

UNIVERSITÄT

BONN

AIS

Computational Intelligence

8. Unüberwachtes Lernen cont.

Evolutionäre Algorithmen

Prof. Dr. Sven Behnke

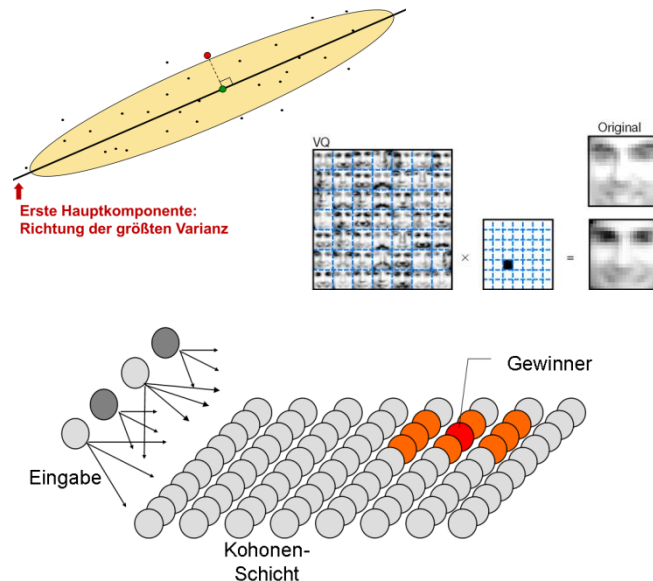
KLAUSUR

- 20. Juli 15:00-17:00 im HS 1+2 (Hörsaalzentrum, Endenicher Allee 19c)
- 90 Minuten Bearbeitungszeit
- Closed Book

Letzte Vorlesung: Unüberwachtes Lernen

■ Entdecken nützlicher Struktur in den Daten

- Autoencoder
- Hauptkomponentenanalyse (PCA)
- Nichtnegative Matrixfaktorisierung (NMF)
- Gruppierung / Clusterung / Vektorquantisierung
- Independent Component Analysis (ICA)
- Selbstorganisierende Karten (SOM)



■ Lernregeln

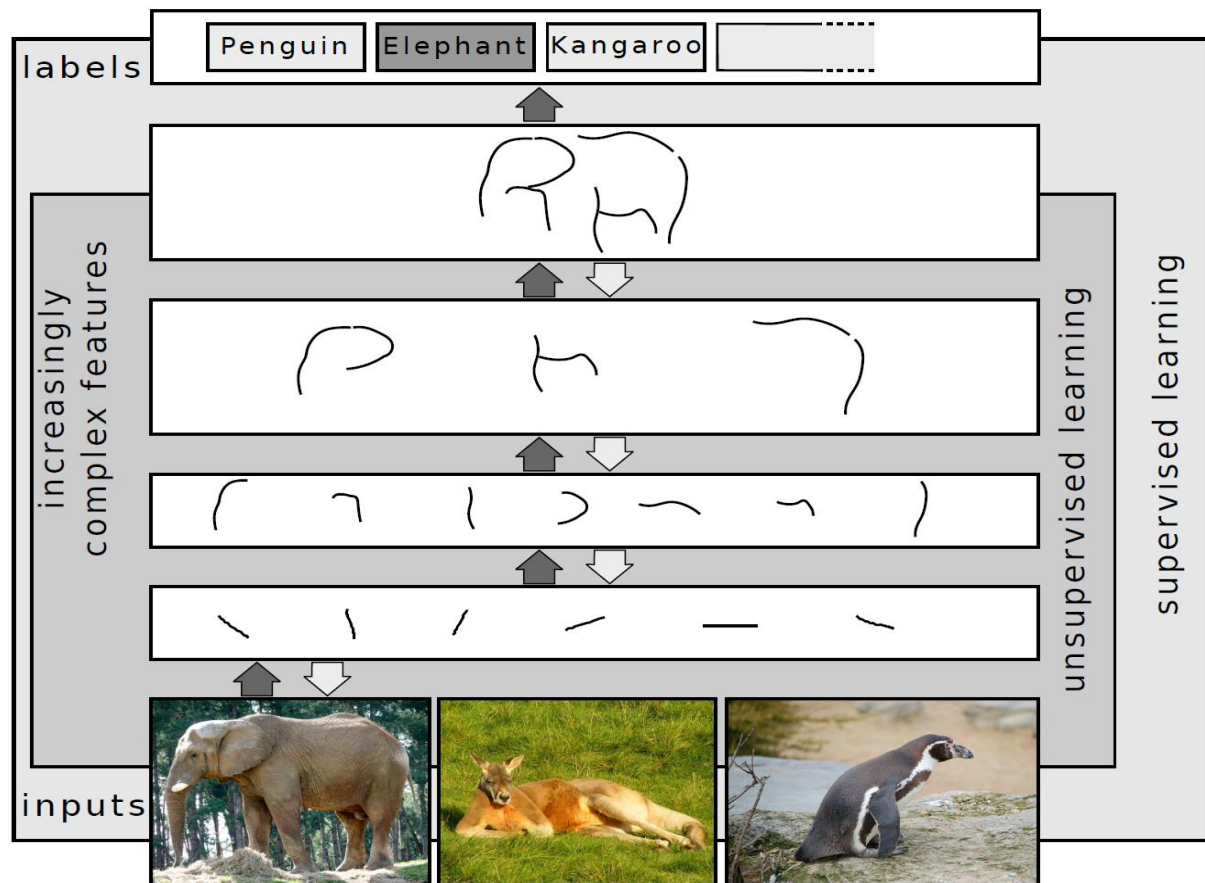
- Hebb'sche Lernregel
- Kompetitive Lernregel

$$\Delta w_{ij} = \eta y_i \left(x_j - \sum_{k=1}^i w_{kj} y_k \right)$$

$$\Delta \mathbf{w}_i = \eta (\mathbf{x}_p - \mathbf{w}_i)$$

DEEP LEARNING

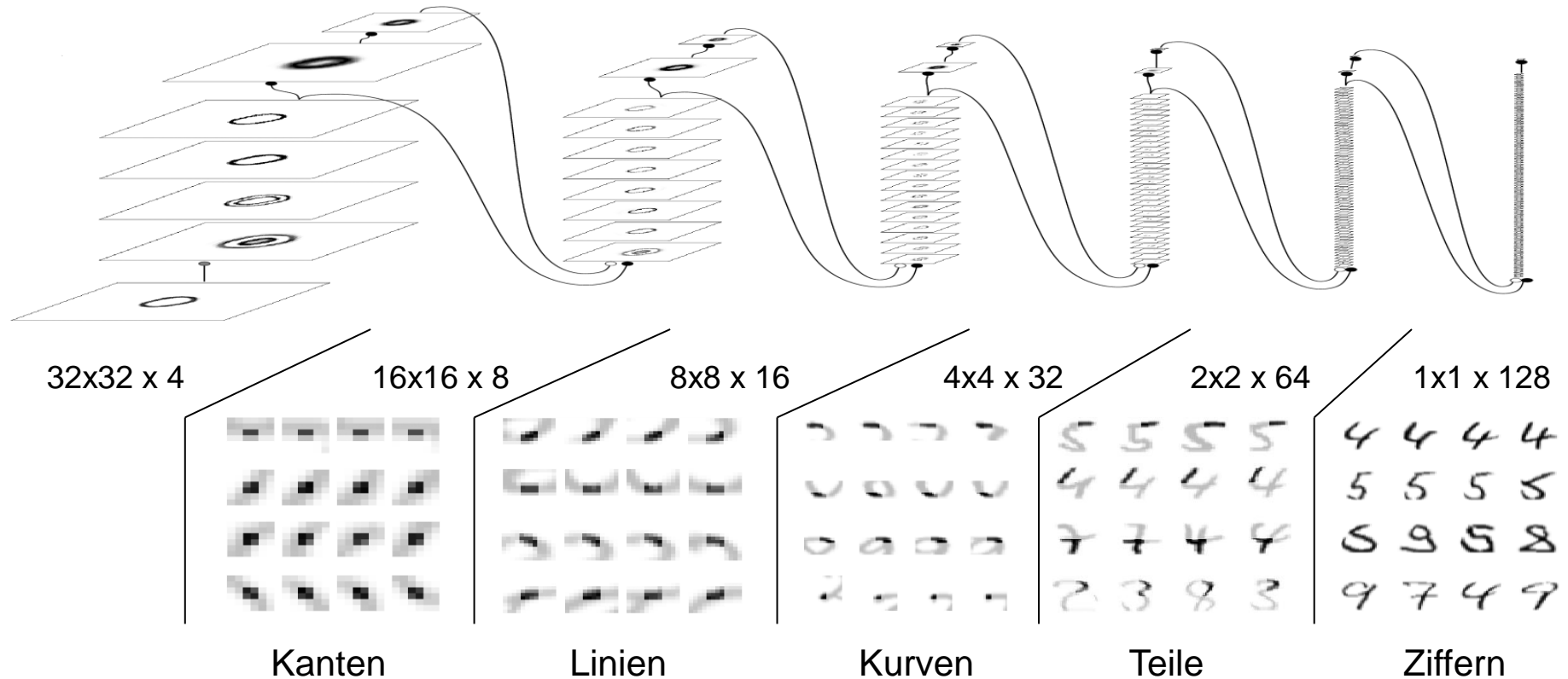
- Lernen hierarchischer Repräsentationen
- Höhere Konzepte durch Kombination niederer Konzepte definiert
- Wiederverwendung von niederen Konzepten in mehreren höheren Konzepten



LERNEN VON MERKMALSHIERARCHIEN

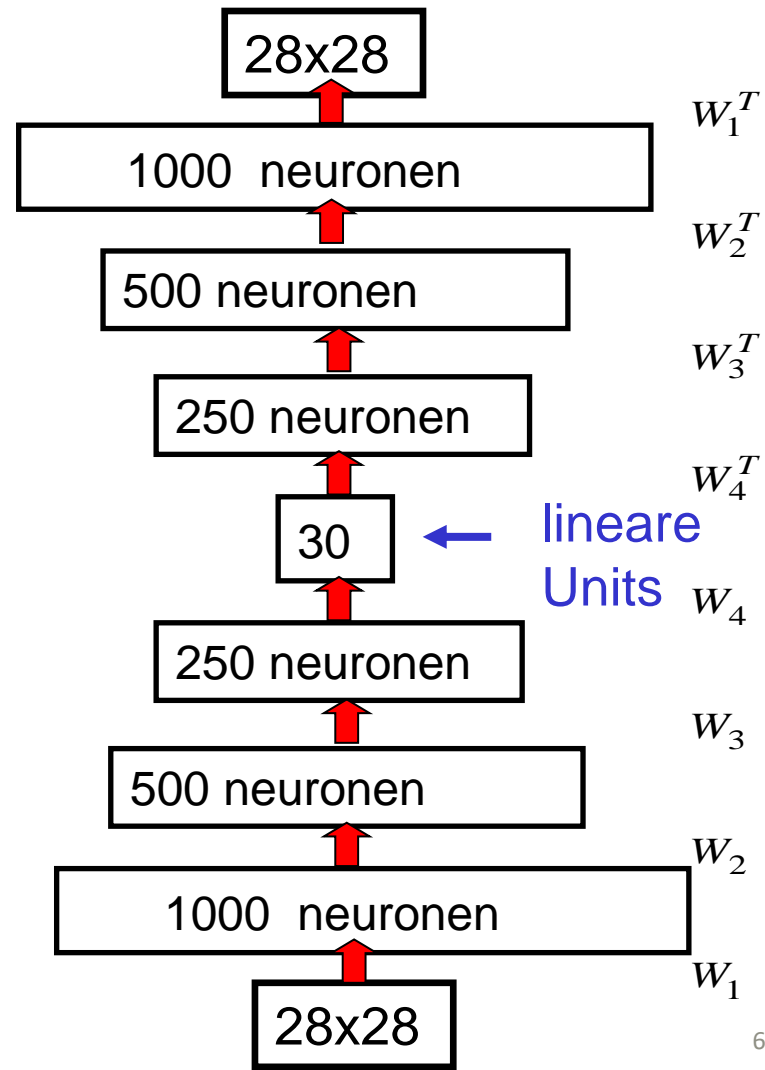
■ Hebbsches Lernen und Wettbewerb

[Behnke, IJCNN1999]

