

Suchalgorithmen im Machine Learning

Optimierungsverfahren für Ingenieure

Jan Hoegen

2. Dezember 2025

Hochschule Karlsruhe
University of Applied Sciences

1. Problemstellung
2. Lösung mit klassischen Algorithmen
3. Lösung mit Bayesian Search
4. Vergleich und Fazit

Problemstellung

Ein einfaches Beispiel

Blackbox b :

Eingabe: Bild

Ausgabe: Klasse (Hund, Katze, Auto, ...)

Ergebnis jedoch nicht immer korrekt

$$b(\text{Bild}) \rightarrow \text{Klasse}$$

Training der Blackbox:

Parameter von b werden verbessert, wenn das Ergebnis falsch ist

Größe: 500 000 Parameter

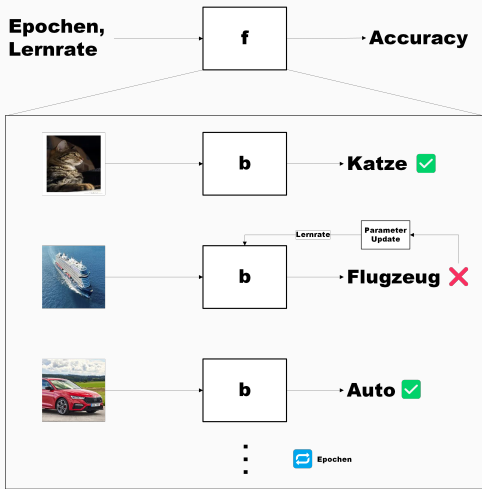
Datensatz: 60 000 Bilder

Zielfunktion $f \rightarrow$ Accuracy:

Prozess: Trainiert die Blackbox b
über alle Bilder

Ausgabe: Genauigkeit von b nach
dem Training

Genauigkeit der Blackbox als Optimierungsziel



$f(\text{Epochen, Lernrate}) \rightarrow \text{Accuracy}$

Prozess: Trainiert die Blackbox b über alle Bilder

Ausgabe: Genauigkeit von b nach dem Training

Eingaben:

Epochen: Wie häufig wird der gesamte Datensatz durchlaufen

Lernrate: Stärke der Parameteranpassung

Abbildung 1: Darstellung der Zielfunktion.

Zielsetzung der Optimierung

Ziel:

Werte für Epochen und Lernrate finden, um beste Accuracy zu erhalten

Problem:

Jede Berechnung von f dauert sehr lange (Minuten bis Stunden)

Konsequenz:

