Wachstum vorhersagen (Exponentielle Regression)

Kapitel online

Wenn etwas immer schneller wächst: wie tiefe Netze komplexe exponentielle Zusammenhänge nachbilden.

Fortgeschritten · Kapitel 20 von 50 · ca. 25 Min. · 250 XP

Einfach gesagt: Zehn Blattläuse werden in einer Woche über tausend — für so eine Explosion stapelst du erstmals mehrere Zwischenschichten übereinander.

Stell dir vor, du entdeckst **10 Blattläuse** an deiner Tomatenpflanze. „Ach, nur zehn“, denkst du. Drei Tage später sind es 80. Nach einer Woche über tausend. Willkommen beim **exponentiellen Wachstum** — derselben Mathematik, die auch virale Posts und Zinseszins explodieren lässt. Und die ist für ein flaches Netz richtig schwer. Heute baust du deshalb zum ersten Mal **mehrere Hidden Layers übereinander**: ein echtes *Deep* Neural Network.

Die mathematische Beziehung:

```
Schädlinge(t) = Startanzahl × Vermehrungsrate^Tage
```

Beispiel: 10 Blattläuse × $2{,}0^3$ = 80 Blattläuse nach 3 Tagen — keine Gerade, keine sanfte Kurve, sondern eine Explosion. Genau daran scheitern einfache Modelle.

Was du hier lernst

- Wie man **mehrere Hidden Layers** stapelt — und warum „Deep“ genau das heißt
- Warum tiefe Netze **komplexere Muster** lernen können (Feature-Hierarchie: einfach → abstrakt)
- Wie du eine **Architektur wählst** (wie viele Layer? wie viele Neuronen?)
- Wie viele **trainierbare Zahlen** so ein Netz wirklich hat (gleich zählen wir nach!)

Für den Unterricht

Einstieg (5 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (15 Min): die interaktiven Module auf herrki.de/210_exponentielle_regression ausprobieren.

Sichern mit Quiz (5 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Dein Werkzeugkasten bis hierher — alles kommt heute zusammen

Dieses Kapitel erfindet fast nichts neu — es **kombiniert alles, was du schon kannst**, zu einem größeren Netz. Kurzer Blick in den Rucksack:

Neuronale Netze

Du kannst schon:

- Tensoren sind Zahlenpakete
- Neuronen transformieren sie
- Formel: `y = x × w + b`

Heute:
[3] Input → mehrere Layer → [1] Output

Lineare Regression

Du kannst schon:

- Ein Neuron lernt eine Gerade
- Training = raten, messen, nachjustieren

Heute:
Exponentiell — mit einer Geraden chancenlos!

⚙

Multiple Regression

Du kannst schon:

- Mehrere Eingaben gleichzeitig
- Lernrate = Schrittgröße
- Adam = cleverer Optimizer

Heute:
3 Inputs — und wir drehen an den Hyperparametern!

Polynomiale Regression

Du kannst schon:

- Hidden Layer + ReLU = Kurven lernen
- Training vs. Inferenz

Heute:
Nicht EIN Hidden Layer — ZWEI übereinander!

Was ist heute wirklich neu?

- **3 Inputs statt 1:** Startanzahl, Vermehrungsrate, Tage → [3]
- **2 Hidden Layers:** 64 und 32 Neuronen für verschachtelte Muster
- **Exponentielle Funktion:** `y = a × b^c` — die härteste Kurve bisher
- **Mehr Trainingsdaten:** 500 simulierte Befalls-Experimente

Runter zum Tensor: Netzwerk-Architektur und Tensor-Fluss

Verfolge den Weg der Zahlen — vom Blattlaus-Tensor bis zur Vorhersage. Das ist der **rote Faden bis ganz nach unten**: Jede Station ist nur „Tensor rein, Tensor raus“.

Tensor-Fluss durch das Netzwerk

Input Layer

[3]

3 Features:
• Startanzahl
• Vermehrungsrate
• Tage

→

⋮

Hidden Layer 1

[64]

64 Neuronen (ReLU)
erkennt einfache Muster

→

⋮

Hidden Layer 2

[32]

32 Neuronen (ReLU)
kombiniert zu abstrakten Mustern

→

Output Layer

[1]

Anzahl Schädlinge
nach x Tagen

Tensor-Transformationen, Schritt für Schritt:

1. **Input:** [3] → `[10, 2.0, 5]` (Startanzahl, Rate, Tage)
2. **Hidden Layer 1:** [3] → [64] via 192 Gewichte
3. **ReLU:** Negative Werte → 0 (die Nicht-Linearität!)
4. **Hidden Layer 2:** [64] → [32] via 2.048 Gewichte
5. **ReLU:** noch einmal nicht-linear verbiegen
6. **Output Layer:** [32] → [1] via 32 Gewichte
7. **Ergebnis:** ≈ 320 Schädlinge — die Explosion ist vorhergesagt!