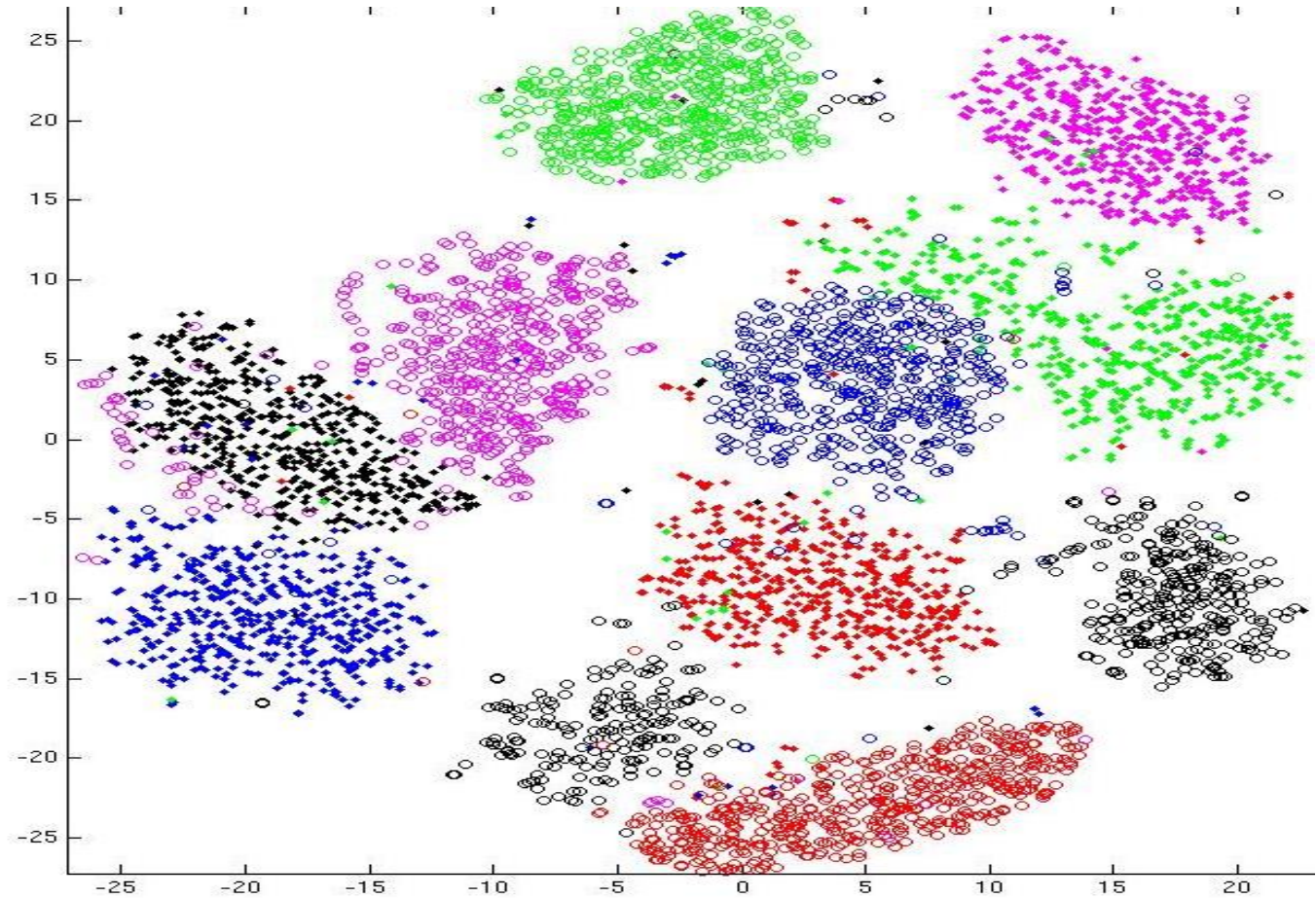


TIEFE AUTOENCODER (HINTON & SALAKHUTDINOV, 2006)

- Trainiert durch schichtweises Lernen
- Zunächst unüberwachtes Training von Boltzmann-Maschinen ohne seitliche Verbindungen (RBM, Contrastive Divergence)
- Danach Finetuning durch Backpropagation

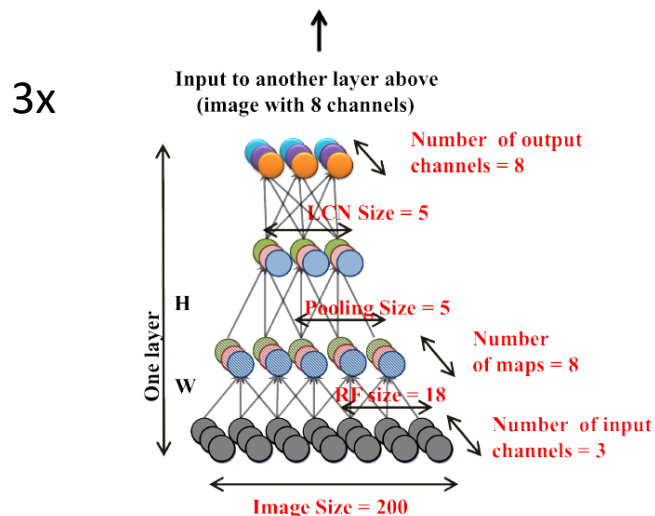
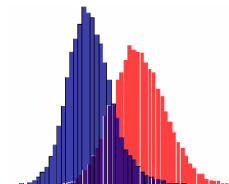


2D-REPRÄSENTATION VON MNIST-ZIFFERN



UNÜBERWACHTES LERNEN VON BILDMERKMALEN

- 9 Schichten
- Lokale Verbindungen
- Spärlicher Autoencoder
- L2-Pooling
- Lokale Kontrastnormalisierung
- 1 Milliarde Verbindungen
- Trainiert auf 10 Millionen Bildern
- Unüberwacht gelernte Merkmale:



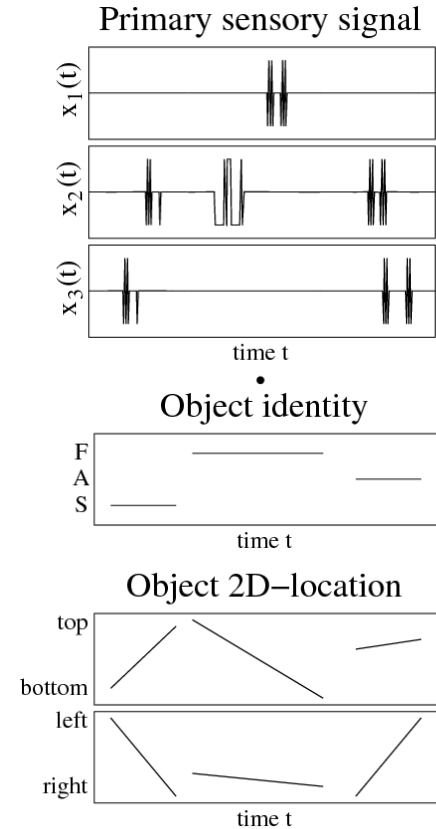
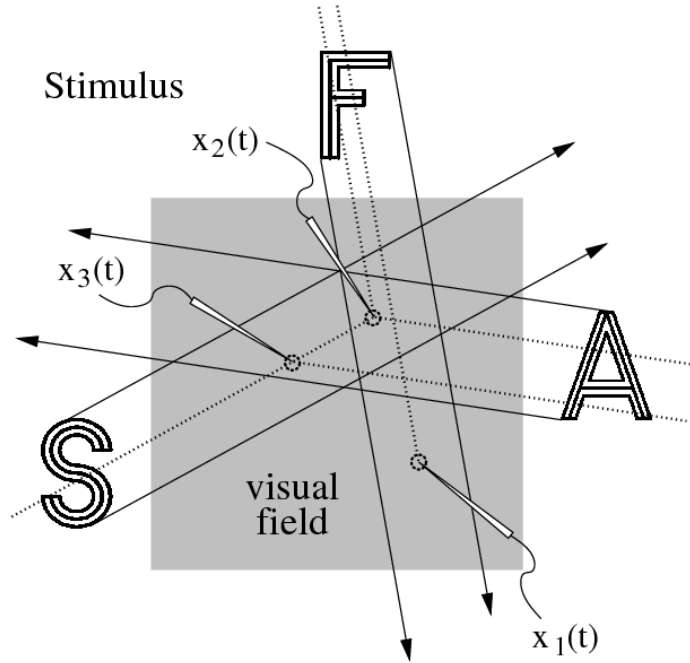
- Überwachtes Training auf ImageNet 2011 (14M Bilder, 22K Kategorien): 15.8%

[Le et al. 2012]

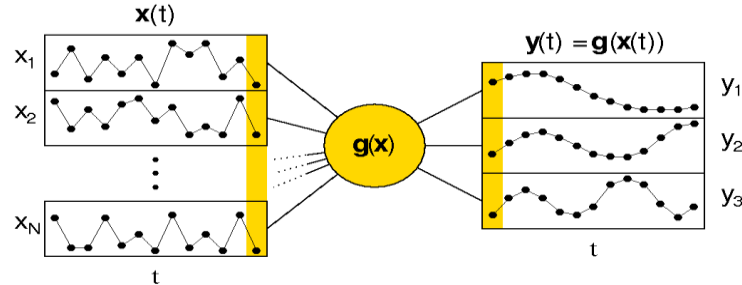
SLOW FEATURE ANALYSIS (SFA) [WISKOTT]

■ Motivation:

- Einzelne Bildpunkte variieren stark
- Objektidentität und Position variieren langsam



SLOW FEATURE ANALYSIS - PROBLEM



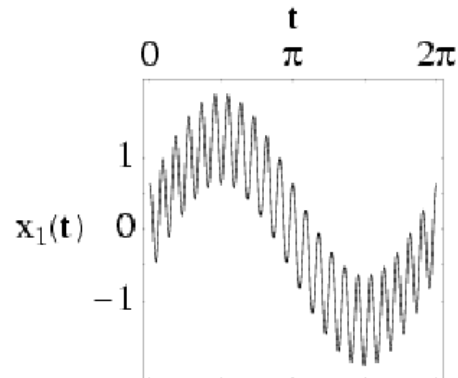
- Gegeben ein (hochdimensionales) Eingabe-Signal $x(t)$.
Finde Funktionen $g_i(x)$, sodass Ausgabe $y_j(t) := g_j(x(t))$ sich möglichst langsam ändert.

Minimiere $\Delta(y_j) := \left\langle \dot{y}_j^2 \right\rangle_t$

unter den **Nebenbedingungen**

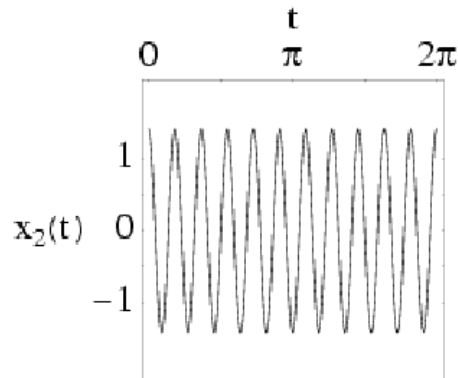
- $\left\langle y_j \right\rangle_t = 0$ (Mittelwert Null),
- $\left\langle y_j^2 \right\rangle_t = 1$ (normierte Varianz),
- $\forall i < j : \left\langle y_i y_j \right\rangle_t = 0$ (Dekorrelation und Reihenfolge).

