

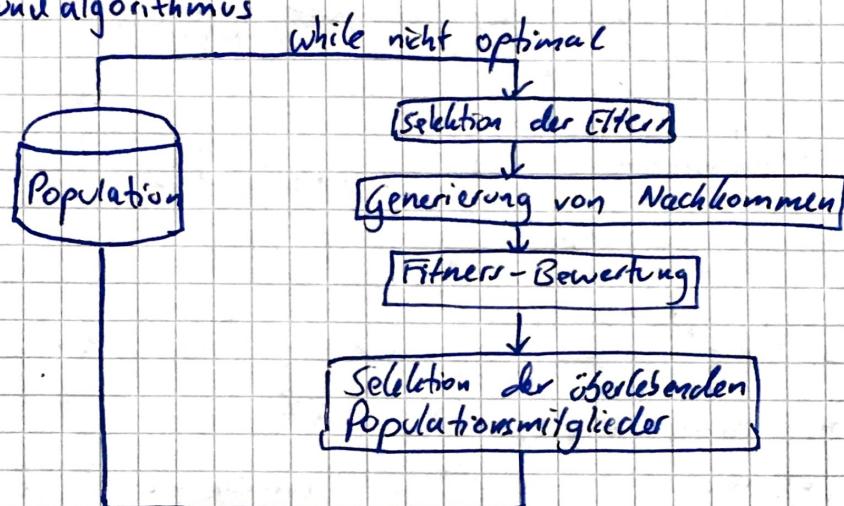
# Maschinelles Lernen 1 Evolutionäre Algorithmen

- Evolution als Optimierung komplexer, künstlicher Systeme
- Durch Zufall und Selektion hochkomplexe, an die Arbeitsumgebung angepasste, Systeme erzeugen

## Nomenklatur

- Individuum: eine mögliche Lösung (Hypothese)
- Population und Generation: Hypothesenmenge / Lösungskandidaten
- Erzeugen von Nachkommen: Generierung neuer Hypothesen, z.B. Mutation
- (Veränderter) Nachfolge: neue Hypothese
- Fitness: Hypothesengüte, zu optimierendes Kriterium (z.B. für Lösungshandlung)  
„Anatom“ ist Wert der fitness  
„Stromungswid.“
- „Selektion der Besten“: Hypothesen, die die beste Problemlösung erzeugen

## Grundalgorithmus



## Repräsentation

- Wissen wird meist strukturiert präsentiert
- dieses wird in Genen kodiert:

- $k$ -Alphabet ( $k=2$ : Binärcodierung): Genetische Algorithmen
- Vektoren reelle Zahlen: Evolutionäre Strategien
- baumartige Strukturen: Genetisches Programmieren

Bsp. Binärkodierung: wenn  $x = T$  und  $y = F$  dann  $z = T$

$$\begin{array}{ccccccccc} & & & & & & & & \\ \downarrow & & & & & & & & \\ 1 & 0 & 0 & 1 & | & 1 & 1 & 1 & 0 & | & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

## Generierung von Nachkommen

- erfolgt durch „genetische Operatoren“

- zwei Strategien: Exploration  $\Leftrightarrow$  Exploitation  
d.h. Erforschen der Hypothesenumfang  $\Leftrightarrow$  lokale Optimierung

## Vergleich:

- lokale Verbesserung bringt die Gefahr von lokalen Minima
- je stärker + zufälliger Änderungen sind, desto geringer die Wahrsch. für bessere Nachkommen
- Explorationsgrad gemäß Fitness der Generation auswählen (z.B. anfangs hoch dann fallend)

## Mutation (Exploration)

- Mutation einzelner (Eltern-)Gene

(2)

Bsp.: Bitinversion

Elternteil: 1010110

Nachkomme: 1011110

### Konzepte:

- Alle Bits einer Sequenz werden unabhängig voneinander mit bestimmter Wahrsch. invertiert
- Für eine bestimmte/zufällige Anzahl von Bits werden die Indizes zufällig ausgewählt
- Stochastisch, wobei  $z$  eine Zufallsvariable z.B. nach  $N(0, \sigma)$  um 0 mit Varianz  $\sigma$ :  $x_i := x_i + z$

### Mit Mutationsoperator:

- Herausnehmen einer Teilsequenz, einfügen an anderer Stelle
- Invertierter Einfügen der Teilsequenz

Bsp.: Sequenz-Inversion

Elternteil:

 $x_1 \dots x_i \underline{x_{i+1} \dots x_j} x_{j+1} \dots x_n$ 

Nachkomme:

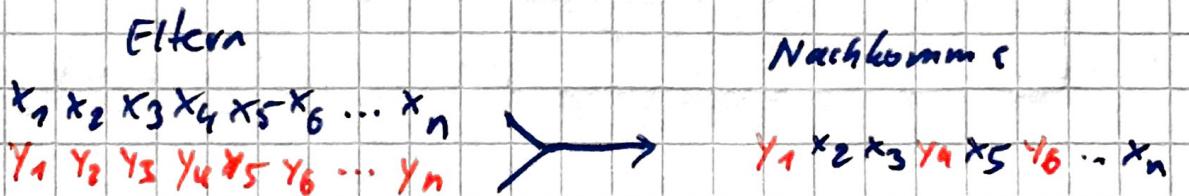
 $x_1 \dots x_i \underline{x_j \dots x_{i-1}} x_{j+1} \dots x_n$ 

- anwendungsspezifische Operatoren

## Rekombination (Exploitation)

Idee: Eigenschaften von zwei oder mehreren Eltern sollen gemischt werden

### • Diskrete Rekombination



### • Intermediäre Rekombination

Sei Elternteil<sub>1</sub> :=  $x$  und Elternteil<sub>2</sub> :=  $y$ , dann ist Nachkomme  $z$  definiert durch

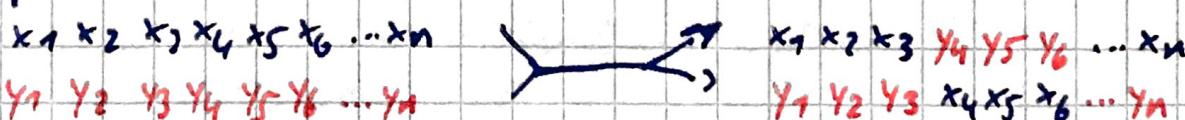
$$z_i := (x_i + y_i)/2$$

### • Crossover - aus 2 Eltern -> 2 Nachkommen

#### - Singlepoint Crossover



#### - Twopoint Crossover



#### - Uniform Crossover

### ③ Maschinelles Lernen 1 Evolutionäre Algorithmen

#### Selektion

2 Arten der Selektion:

(1)

- Selektion der Eltern für jeweilige Erzeugung von Nachkommen (Mating)

(2)

- Selektion der Population für die nächste Iteration

#### Probleme:

- Genetischer Drift:

→ Individuen vermehren sich zufällig mehr als andere

- Crowding / Ausreißerproblem:

→ „fitte“ Individuen und ähnliche Nachkommen dominieren die Population

#### Folgen:

→ Entwicklung der Individuen wird verlangsamt

→ Vielfalt der Population wird eingeschränkt

#### Lösung:

- Verschiedene Populations- und Selektionsmodelle

- Populationsgröße optimieren

#### Populationsmodelle

- Inselmodell (lokal): Evolution verläuft weitgehend getrennt, nur manchmal werden Individuen ausgetauscht.

- Nachbarschaftsmodell (nahe Umgebung): Nachkommen dürfen nur von Individuen erzeugt werden, die in ihrer Nachbarschaft die beste Fitness aufweisen

- Eine einfache Menge (global): die global Besten entwickeln sich rasch weiter, andere Entwicklungslinien werden unterdrückt.

#### Größer Ablauf von evolutionären Algorithmen

(Schleife bis Abbruchkriterium\* erreicht)

1. Initialisierung: die erste Generation von Lösungskandidaten wird (meist zufällig) erzeugt.

2. Evaluation: jedem Lösungskandidaten (LK) der Generation wird entsprechend seiner Güte ein Wert der Fitnessfunktion zugewiesen

(Evolution) 3. Durchlaufe folgende Schritte, bis Abbruchkriterium erfüllt:

(1) 1. Selektion: Auswahl von Individuen für die Rekombination

2. Rekombination: Kombination der ausgewählten Individuen

3. Mutation: Zufällige Veränderung der Nachkommen

4. Evaluation: (wie oben)

5. Selektion: Bestimmung einer neuen Generation (2)

(ersetzt „neat-fit“ Individuen mit neuen Individuen)

\* mögliche Abbruchkriterien:

(Fitness > X) oder (Generationen > X)

oder (keine Verbesserung der Fitness seit X Generationen)

## Populationsmitglieder

(4)

- Populationsgröße: - Soll sie konstant bleiben? ( $\mu$ )  
 - Wie viele neu erzeugte Nachkommen? ( $\lambda$ )  
 - Wie viele Eltern verwenden? ( $p$ )  
 - Wie werden diese bestimmt?

