Neuronale Netze mit genetischen Algorithmen

Rodion Kovalenko

Matrikelnummer 3009393

Studiengang: Master Wirtschaftsinformatik

SS 2019

Zielsetzung

- Trainieren des dreischichtigen neuronalen Netzes (NN) mit genetischen Algorithmen
- Vergleich von NN trainiert mit Backpropagation und genetischen Algorithmen (GA)
- NN für Erkennung handgeschriebene Zahlen
- Verwendung von MNIST-Datenbank mit 60.000 Trainingsmustern
- Test mit MNIST-Testdatensatz auf 10.000 Testmustern

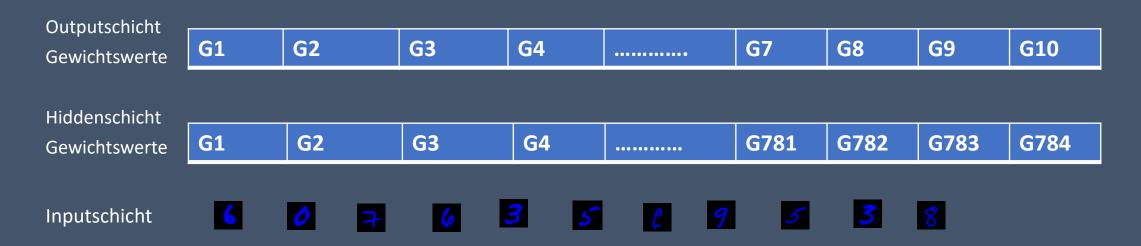
Genetischer Algorithmus

- 1. Populationserzeugung
- 2. Fitnessfunktion
- 3. Selektion
- 4. Rekombination (Crossover)
- 5. Mutation (max. 5% der Chromosomem)

Wiederhol die Schritte von 2 bis 5 bis eine passende Lösung gefunden wird.

Populationgröße

- Population besteht aus Chromosomen
- Jedes Chromosome stellt eine potenzielle Lösung dar



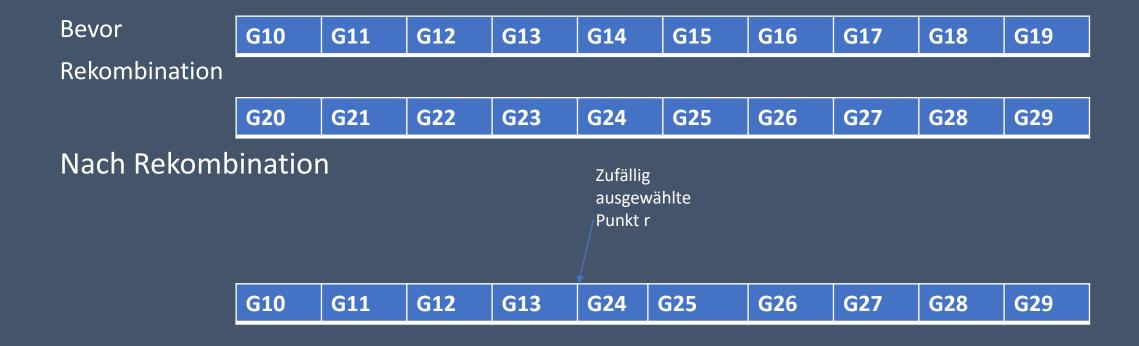
Wo G = Gewichtswert

Fitnessfuktion und Selektion

- Jede Lösung (Vektor) wird mit einem Score bewertet
- Der Score ist in unserem Fall die Zahl der richtig erkannten Muster
- Es werden zwei beste Lösungen (Eltern) für die Erzeugung neuer Population selektiert, die den kleinsten Score haben.
- Eltern bleiben solange existieren, bis eine besser Lösung gefunden wird
- Elitistische Variante
- Populationsgröße bleibt konstant

Rekombination

Neue Population wird durch Rekombination erzeugt. Es passiert sowohl für die Zwischenschicht als auch für Ausgabeschicht.



Mutation

- Nachdem mithilfe von Rekombination neue Population erzeugt wurde, werden die einzelnen Gewichtswerte (Genen) jedes Chromosoms zufällig mutiert.
- Mutationsrate wird in unserem Fall zufällig ausgewählt, hält aber in Grenzen von 1 bis 3 Prozenten.

Bevor Mutation



Nach Mutation

0.5	0.3	0.1	0.6	0.63	0.42	0.93	0.33	0.23	0.33

Bewertung

- Neuronales Netz trainiert mit GA zeigte Genauigkeit von 88 % bei 4000 Iterationen und Populationsgröße von 100.
- Neuronales Netz trainiert mit BP zeigte Genauigkeit von 94 % bei 4000 Iterationen und Lernrate von 0.8

NN trainiert mit GA	NN trainiert mit Backpropagation
Accuracy: 88,0 %	Accuracy: 94,0 %
Precision:	Precision:
class 1: 91,4 %	class 1: 96,1 %
class 2: 95,3 %	class 2: 96,5 %
class 3: 90,2 %	class 3: 87,5 %
class 4: 85,7 %	class 4: 81,9 %
class 5: 84,7 %	class 5: 86,7 %
class 6: 75,3 %	class 6: 72,6 %
class 7: 88,9 %	class 7: 92,0 %
class 8: 93,6 %	class 8: 91,4 %
class 9: 87,0 %	class 9: 87,6 %
class 10: 84,8 %	class 10: 85,3 %

Schlussfolgerungen Ausblick

- Rechnerkapazitäten und Laufzeit
 - Je größer die Population desto länge dauert das Training mit GA im Vergleich zum Training mit Backpropagation
- Man erreicht ähnliche Ergebnisse mit GA und Backpropagation
- Das Training mit Backpropagation liefert bessere Ergebnisse auch wegen der einfachen Struktur.

Ausblick:

- GA konnte für das Training von NN eigenständig verwendet werden
- oder alternativ eingesetzt werden, wo das Training mit Backpropagation die Genauigkeit des Netzes nicht verbessert.
- Somit kann man die Vorteile verschiedener Trainingsansätze kombinieren