SLOW FEATURES - BEISPIEL



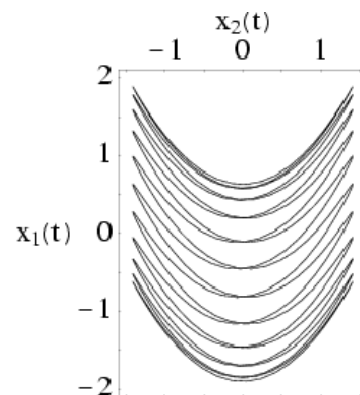
input component $x_1(t)$

$$x_1(t) := \sin(t) + \cos(11t)^2$$



input component $x_2(t)$

$$x_2(t) := \cos(11t)$$

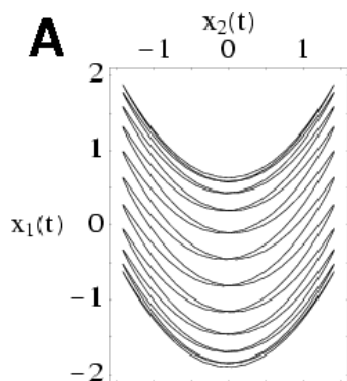


input trajectory $\mathbf{x}(t)$

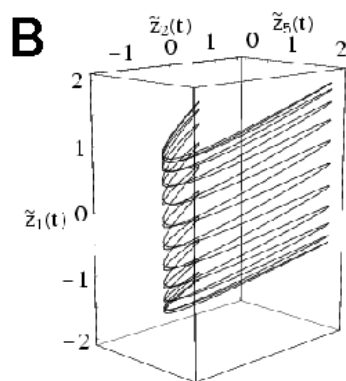
■ Langsames Merkmal: $y(t) = x_1(t) - x_2(t)^2 = \sin(t)$

■ Extrahiert mit Polynom 2. Grades: $g(\mathbf{x}) = x_1 - x_2^2$

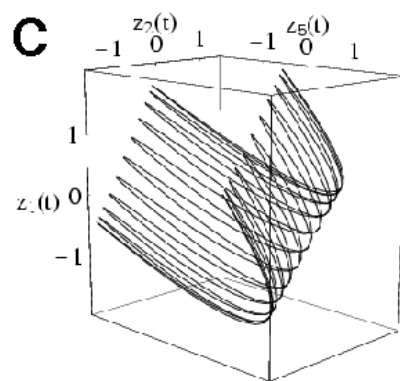
SFA-ALGORITHMUS



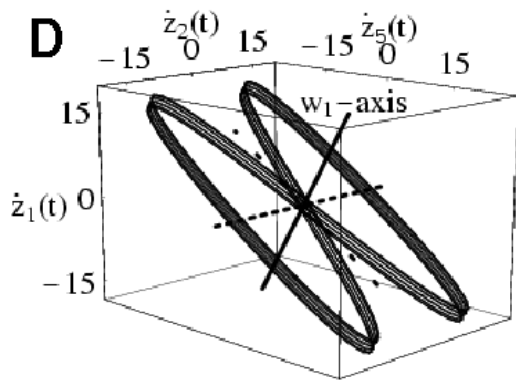
input signal $\mathbf{x}(t)$



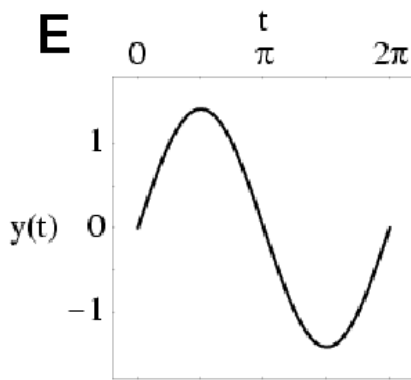
expanded signal $\tilde{\mathbf{z}}(t)$



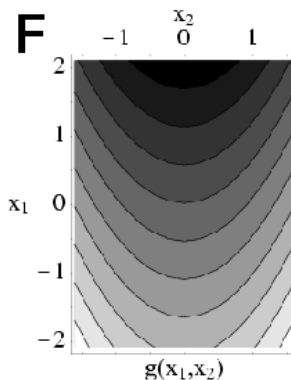
sphered signal $\mathbf{z}(t)$



time derivative signal $\dot{\mathbf{z}}(t)$



output signal $y(t)$



function $g(\mathbf{x})$

1. Signal expandieren
(z.B. in Polynome
2. Grades)
-> Lineares Problem

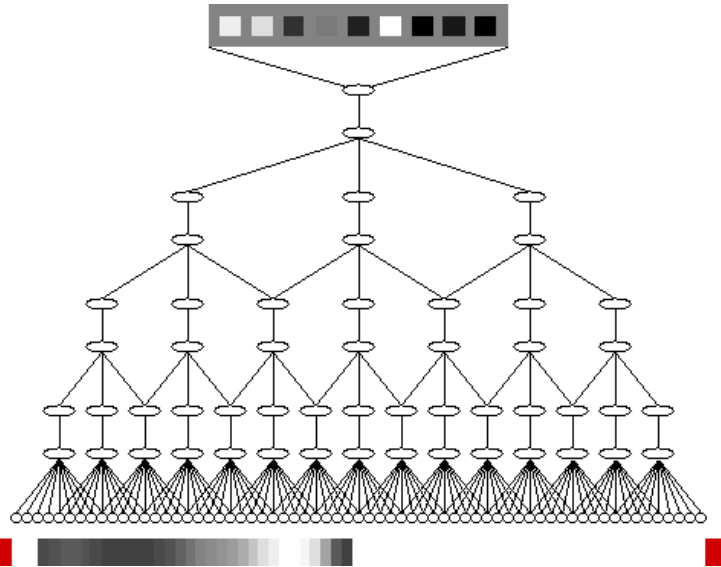
$$\tilde{z}_1 := x_1, \tilde{z}_2 := x_2, \tilde{z}_3 := x_1^2, \\ \tilde{z}_4 := x_1 x_2, \tilde{z}_5 := x_2^2$$

2. Signal normalisieren
(Whitening) (Mittelwert
abziehen, PCA)

3. Zeitliche Ableitung
berechnen $\dot{\mathbf{z}}(t)$

4. Finde die Richtungen
mit den kleinsten
Änderungen (PCA mit
kleinsten Eigenwerten)

HIERARCHISCHE SFA



- Verschiedene 1D-Signale
- Trainiert mit Transformationen

■ Untrainiert:

naive network



■ Translationsinvarianz:

trained network
training data



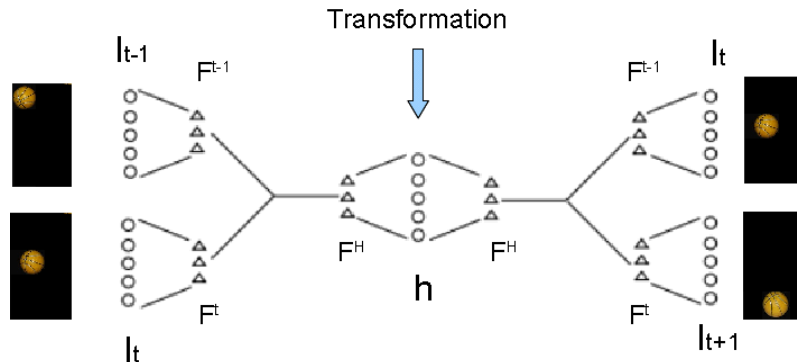
■ Skalierungsinvarianz:

size invariance
trained network
training data

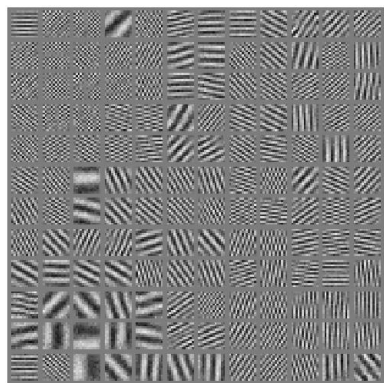


RELATIONALER AUTOENCODER

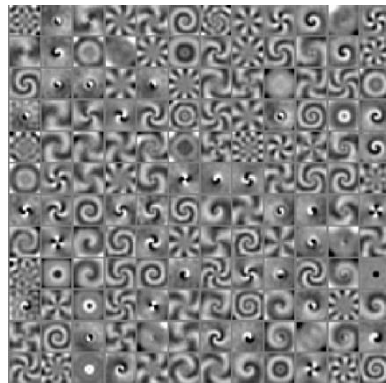
■ Lerne Bildtransformation:



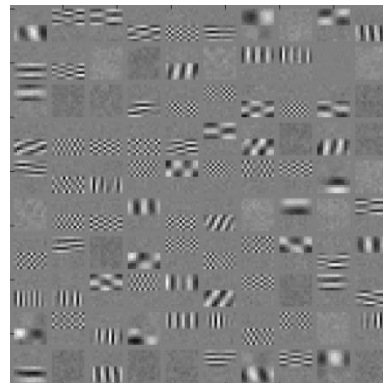
■ Gelernte Basisfunktionen:



Verschiebungen



Drehungen



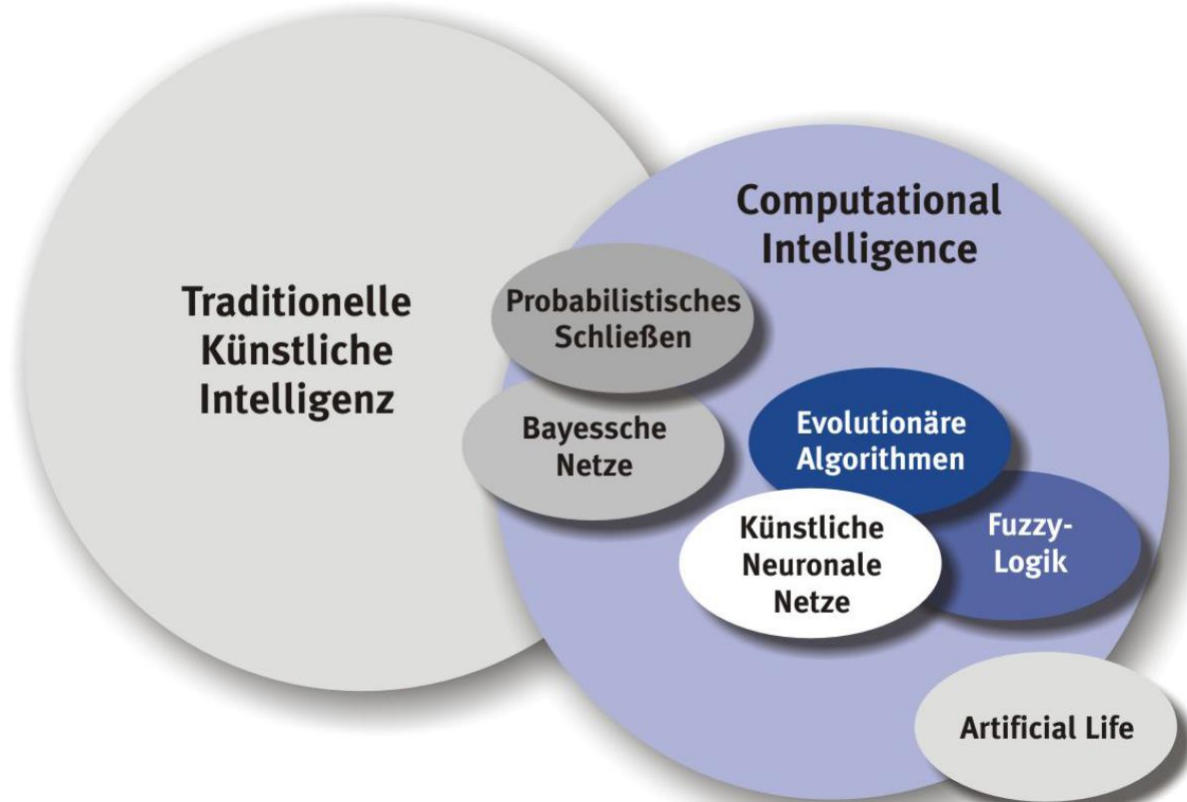
Zwei Verschiebungen

$$h_k(I_{t-1}, I_t) = \sigma\left(\sum_f F_{kf}^H \sum_i F_{if}^{I_{t-1}} I_{t-1,i} \sum_j F_{jf}^{I_t} I_{t,j}\right)$$