Gesamt (mit Bias-Werten): 192 + 64 + 2.048 + 32 + 32 + 1 = **2.369 trainierbare Zahlen** — und JEDE einzelne wird beim Training per „raten, messen, nachjustieren“ eingestellt.

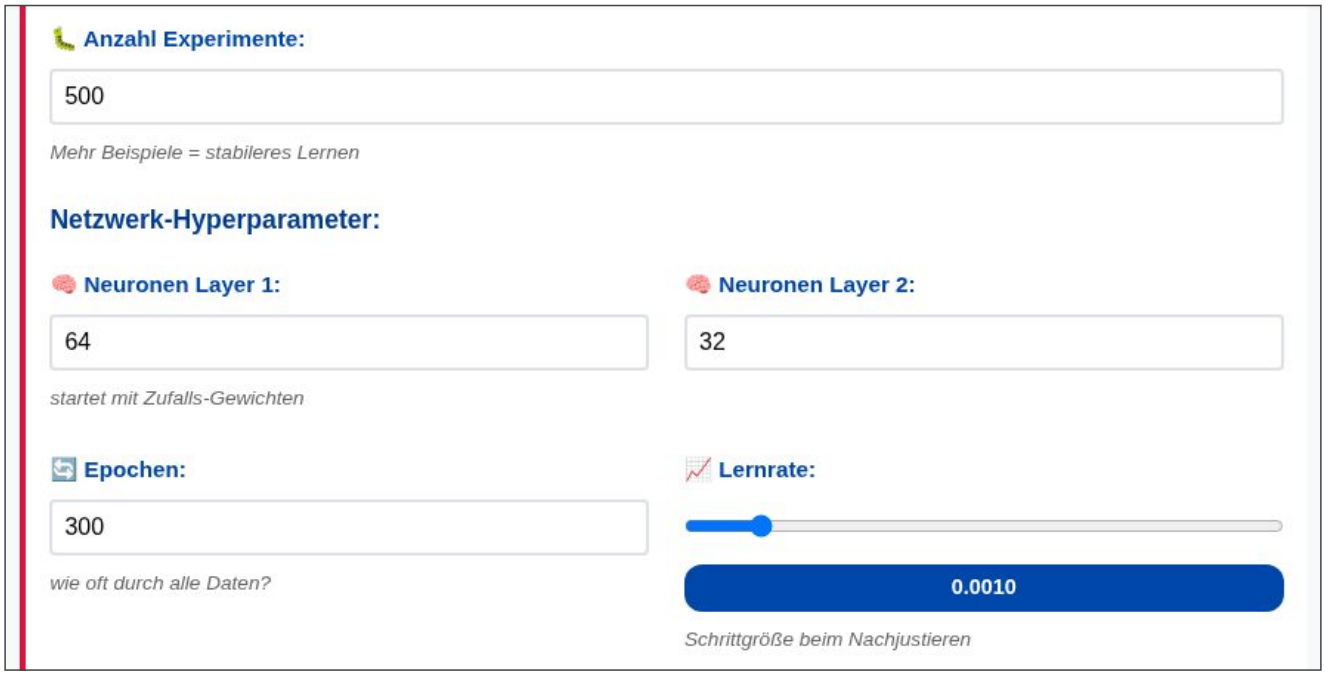
Experimentier-Bereich: Hyperparameter anpassen

Jetzt bist du die Architektin: Wie tief, wie breit, wie schnell? Dreh an den Reglern und beobachte, was mit dem Lernerfolg passiert:

Trainingsdaten-Parameter:

Am Gerät ausprobieren: herrki.de/210_exponentielle_regression#abschnitt-4

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Mehr Beispiele = stabileres Lernen

Netzwerk-Hyperparameter:

startet mit Zufalls-Gewichten

wie oft durch alle Daten?

0.0010 Schrittgröße beim Nachjustieren

Forschungsaufträge (probier sie wirklich aus!):

- Teste: 10 Blattläuse, Vermehrungsrate 2,0 — wie viele nach 5 Tagen? (Kopfrechnung: $10 \times 2^5 = 320$. Trifft dein Netz das?)
- ⚙ Architektur-Duell: 16 vs. 128 Neuronen in Layer 1 — lohnt sich „größer“ immer?

- Lernraten-Extreme: 0,0001 (Schneckentempo) vs. 0,01 (Vollgas) — was geht schief?
- 50 vs. 300 Epochen — ab wann wird das Modell kaum noch besser?

