, umso bevorzugter werden diese beiden Neuronen aufeinander reagieren
- ▶ Veränderung der synaptischen Übertragung als Gewichtsänderung des neuronalen Graphen



## Delta Regel

- ▶ beruht auf dem Vergleich zwischen dem gewünschten und dem tatsächlichen beobachteten Output
- ▶ beobachtete Aktivität zu niedrig: Gewichte zwischen dem Sender/Empfänger erhöht
- ▶ beobachtete Aktivität zu hoch: Gewichte verringert, das heißt die Verbindungen werden geschwächt
- ▶ gleich groß: keine Änderung, da der gewünschte Wert ausgegeben wird

## Backpropagation



- ▶ wird bei Neuronalen Netzen mit Hidden- Units angewendet
- ▶ Rechenvorschrift für die Modifizierung der Gewichte

# Backpropagation

## Problem

- ▶ Keine direkte Fehlerbestimmung in der Hidden- Schicht
- ▶ Modifikation: 3 Schritte der Trainingsphase für Gewichtsveränderungen

## Forward- Pass

- ▶ wie in Trainingsphase üblich: Präsentation der Reize und Berechnung des Output

## Fehlerbestimmung

- ▶ ermitteln der Fehlergröße und gegebenenfalls Abbruch oder Backward- Pass

## Backward-Pass

- ▶ Ausbreiten der Fehlerterme in entgegengesetzter Richtung bis zur Input- Schicht
- ▶ Auf diesem Weg erfolgt die Neumodifizierung der Gewichte

## Gewichtsanpassung

- ▶ erfolgt über Gradientenabstiegsverfahren

## Kohonen Netze

- ▶ wird auch Self- Organizing Map genannt
- ▶ Arbeitet ohne externen Lehrer
- ▶ Lernen in selbstorganisierender Weise Karten eines Inputraums zu erstellen
- ▶ Der Inputraum wird also geclustert

## Geschichtliches

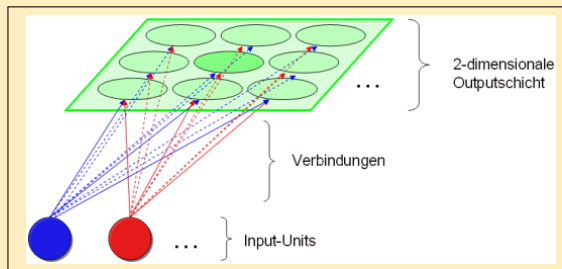
- ▶ Konzipiert in den 80er Jahren vom finnischen Ingenieur Teuvo Kohonen
- ▶ Allerdings auch schon in den 70er Jahren Ansätze

## Vorteil

- ▶ es ist auf Grund der Tatsache, dass der Mensch auch ohne externen Lehrer Probleme löst, biologisch plausibler.

## Aufbau

- ▶ besitzt eine Input- und Outputschicht
- ▶ Von jedem Input-Neuron besteht eine Verbindung zu jedem Output-Neuron



## Startwert festlegen

- ▶ zufälliges generieren der Gewichte
- ▶ festlegen der Lernkonstanten
- ▶ festlegen der Nachbarschaftsfunktionen und maximale Anzahl an Durchläufen

## Auswahl eines Inputvektors

- ▶ Auswahl eines Inputvektors, oder auch zufällige Generierung dessen



## Aktivitätsberechnung und Auswahl

- ▶ Berechnung der Aktivität der Output-Neuronen
- ▶ Auswahl der Unit mit der maximalen Erregung

## Gewichtsmodifikation

- ▶ Modifikation der Gewinner-Unit
- ▶ Modifikation der Nachbarschaft
- ▶ Radius für Nachbarschaft eingrenzen
- ▶ Lernparameter reduzieren

## Abbruch

- ▶ Wenn die maximale Anzahl der Durchläufe erreicht ist

## Hopfield Netze

- ▶ 1982 durch John Hopfield bekannt gemacht
- ▶ Ist ein Feedback-Netz
- ▶ Nur eine Schicht, welche Ein- und Ausgabeschicht zugleich ist
- ▶ Neuronen alle untereinander verbunden
- ▶ Alle Gewichte symmetrisch

## Anwendung

- ▶ Autoassoziativspeicher zum Zwecke der Musterrekonstruktion

In 3 Phasen unterteilt

## Trainingsphase

- ▶ Einspeichern von vorgegebenen Mustern

## Anlegen neuer Eingaben

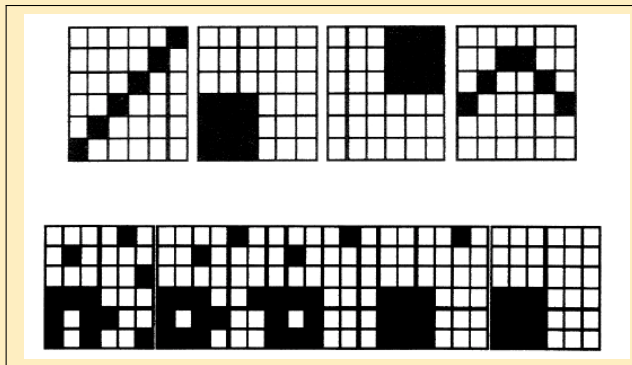
- ▶ Zurücksetzen der Knoten in den Anfangszustand
- ▶ Initialisieren eines unvollständigen oder verrauschten Testmusters

## Rechenphase

- ▶ Neuron für Neuron wird aktualisiert und getestet ob es nahe einem gelernten Muster ist
- ▶ Abbruch wenn stabiler Endzustand erreicht ist, welcher der Assoziation gleichbedeutend ist

# Hopfield Netze

## Hopfield Netze



## Anwendungsbeispiele

- ▶ Steuerung
- ▶ Diagnostik
- ▶ Vorhersage
- ▶ Mustererkennung

Vielen Dank für die Aufmerksamkeit!