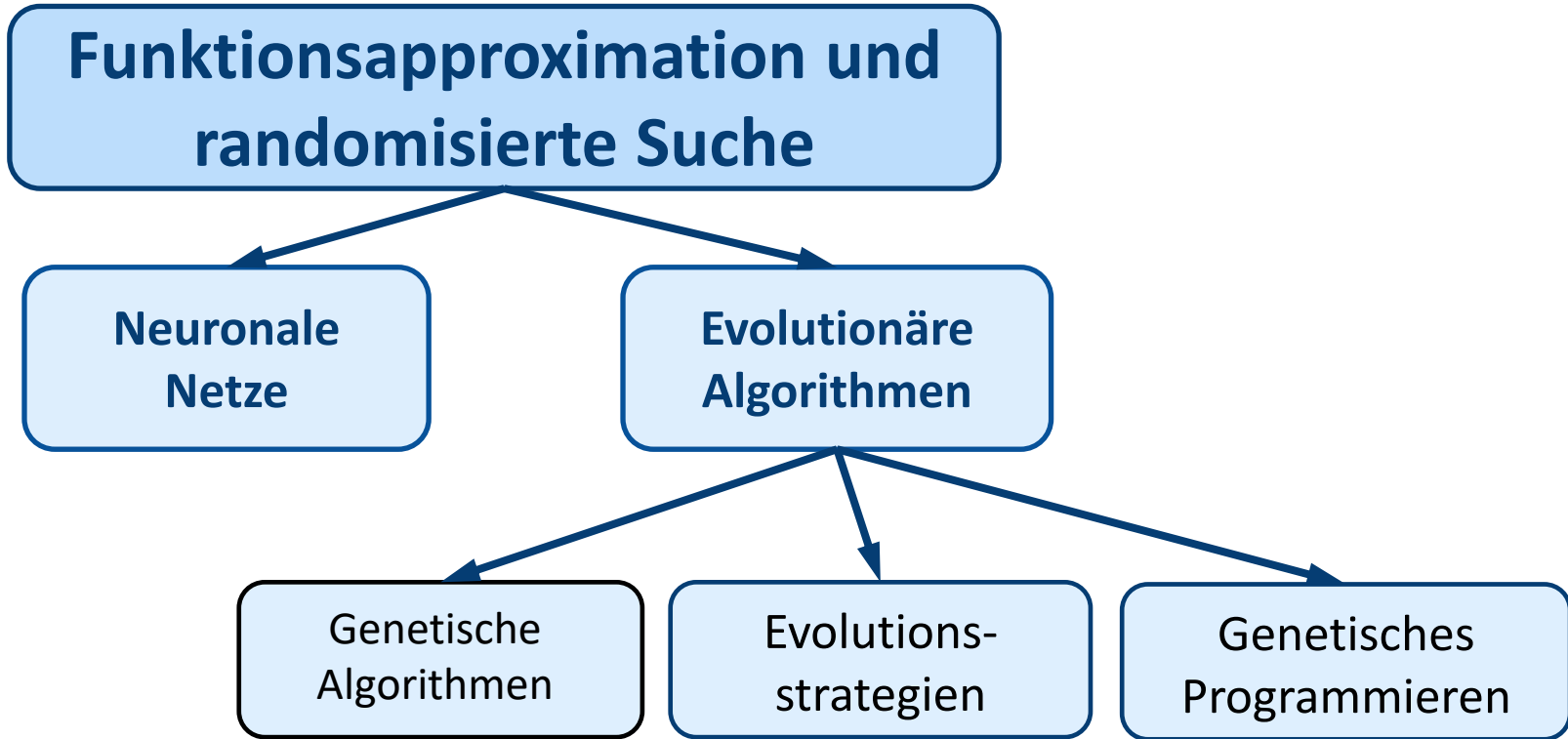
$$\hat{I}_{t+1,i}(h_k(I_{t-1}, I_t)) = \sum_f F_{jf}^{I_t} \sum_i F_{if}^{I_{t-1}} I_{t,i} \sum_k F_{kf}^H h_k$$

[Memisevic, PAMI 2013]

ERINNERUNG: BAUSTEINE DER COMPUTATIONAL INTELLIGENCE



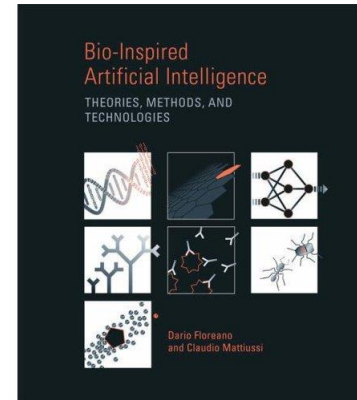
EVOLUTIONÄRE ALGORITHMEN



EVOLUTIONÄRE ALGORITHMEN



- Dario Floreano and Claudio Mattiussi:
Bio-Inspired Artificial Intelligence
MIT Press, 2008
- <http://baibook.epfl.ch/>

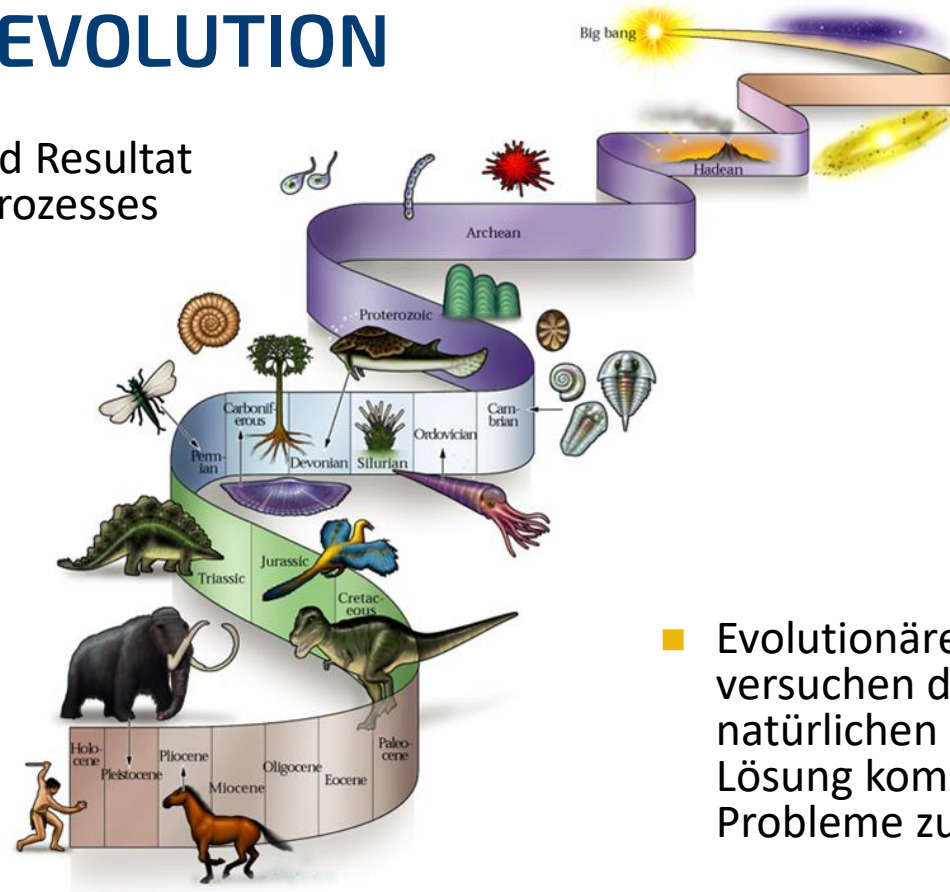


MOTIVATION DURCH BIOLOGISCHE EVOLUTION

- Existierende Arten sind Resultat eines evolutionären Prozesses

- Biologische Systeme sind

- robust
- komplex
- adaptiv

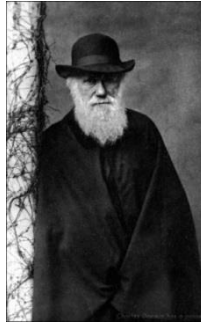


- Evolutionäre Algorithmen versuchen den Prozess der natürlichen Evolution auf die Lösung komplexer technischer Probleme zu übertragen

- Erzeugt die natürliche Evolution immer komplexere Systeme?

VIER GRUNDELEMENTE DER EVOLUTION

- All species derive from common ancestor [Charles Darwin, 1859 *On the Origins of Species*]



- **Population**

- Gruppe von Individuen

- **Diversität**

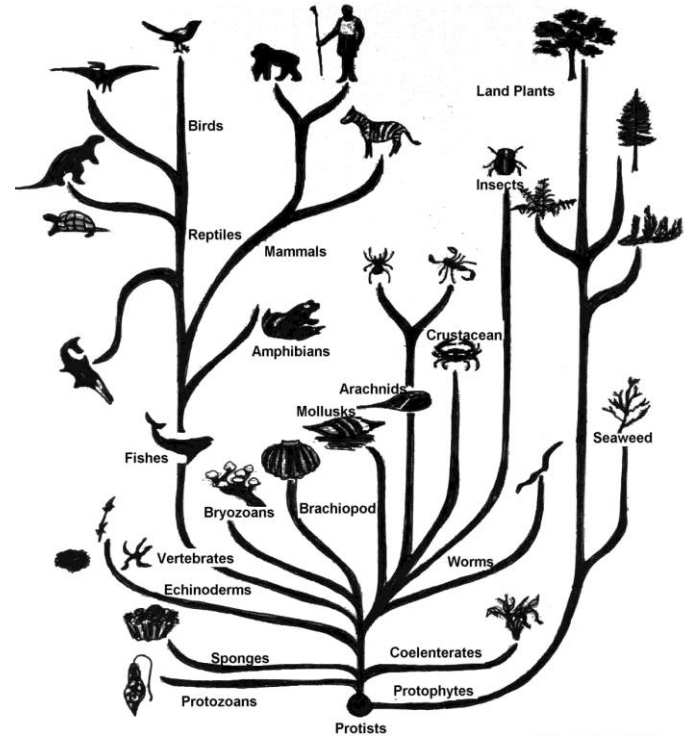
- Individuen haben verschiedene Eigenschaften

- **Vererbung**

- Eigenschaften werden an Nachkommen weiter gegeben

- **Selektion**

- Es werden mehr Nachkommen erzeugt als die Umgebung zulässt
 - Nur am besten an den Lebensraum angepasste Nachkommen überleben



PHÄNOTYP & GENOTYP

- **Phänotyp** = Ausprägung des Organismus (Aussehen, Verhalten, etc).
 - Selektion erfolgt anhand des Phänotyps
 - Hängt von vielen Umweltfaktoren ab
 - Ändert sich durch Entwicklung, Lernen, Alterung, ...
- **Genotyp** = Das Genmaterial eines Organismus
 - Wird durch Fortpflanzung übertragen
 - Wird durch Mutationen verändert
 - Selektion erfolgt nicht direkt auf dem Genotyp
- **Genomik** = Vererbungslehre, Struktur und Funktion der Gene
- **Funktionelle Genomik** = Aufklärung der Rolle der Gene im Organismus
- Zwillingsstudien beschäftigen sich mit der Frage, zu welchem Anteil wir durch das Genom und zu welchem Anteil wir durch die Umwelt bestimmt sind.

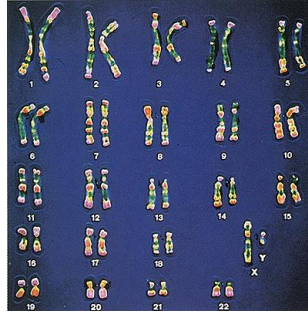


Jean-Felix & Auguste Piccard

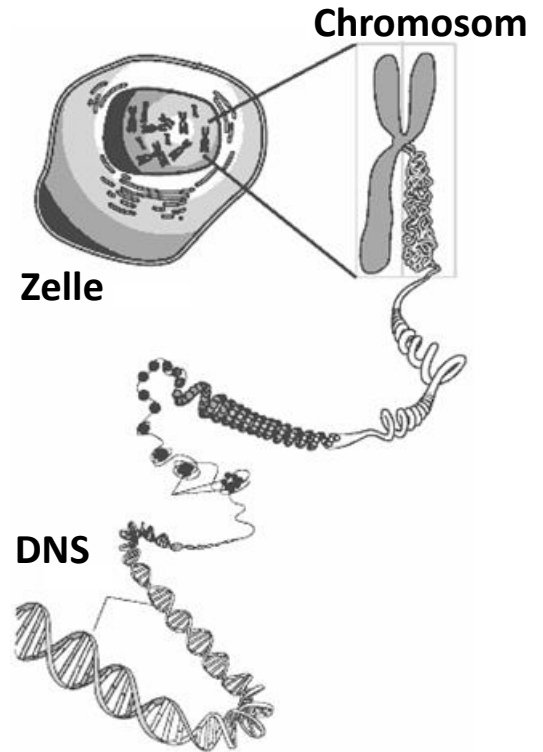
DNS (DESOXYRIBONUKLEINSÄURE)

- Lange Molekülspiralen, zusammengefoldet

- Menschen haben 23 DNA/DNS-Molekülpaare (**Chromosomen**)