Mit lediglich 25 Berechnungen von f das beste Ergebnis finden

Effiziente Suchstrategien nutzen

Lösung mit klassischen Algorithmen

Grid Search

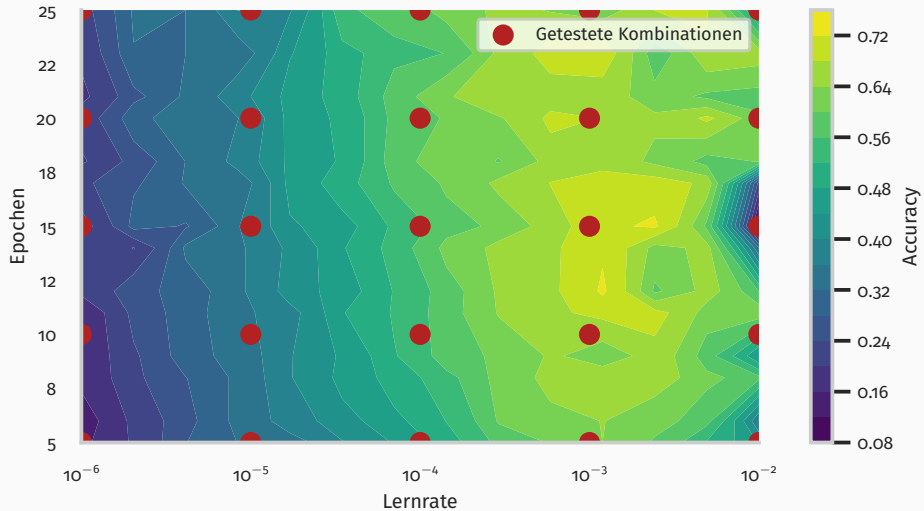


Abbildung 2: Accuracies mit Grid Search Algorithmus

Random Search

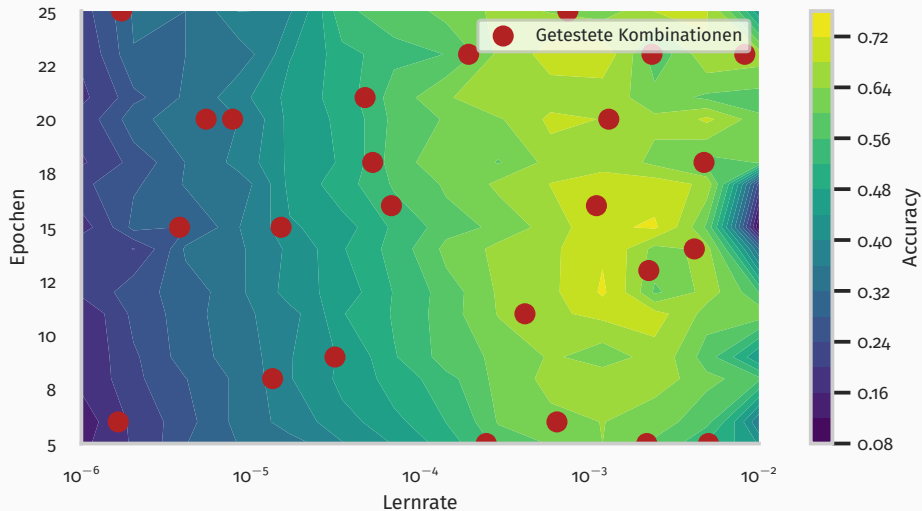


Abbildung 3: Accuracies mit Random Search Algorithmus

Lösung mit Bayesian Search

Die Idee von Bayesian Search

Modell: $m(\text{Epochen}, \text{Lernrate}) \rightarrow \text{Accuracy}$

Näherung der Zielfunktion $f(\text{Epochen}, \text{Lernrate})$

schnell auswertbar

Wird nach jeder neuen Auswertung von f weiter verbessert

Exploitation: Vorhandenes Wissen nutzen

Wähle Punkte, an denen m eine hohe Accuracy vorhersagt

f an dieser Stelle berechnen \rightarrow vermutlich neuer Bestwert

Exploration: Unbekannte Bereiche ausprobieren

Wähle Punkte, an denen m und f stark voneinander abweichen könnten

Verbessert das Modell und entdeckt neue gute Bereiche

Abbildung 4: Bayes'sche Scoreoptimierung für einen Random-Forest-Klassifizierer

Quelle: [1]

Legende:

x-Achse: Parameter des Random-Forest-Klassifizierers.

Schwarz: Zielfunktion.

Lila: Modell mit Unsicherheitsbereich.

Expected Improvement: Erwarteter Zugewinn gegenüber dem aktuellen Bestwert.

Upper Confidence Bound: Finde vielversprechende, aber unerkundete Bereiche.

Probability of Improvement: Wahrscheinlichkeit, einen besseren Punkt zu finden.

Eregbnisse mit Bayesian Search

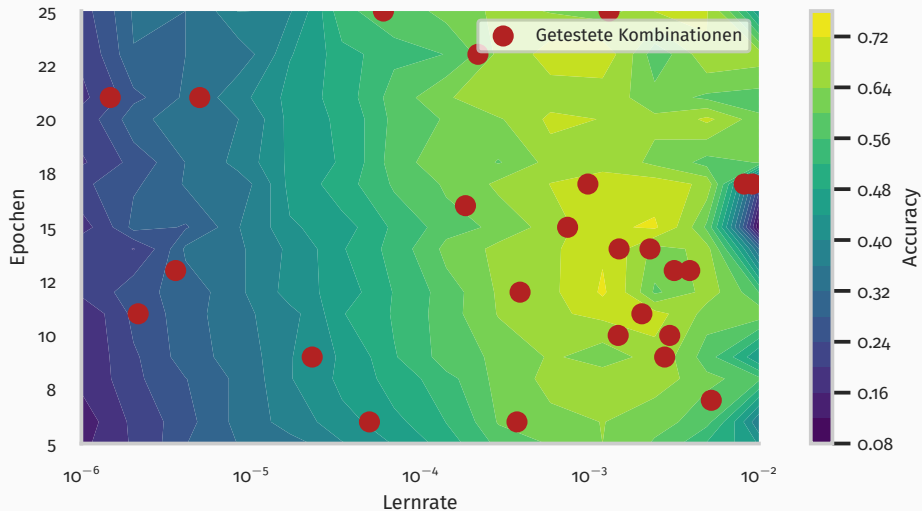


Abbildung 5: Accuracy Ergebnisse für Bayesian Search

Vergleich und Fazit

Vergleich der Suchalgorithmen

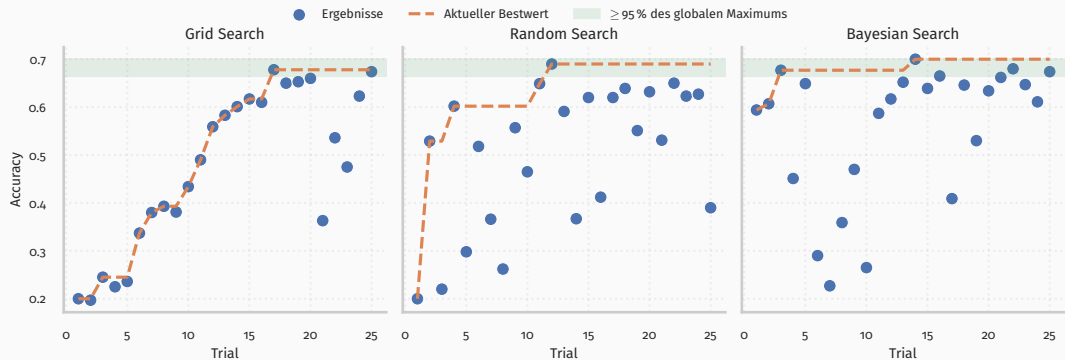


Abbildung 6: Geschwindigkeit und Genauigkeit der Suchalgorithmen

Fragen?

- [1] AnotherSamWilson, „**Bayesian optimization of a function with a Gaussian process**“, besucht am 27. Okt. 2025. Adresse:
<https://github.com/AnotherSamWilson/ParBayesianOptimization>