## Mitgliedersélection:

Stochastisch ausgewählt  $\Rightarrow$  die besten  $\mu$  Individuen

- ( $\mu, \lambda$ ) Strategie: Auswahl bezieht sich nur auf die Nachkommen
- ( $\mu + \lambda$ ) Strategie: Auswahl bezieht auch Eltern mit ein  
(die Besten werden berücksichtigt, Suche nach Eliten  $\rightarrow$  Exploitation, günstig bei gut berechenbaren Fitnessfunktionen)

## Ersetzungsregel für Mitglieder:

- Nachkommen ersetzen alle Eltern
- " " " einen Teil der Eltern
- " " " Eltern, die ihnen am ähnlichsten sind
- usw.

Daumenregel: Das beste Viertel der Population sollte drei Viertel der Nachkommen erzeugen

## Selektionsmethoden

• Fitness Based Selection:  $P(x) = \frac{f(x)}{\sum_{x' \in \text{Pop.}} f(x')}$  genau:  $P(x) = \frac{\lambda}{\mu} \cdot \frac{f(x)}{\sum_{x' \in \text{Pop.}} f(x')}$

$P(x)$ : Wahrsch. der Auswahl von Individuum  $x$   
 $\lambda$ : Anzahl von Nachkommen  
 $\mu$ : Populationsgröße  
 $f$ : Fitnessfunktion

• Ranking Based Selection:  $P(x) = \frac{g(r(x))}{\sum_{x' \in \text{Pop.}} g(r(x'))}$

(bessere Anpassung von Exploration/Exploitation)

$r(x)$ : ranking von  $x$  in der aktuellen Population gemäß Fitnessfunktion

$g$ : mit der Güte des Ranges monoton steigende Funktion größer 0  
 Exp.:  $g(x) = a^{-x}$   
 Hyperbolisch:  $g(x) = x^{-a}$

## Tournament Selection (Turnier):

- wähle für jedes zu erzeugende Individuum ( $n=2$ ) Individuen
- belohnen (Bewertung erhöhen) davon, das gemäß der Fitness beste Individuum
- wähle Individuen mit höchster Bewertung

# (5) Maschinelles Lernen 1 Evolutionäre Algorithmen

## Evolution

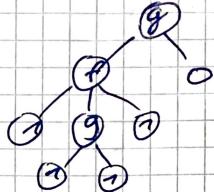
- lamarckische Evolution
  - Individuen ändern sich nach der Erzeugung
  - Genotyp (alle gene) wird verändert und anschließend vereint
- Baldwinische Evolution
- Hybride Verfahren

## Genetische Programmierung

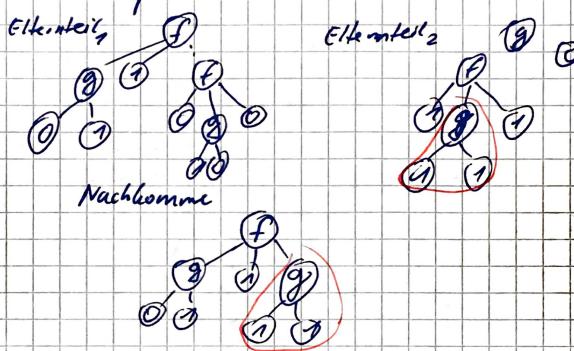
Ziel: Erstellung optimierter Programme

- Individuen sind Programme
- Repräsentation z.B. als Baum
- Selektion, Mutation und Rekombination auf Baumstrukturen
- Fitnes: Programmtest auf einer Menge von Testdaten

$$g(f(1 \dots g(1,1) \dots 1,0))$$



## Rekombination Bsp.



## Eigenschaften von EA

- ④ gute Parallelisierbarkeit (Population von Individuen)
- ⑤ meist sehr rechenintensiv
- ⑥ a priori Wissen kann bei Initialisierung integriert werden
- ⑦ Anwendungsspezifisches Wissen erfordert Anpassung allgemeiner Algorithmen

- heuristische Verfahren wie EA sind gut bei komplexen / schweren Problemen (NP-vollst.) ohne Strukturwissen
- wenn a priori Wissen zum Problem vorliegt, optimales Ergebnis gefordert wird und nicht heuristische Verfahren für das Problem existieren, sollte stets von EA abgesehen werden