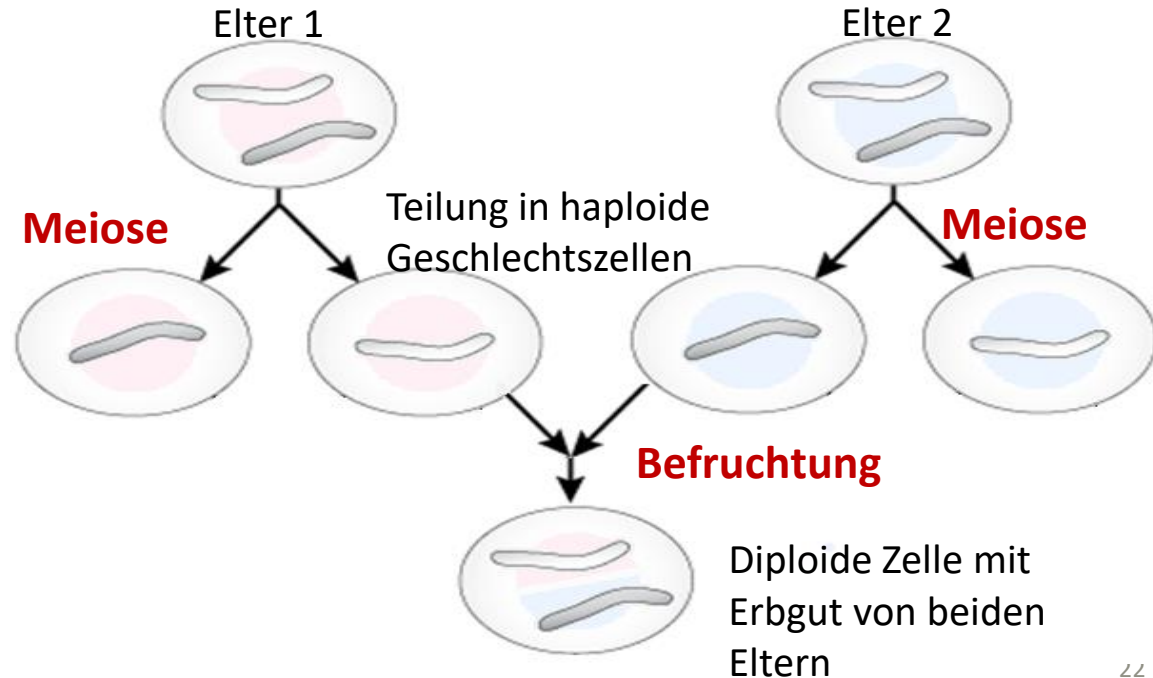
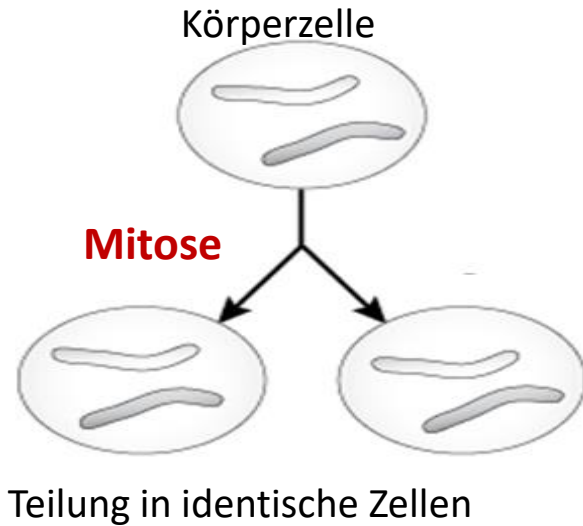
- DNS besteht aus zwei komplementären Basensequenzen aus vier Nukleotiden (A, T, C, G), die jeweils in Paaren binden (A-T und C-G)



- Ein **Gen** ist eine Nukleotidsequenz die den Bau eines Proteins beschreibt.

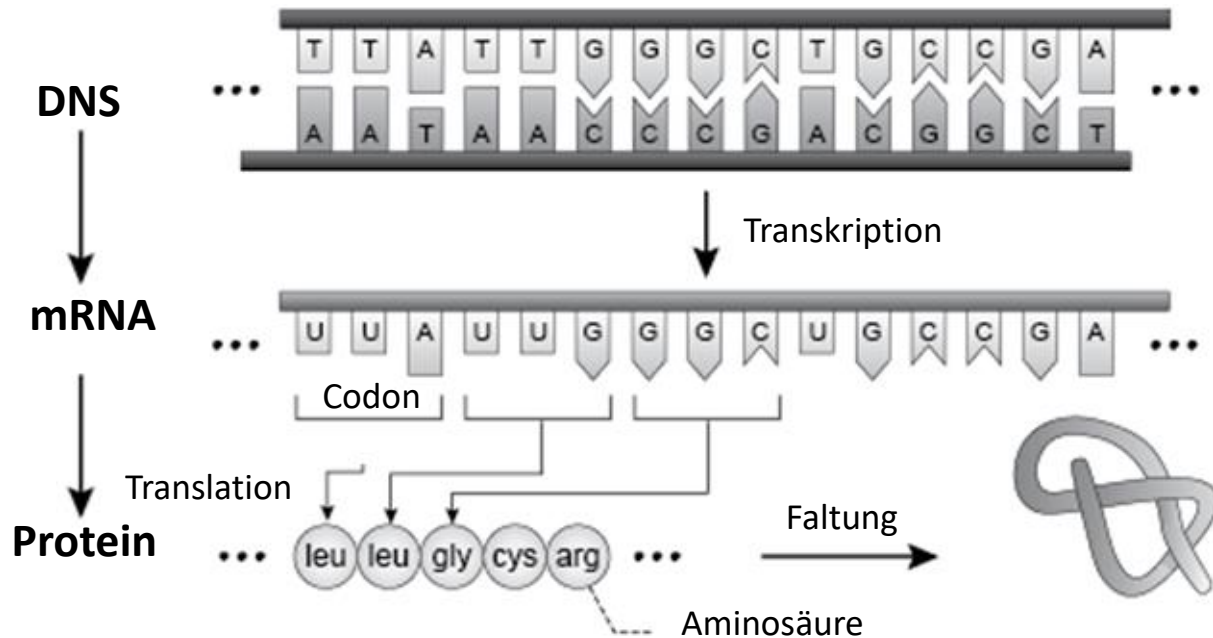
ZELLTEILUNG (REPLIKATION)

- Zellen werden auf zwei Wegen repliziert:
 - **Mitose**: Während der Entwicklung eines Organismus aus Eizelle
 - **Meiose**: Bei der sexuellen Fortpflanzung



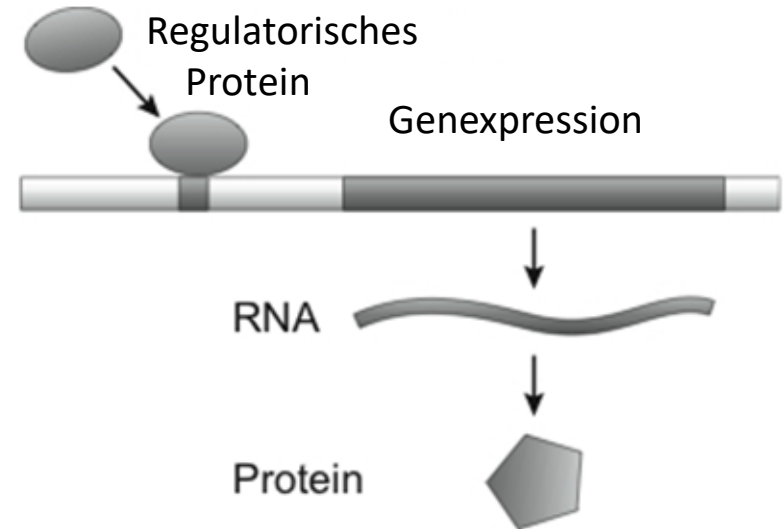
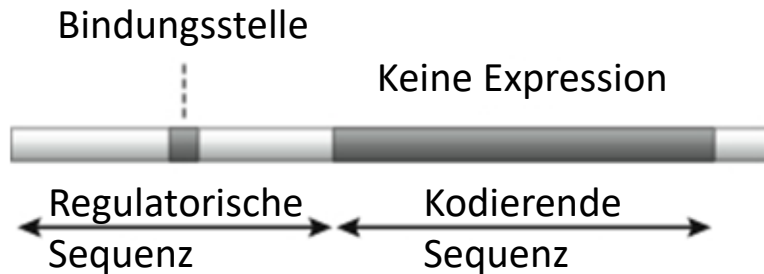
GENEXPRESSION

- **Proteine** bestimmen Typ und Funktion von Zellen (z.B. bestehen Haarzellen und Muskeln aus verschiedenen Proteinen).
- Die **Nukleotidsequenz** beschreibt den Zusammenbau des Proteins.
- Die Expression eines Gens in ein Protein wird durch Boten-RNA (Messenger/m-RNA) vermittelt.



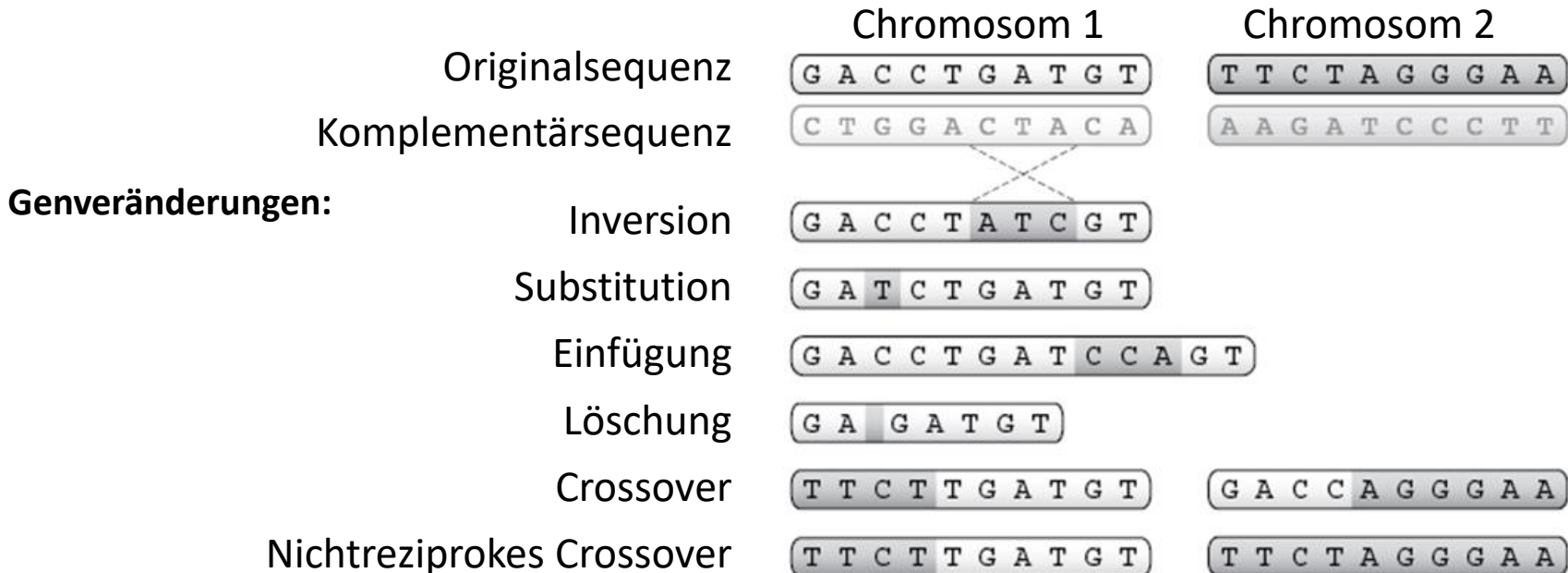
GENREGULATION

- Steuerung der Genexpression durch regulatorische Sequenzen außerhalb der kodierenden Region.
- Bindung eines bestimmten Proteins an die regulatorische Sequenz kann die Genexpression befördern oder hemmen.



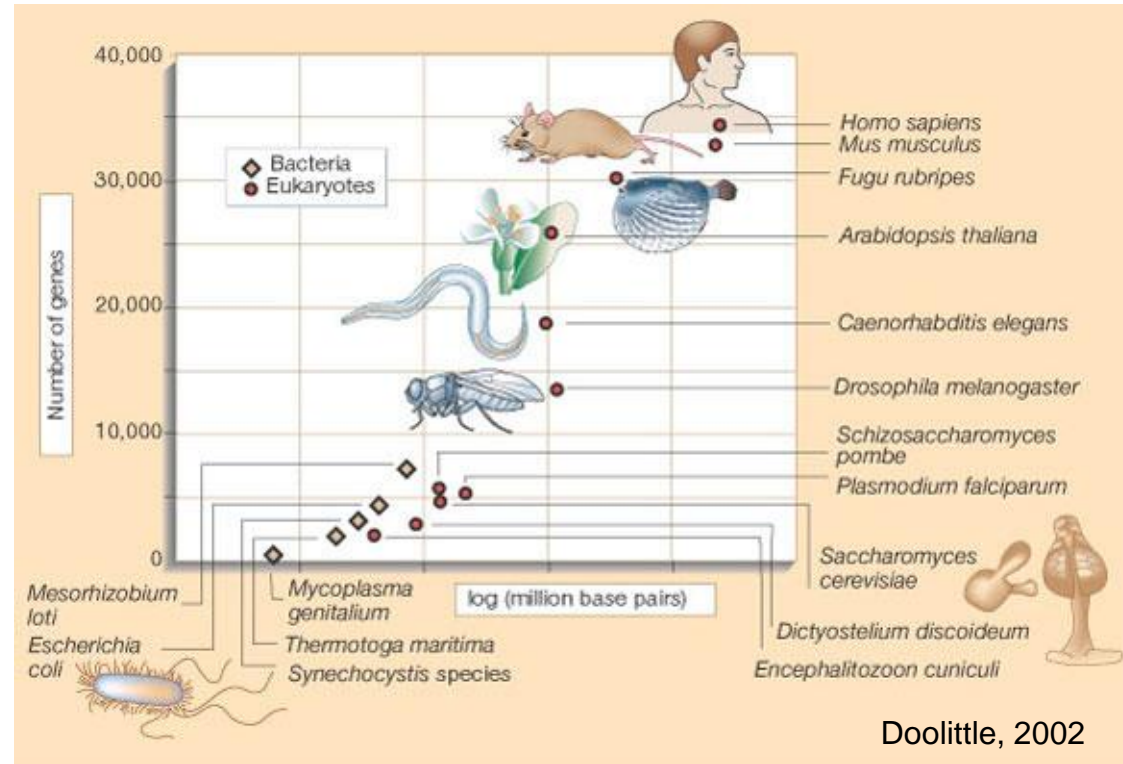
GENETISCHE VERÄNDERUNGEN (MUTATIONEN, CROSSOVER)

- Genetische Veränderungen kommen bei Replikation vor (geringe Rate, ca. $4 \cdot 10^{-10}$ pro Nukleotid pro Jahr)
- Nur die Veränderungen der Geschlechtszellen werden an Nachkommen weiter gegeben
- Crossover tauscht Genmaterial zweier homologer (in Gestalt und Abfolge der Gene übereinstimmender) Chromosomen aus



GENOMGRÖßE

- Genomgröße ist innerhalb einer Art konstant, variiert aber erheblich zwischen den Arten (C-Wert)
- Genomgröße korreliert nicht mit Komplexität des Phänotyps
- Bestandteile des Genoms
 - Kodierende DNA
 - Nichtkodierende DNA
- Typen nichtkodierender DNA
 - Repetitive Sequenzen
 - Satelliten-DNA
 - Genduplikation
- Nichtkodierende DNA kann adaptiven Wert haben
 - Pseudogene können reaktiviert werden
 - Neutrale Mutationen können Pseudogene in neue Gene verwandeln



EVOLUTIONÄRE ALGORITHMEN

- Finden Lösungen für nichttriviale Probleme, ohne viel über das Problem zu wissen
- Ähnlichkeiten zwischen natürlicher und künstlicher Evolution:
 - Phänotyp (Computerprogramm, Objektform, Schaltkreis, Roboter, etc.)
 - Genotyp (Genetische Repräsentation des Phänotyps)
 - Population
 - Diversität
 - Selektion
 - Vererbung
- Unterschiede zwischen natürlicher und künstlicher Evolution:
 - Fitness ist Maß der Lösungsgüte, die ein Individuum repräsentiert
 - Selektion anhand Fitnessfunktion
 - Verbesserung einer Anfangslösung wird erwartet

PROBLEMKLASSE VON EA: LÖSEN VON OPTIMIERUNGSPROBLEMEN

■ Gegeben:

- Ein Suchraum S
- Die zu optimierende Funktion f (Zielfunktion)

$$f : S \rightarrow \mathbb{R}$$

- Gegebenenfalls einzuhaltende Nebenbedingungen

■ Gesucht:

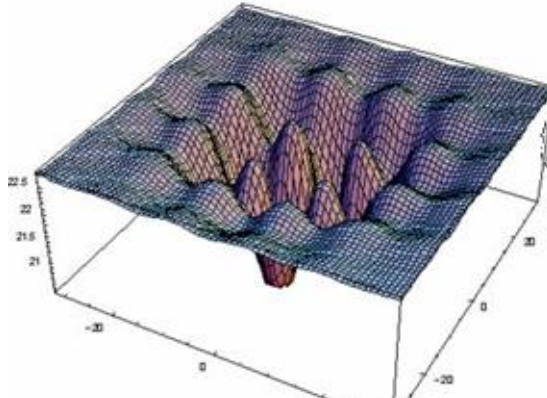
- Ein Element $s \in S$ das die Funktion f optimiert (maximiert oder minimiert)

BSP: MINIMUM DER ACKLEY-FUNKTION

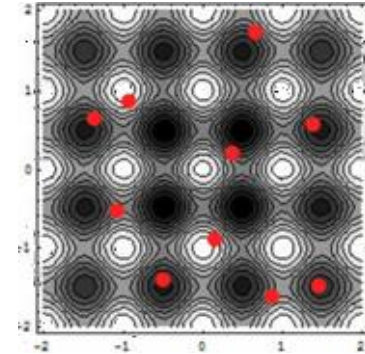
Ackley-Funktion

$$f(x, y) = -20 \exp \left(-0.2 \sqrt{\frac{1}{2}(x^2 + y^2)} \right) - \exp \left(\frac{1}{2} \cos(2\pi x) + \cos(2\pi y) \right) + 20 + e$$

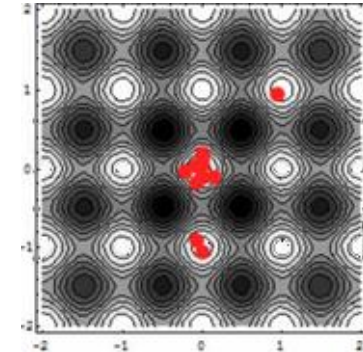
Optimum $\min(f(x,y))=f(0,0)=0$



Generation 0



Generation 20

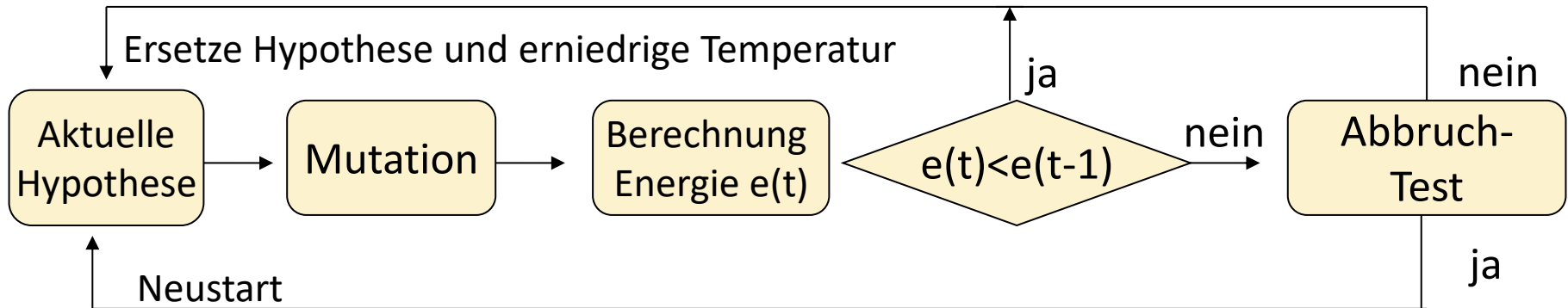
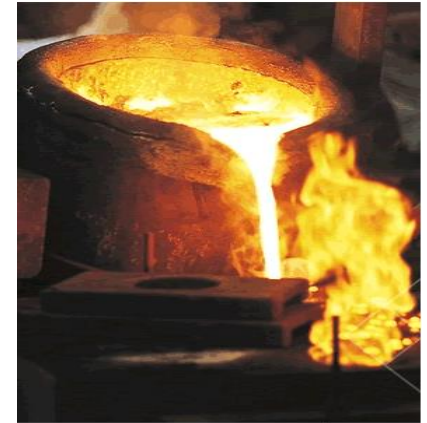


Globale vs. Lokale Optimierung

- **Globale Optimierung** versucht für ein gegebenes Problem, das insgesamt (global) beste Optimum zu finden. Dies ist schwierig wenn das Problem nicht-konvex ist.
- Neben dem globalen Optimum kann es noch weitere lokale Optima geben.
- **Lokale Optimierung** versucht in der Nähe eines Startpunktes das nächst gelegene „gute“ Extremum zu finden.
- Es ist möglich, mit einer rein lokalen Optimierung das globale Optimum zu finden, aber das ist nicht leicht festzustellen.

SIMULATED ANNEALING

- Inspiriert durch Abkühlung von Metallen
- Einzelnes Individuum wird mutiert (keine Population)
- Ziel: Finde Individuum mit minimaler Energie ($1/\text{fitness}$)



- **Stochastische Entscheidung:** $p(\text{Energy diff, Temp})$: hohe Temperatur \Rightarrow hohe Wahrscheinlichkeit auch Verschlechterungen zu übernehmen

EA-GRUNDBEGRIFFE

Begriff	Biologie	Evolutionärer Algorithmus
Individuum	Lebewesen / Organismus	Punkt im Suchraum
Genotyp	Genetische Kodierung eines Lebewesens	Kodierung eines Punkts im Suchraum
Phänotyp	Äußere Erscheinung eines Lebewesens	Dekodierung / Implementierung eines Punkts des Suchraums
Chromosom	DNS-Strang aus Nukleotiden	Vektor von Informationseinheiten
Allel	Ausprägung eines Gens	Ausprägung einer Informationseinheit
Locus	Position eines Gens	Position einer Informationseinheit
Population	Menge von Lebewesen	Menge von Punkten im Suchraum
Generation	Population zu einem Zeitpunkt	Population zu einem Zeitpunkt
Reproduktion	Erzeugung von Nachkommen durch ein oder mehrere Eltern	Erzeugung von Nachkommen durch ein oder mehrere Eltern
Fitness	Tauglichkeit eines Lebewesens	Güte eines Punktes im Suchraum

EA-GRUNDELEMENTE I

■ Kodierungsvorschrift

- Repräsentation der Individuen ist problemspezifisch
- Auswahl einer angemessenen Kodierung hat erheblichen Einfluss auf Funktion des EA

■ Initialisierungsmethode

- Erzeugt Ausgangspopulation
- Gene werden üblicherweise mit zufälligen Werten belegt
- Komplexere Probleme erfordern ggf. spezialisierte Initialisierung

■ Fitnessfunktion

- Bewertet die Individuen und spiegelt damit die „Umwelt“ wieder
- Fitnessfunktion und Zielfunktion sind häufig identisch
- Weitere Kriterien können Fitness beeinflussen, z.B. Nebenbedingungen

EA-GRUNDELEMENTE II

■ Selektionsoperator

- Wählt auf Basis der Bewertung durch Fitnessfunktion Individuen zur Reproduktion aus

■ Reproduktionsoperatoren

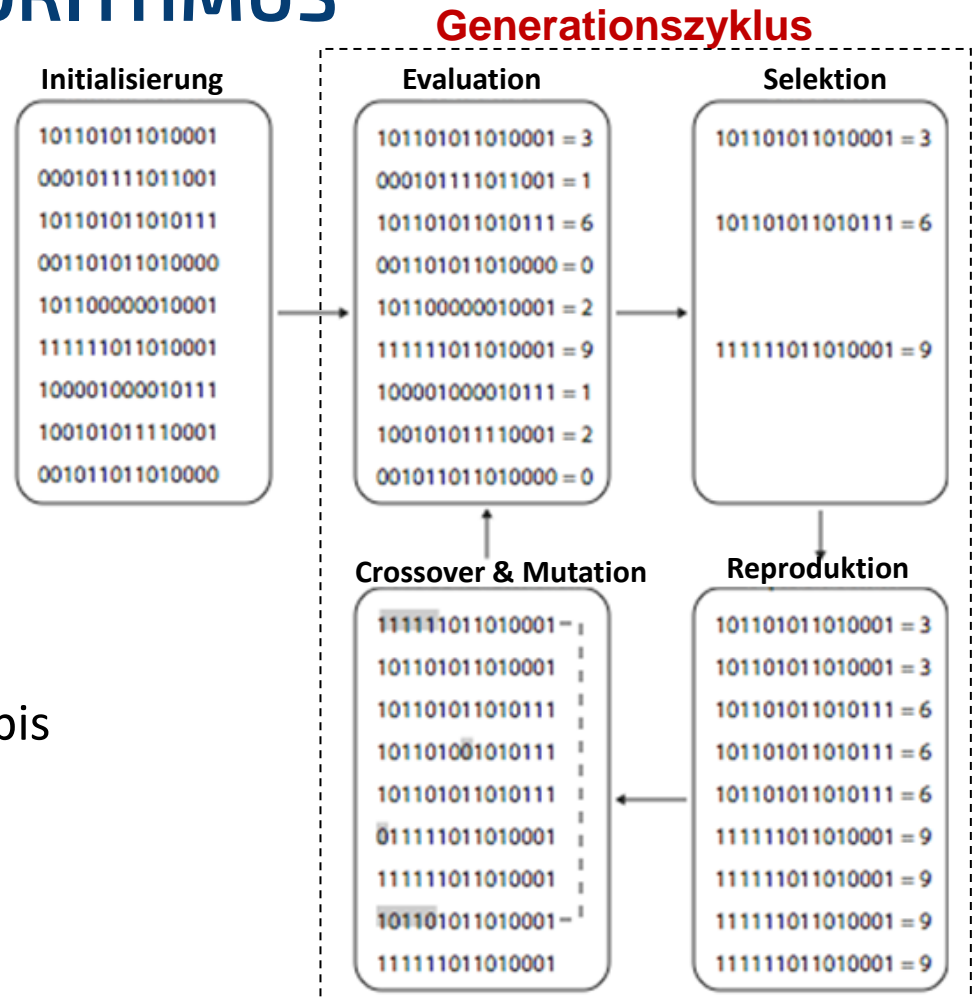
- Erzeugt Nachkommen durch Variation der genetischen Information eines oder mehrere Eltern
- Crossover: Rekombination von Chromosomen
- Mutation: Zufällige Variation einzelner Gene

■ Abbruchkriterium

- Bestimmt das Ende des EA, z.B. durch
 - Festgelegte Anzahl von Generationen erreicht
 - Festgelegte Anzahl von Auswertungen der Zielfunktion erreicht
 - Festgelegte Mindestgüte der Lösung erreicht
 - Stagnation der Optimierung
 - Zeitüberschreitung

EVOLUTIONÄRER ALGORITHMUS

- Wähle genetische Repräsentation
- Erzeuge Population
- Wähle Fitnessfunktion
- Wähle Selektionsmethode
- Wähle Crossover & Mutation
- Wähle Monitoring-Methode
- Wiederhole Generationszyklus bis Abbruchkriterium erfüllt



GENETISCHE OPERATOREN

- Genetische Operatoren werden auf einen Teil der durch die Selektion ausgewählten Individuen (Zwischenpopulation) angewendet, um Rekombinationen und Variationen der bestehenden Lösungskandidaten zu erzeugen
- Einteilung nach der Zahl der Eltern
 - Ein-Elter-Operatoren: Mutation
 - Zwei-Elter-Operatoren: Crossover
 - Mehr-Elter-Operatoren
- Operatoren müssen entsprechend der Kodierung gewählt werden
 - Kodieren Lösungskandidaten Permutationen, müssen Operatoren permutationserhaltend sein
 - Ungültige Individuen sollen nicht erzeugt werden

EIN-ELTER-OPERATOREN

- Ein-Elter-Operatoren werden als **Mutation** bezeichnet

- Standardmutation

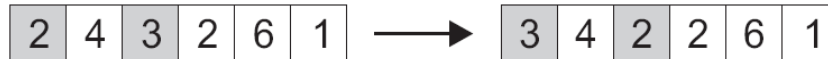
- Variiere die Ausprägung eines Gens des Chromosoms



- Eventuell werden mehrere Gene mutiert
 - Parameter: Mutationswahrscheinlichkeit: $p_m, 0 < p_m \ll 1$
 - Für Bitstrings z.B. $p_m = 1/\text{length}(x)$

- Zweiertausch

- Tausche die Ausprägung zweier Gene des Chromosoms



- Verallgemeinerung: zyklischer Tausch von 3, ..., n Genen
 - Bedingung: Allelmengen getauschter Gene sind gleich

EIN-ELTER-OPERATOREN

■ Permutation eines Teilstücks

- Mische die Gene eines Teilstücks



■ Inversion eines Teilstücks

- Drehe die Reihenfolge der Gene eines Teilstücks des Chromosoms um



■ Verschiebung eines Teilstücks

- Verschiebe einen Teil des Chromosoms an andere Stelle

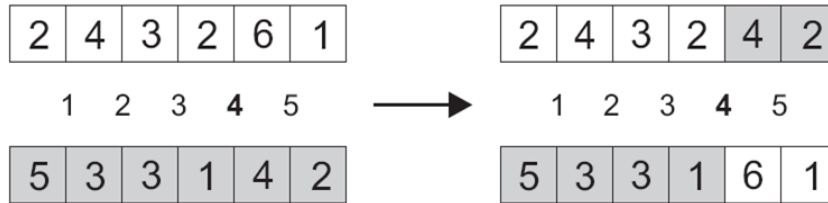


ZWEI-ELTER-OPERATOREN

■ Zwei-Elter-Operatoren werden als **Crossover** bezeichnet

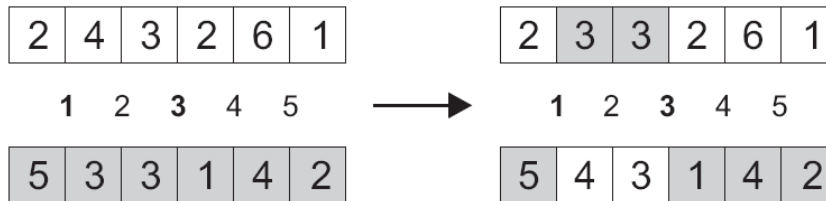
■ Ein-Punkt Crossover

- Wähle einen zufälligen Schnittpunkt
- Tausche Gene der Chromosomen ab Schnittpunkt



■ Zwei-Punkt Crossover

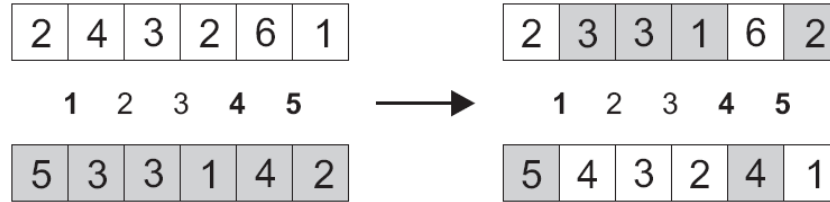
- Wähle zwei zufällige Schnittpunkte
- Tausche Gene zwischen den Schnittpunkten



ZWEI-ELTER-OPERATOREN

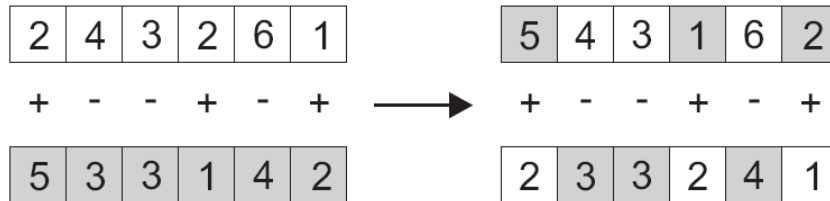
■ n-Punkt Crossover

- Wähle n zufällige Schnittpunkte
- Tausche Chromosomen nach jedem zweiten Schnittpunkt



■ Uniformes Crossover

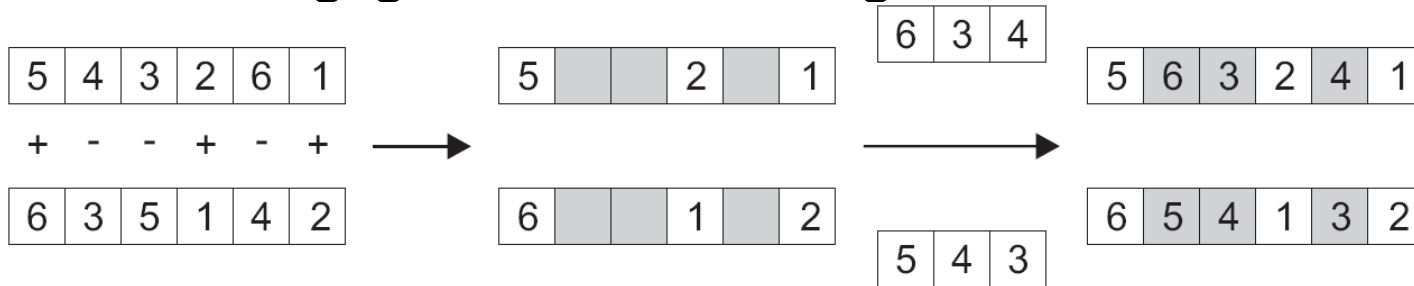
- Bestimme für jedes Gen zufällig ob Austausch stattfindet



- Beachte: Uniformes Crossover ist nicht äquivalent zu $(\text{length}(x)-1)$ -Punkt Crossover

UNIFORMES ORDNUNGSBASIERTES CROSSOVER

- Entscheide wie beim uniformen Crossover für jedes einzelne Gen, ob es übernommen werden soll oder nicht
- Fülle Lücken mit den fehlenden Allelen in der durch das andere Chromosom gegebenen Reihenfolge

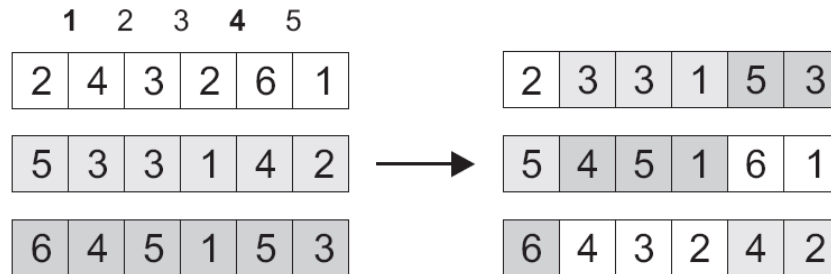


- Eigenschaften:
 - Permutationserhaltend
 - Reihenfolgeerhaltend

MEHR-ELTER-OPERATOREN

■ Diagonales Crossover

- Wähle bei n Eltern $n-1$ Crossover-Positionen
- Verschiebe die Gensequenzen zyklisch diagonal über die Chromosomen



ORTSABHÄNGIGE VERZERRUNG

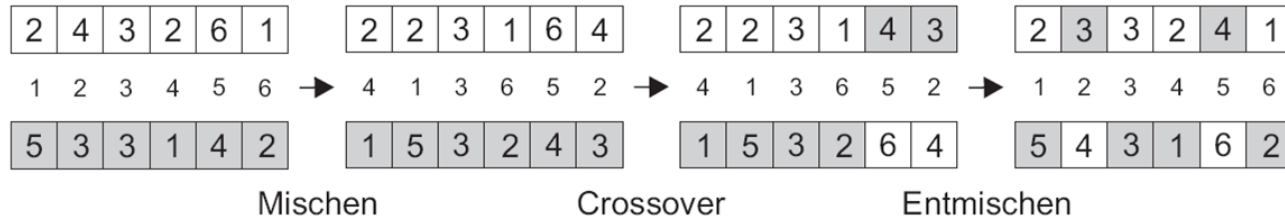
- Ist die Wahrscheinlichkeit zwei Gene zusammen zu vererben abhängig von der relativen Lage im Chromosom spricht man von ortsabhängiger Verzerrung
- Unerwünschter Effekt, da Erfolg des Evolutionären Algorithmus von Anordnung der Gene im Chromosom abhängig

Beispiel (Ein-Punkt-Crossover)

- Zwei Gene werden getrennt, wenn die Schnittposition zwischen ihnen liegt
- Offensichtlich werden weiter entfernt liegende Gene häufiger getrennt als enger benachbarte, da zwischen letzteren weniger potentielle Schnittpositionen liegen

SHUFFLE CROSSOVER

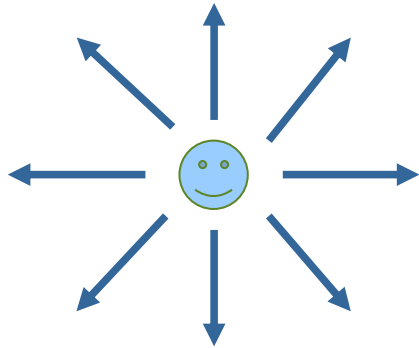
- Ein-Punkt Crossover wird auf gemischtes Chromosom angewandt, das anschließend wieder entmischt wird



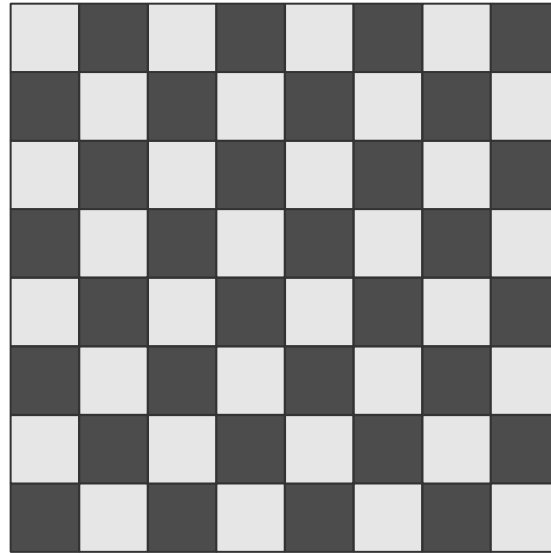
- Beachte: Shuffle Crossover ist nicht äquivalent zum uniformen Crossover. Beim Shuffle Crossover ist jede Anzahl von Gen-Vertauschungen gleichwahrscheinlich, beim uniformen Crossover ist die Wahrscheinlichkeit der Vertauschungszahl binomialverteilt mit Vertauschungswahrscheinlichkeit p_c
- Shuffle Crossover ist einer der empfehlenswertesten Crossover-Operatoren

BEISPIEL: 8 DAMEN PROBLEM

- Die Aufgabe ist es, 8 Damen auf einem Schachbrett so zu positionieren, dass diese sich nicht schlagen können.



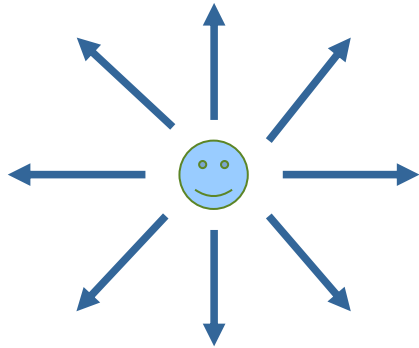
Für die Dame
erlaubte Züge



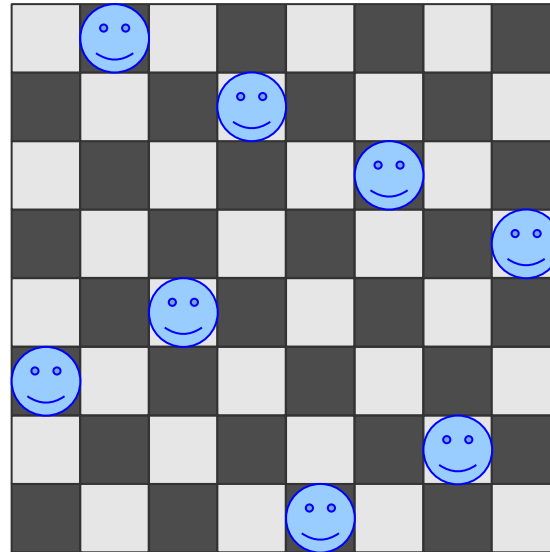
Schachbrett mit
 $8 \times 8 = 64$ Positionen

BEISPIEL: 8 DAMEN PROBLEM

- Die Aufgabe ist es, 8 Damen auf einem Schachbrett so zu positionieren, dass diese sich nicht schlagen können.



Für die Dame
erlaubte Züge



Schachbrett mit
 $8 \times 8 = 64$ Positionen

Eine von 92
Lösungen

BEISPIEL: 8 DAMEN PROBLEM

■ **Sehr naive** Implementation des Genoms:

- ein 64 Bit Binärvektor; Dame ist **1**, keine Dame ist **0**;
=> mehr als 8 gesetzte Bits=Damen sind möglich.
- Diese sehr naive Implementation erzeugt einen extrem großen Suchraum mit 2^{64} denkbaren Genomen. => Lösung schwer zu finden.



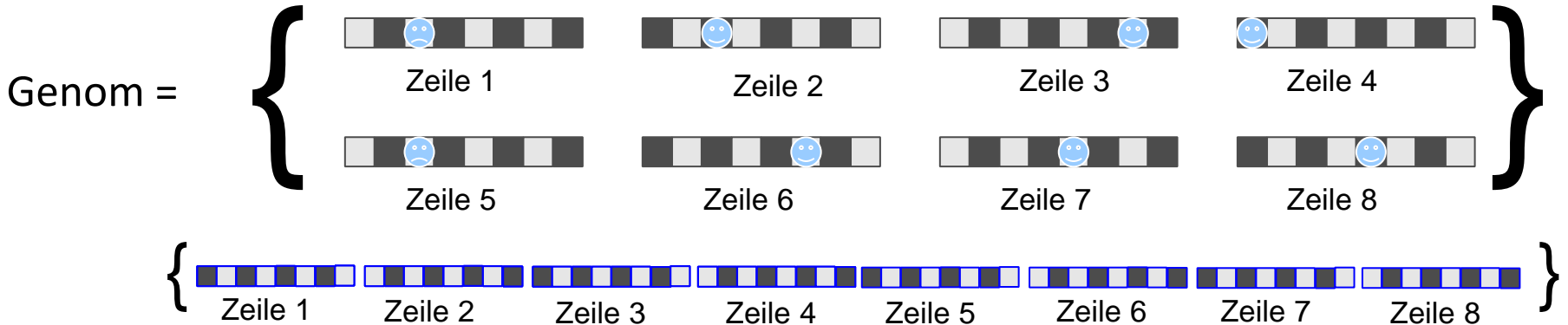
■ **Halb-naive** Implementation des Genoms:

- ein 64 Bit Binärvektor; Dame ist 1, keine Dame ist 0;
genau 8 gesetzte Bits=Damen sind Bedingung.
- Der resultierende Suchraum ist immer noch sehr groß.

BEISPIEL: 8 DAMEN PROBLEM

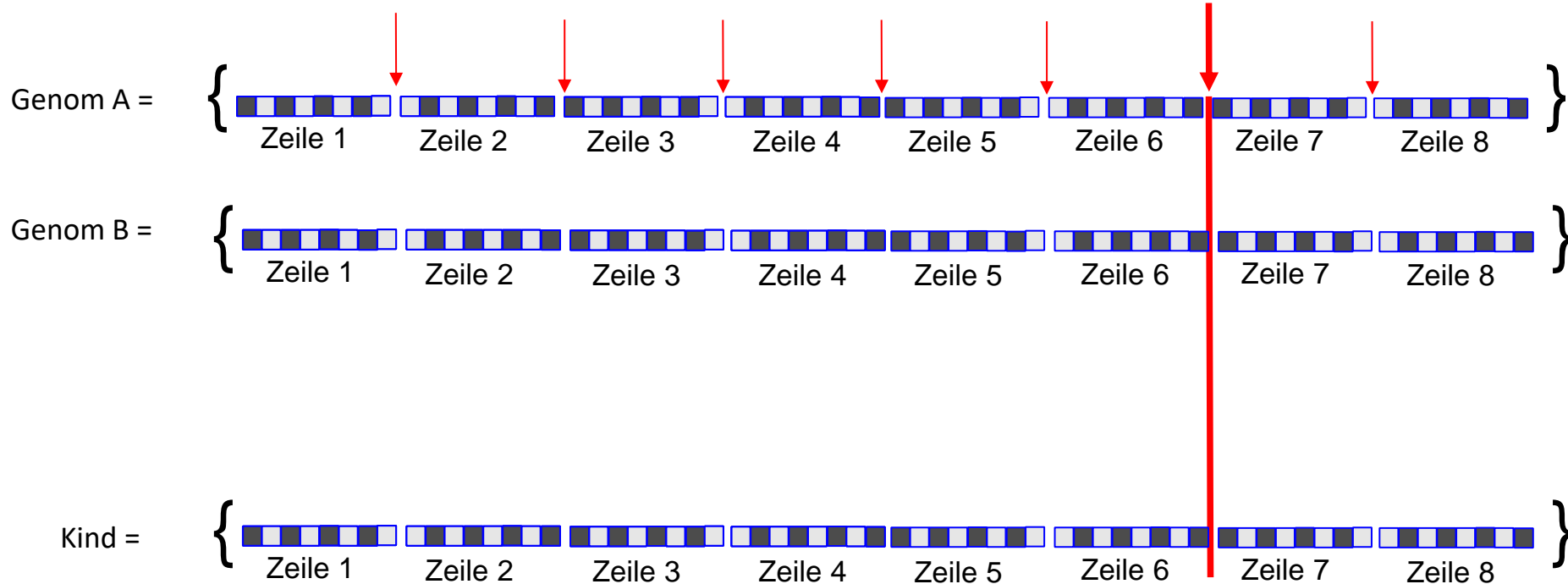
■ Mehr Vorwissen bei Kodierung:

- 8 Zeilen zu je 8 Bit; Dame ist **1**, keine Dame ist **0**;
jede Zeile enthält genau ein gesetztes Bit = eine Dame
- Das reduziert den Suchraum auf $8^8 = 16.777.216$ Möglichkeiten.
Aber: Warum nicht spaltenweise?



BEISPIEL: 8 DAMEN PROBLEM

- **Vererbung:** Rekombination, Ein-Punkt-Crossover
Crossover Punkte sind nur zwischen den Zeilen erlaubt.



BEISPIEL: 8 DAMEN PROBLEM

■ **Mutation:** die Position der Dame wird nur innerhalb einer Zeile verändert.



Wähle eine zufällige Zeile: **z** (1, ..., 8):

wähle eine zufällige neue Position (1, ..., 8) für die Dame 

Vor der Mutation

Zeile **3**



Nach der Mutation

Zeile **3**



BEISPIEL: 8 DAMEN PROBLEM

- Das Ziel ist es eine Anordnung der 8 Damen zu finden, sodass diese sich nicht gegenseitig schlagen können.
- Die **Fitness-Funktion** sollte diese Aufgabe widerspiegeln.
- ein hoher Wert für f wenn die Anordnung O.K. ist. 1
ein kleiner Wert für f wenn die Anordnung nicht O.K. ist 0
- Eine Fitness-Funktion die nur einen binären Wert liefert (0, 1) ist wenig informativ für einen Evolutionären Algorithmus.
- Die Oberfläche der **Fitness-Funktion** ist flach (0), lediglich an einigen Stellen sind isolierte Spitzen (1).
- Eine solche **Fitness-Funktion** gibt nicht wieder, dass das Genom schon dicht an einer möglichen Lösung dran sein kann.

BEISPIEL: 8 DAMEN PROBLEM

- Das Ziel ist es eine Anordnung der 8 Damen zu finden, sodass diese sich nicht gegenseitig schlagen können.
- Die **Fitness-Funktion** sollte diese Aufgabe widerspiegeln.
- Eigentlich sollte die Fitness-Funktion für Genome g , die dicht an einer Lösung liegen einen höheren Wert $f(g)$ liefern als für Genome die weiter von der Lösung entfernt sind
- Das Beste wäre es, wenn die Fitness-Funktion einen Wert $f(g)$ liefert der proportional zur Distanz zwischen dem Genom g und der optimalen Lösung g^* ist.
- Leider ist dies nicht einfach. Wenn dies möglich ist, brauchen wir keinen EA.

BEISPIEL: 8 DAMEN PROBLEM

- Vorschlag für eine **Fitness-Funktion** für das 8 Damen Problem:
- Jede Möglichkeit, dass eine Dame eine andere Dame schlagen kann wird mit **-1** gezählt.
Der Fitness-Wert **$f(g)$** ist die Summe der Angriffsmöglichkeiten.
- Dies liefert eine abgestufte Einschätzung der Situation, mit einem maximalen Wert von **$f(g^*) = 0.0$** wenn kein Angriff möglich ist.
- **Achtung:**
Diese Fitness-Funktion funktioniert nur, wenn die Zahl der Damen genau 8 ist:
weniger Damen => weniger Angriffsmöglichkeiten

LÖSUNG DURCH BACKTRACKING

Lösung des n-Damen-Problems mittels Backtracking

- ① Damen werden zeilenweise (oder spaltenweise) platziert
- ② Für jede Zeile gilt
 - In der Zeile wird die Dame der Reihe nach von links nach rechts auf die Felder gesetzt
 - Es wird jeweils überprüft, ob es zu Kollisionen mit bereits bearbeiteten Zeilen gekommen ist
 - Gab es keine Kollisionen, wird rekursiv mit der nächsten Zeile fortgefahren
 - Andernfalls wird die Dame eine Position nach rechts verschoben und erneut auf Kollisionen geprüft
 - Kann eine Dame nicht konfliktfrei in einer Zeile positioniert werden, wird auf die nächst höhere Rekursionsebene zurückgesprungen
- ③ Kann eine Dame auf der letzten Zeile des Brettes kollisionsfrei platziert werden, wird die Lösung ausgegeben.

DIREKTE LÖSUNG

- Wird lediglich *eine* Lösung gesucht, können die Positionen der Damen für alle $n > 3$ folgendermaßen berechnet werden:

- **n ist ungerade**

Setze eine Dame auf $(n - 1, n - 1)$ und verringere n um 1

- **$n \bmod 6 \neq 2$**

Setze die Damen

- in den Zeilen $y = 0, \dots, n/2 - 1$ in die Spalten $x = 2y + 1$
- in den Zeilen $y = n/2, \dots, n - 1$ in die Spalten $x = 2y - n$

- **$n \bmod 6 = 2$**

Setze die Damen

- in den Zeilen $y = 0, \dots, n/2 - 1$ in die Spalten $x = (2y + n/2) \bmod n$
- in den Zeilen $y = n/2, \dots, n - 1$ in die Spalten $x = (2y - n/2 + 2) \bmod n$

Verständnis des Problems vermeidet heuristische Suche!