0%

Trainingsdaten (Erste 10 von X)
Modell-Struktur
Trainings-Ergebnis
Eigene Vorhersage machen (Inferenz) Simuliere deinen eigenen Schädlingsbefall: Wie schlimm wird es nach X Tagen? **Inferenz heißt:** nur noch Forward Pass — die 2.369 gelernten Zahlen bleiben unverändert, es wird nur noch durchgerechnet.

Zurück zur Anwendung

Zurück zu den zehn Blattläusen auf deiner Tomatenpflanze: Genau diese Kurve — erst harmlos flach, dann plötzlich senkrecht — hat dein Netz eben aus 500 Beispielen gelernt. Und weil dieselbe Mathematik hinter Zinseszins, viralen Posts und Infektionszahlen steckt, ist ein Modell, das exponentielles Wachstum sauber trifft, weit mehr wert als eine Schädlingszählung. Denk dabei an die dritte Quizfrage: Innerhalb der gesehenen Tage ist die Vorhersage stark, weit dahinter wird sie zur Fantasie — wer eine Zahl für Tag 100 verlangt, bekommt geraten statt gerechnet.

Bis hierher hat dein Netz immer eine *Menge* ausgerechnet: wie viele Läuse, wie viel Ertrag, wie viel Umsatz. Im nächsten Kapitel bekommt es eine ganz andere Aufgabe — es soll sich **entscheiden**: „Ja oder Nein? Binäre Klassifikation“.

Zum Weiterlesen

- **Warum „tief“ überhaupt hilft:** Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Kapitel 6 „Deep Feedforward Networks“ (frei online) — deeplearningbook.org/contents/mlp.html
- **Warum tiefe Netze schwer zu trainieren sind:** Michael Nielsen, *Neural Networks and Deep Learning*, Kapitel 5 — neuralnetworksanddeeplearning.com/chap5.html
- **Zum Selberdrehen:** TensorFlow Playground — Schichten und Neuronen im Browser hinzufügen und live zusehen, was das mit der gelernten Kurve macht — playground.tensorflow.org

Kurzer Check – mit Lösung

Lösungen für Lehrkräfte – für Lernende die Online-Version nutzen.

Frage 1: Kopfrechnen wie im Kapitel: 10 Blattläuse, Verdopplung pro Tag — wie viele nach 5 Tagen?

- 100
- ✔ **320**
- 50

$10 \cdot 2^5 = 10 \cdot 32 = 320$. Genau diese explosive Kurve (erst flach, dann steil) muss dein Netz aus Beispielen nachbauen.

Frage 2: Warum braucht exponentielles Wachstum ein tieferes Netz als die Parabel aus dem letzten Kapitel?

- Weil tiefe Netze grundsätzlich schneller rechnen.
- ✔ **Die Kurve krümmt sich über riesige Wertebereiche immer stärker — mehrere Schichten kombinieren einfache Knicke zu solch steilen Formen.**
- Weil Blattläuse mehr Features haben als Pflanzen.

Schicht für Schicht werden aus einfachen Knicken zusammengesetzte, immer ausdrucksstärkere Formen — dein Netz hier verrechnet dafür 2.369 gelernte Zahlen.

Frage 3: Trainiert wurde mit Tag 1–15. Jemand fragt nach Tag 100 — wie gut ist die Vorhersage?

- Genauso zuverlässig wie im Trainingsbereich.
- ✔ **Unzuverlässig: Außerhalb der gesehenen Daten (Extrapolation) rät das Netz nur noch.**
- Das Netz gibt dann automatisch 0 zurück.

Wichtige Ehrlichkeits-Regel: Ein Netz kennt die Welt nur aus seinen Beispielen. Je weiter du dich davon entfernst, desto mehr wird die Antwort zur Fantasie — merk dir das fürs Kapitel über Halluzinationen!

Das nimmst du mit

- **„Deep“ heißt gestapelt.** Zwei Hidden Layers (64 und 32 Neuronen) statt einem — Schicht für Schicht werden aus einfachen Knicken immer ausdrucksstärkere Formen.
- **Exponentielles Wachstum ist die härteste Kurve bisher.** 10 Blattläuse × 2^5 = 320 nach fünf Tagen — erst flach, dann Explosion. Daran scheitern flache Netze.
- **Der Tensor-Fluss bleibt langweilig einfach.** [3] → [64] → ReLU → [32] → ReLU → [1] — jede Station nur „Tensor rein, Tensor raus“.
- **2.369 trainierbare Zahlen** stecken in diesem kleinen Netz — und jede einzelne wird per „raten, messen, nachjustieren“ eingestellt.
- **Außerhalb der Trainingsdaten rät das Netz.** Tag 1–15 gesehen, Tag 100 gefragt: Extrapolation ist keine Vorhersage.

Stark! Du kannst jetzt Zahlen vorhersagen — sogar explodierende.

Als Nächstes lernt dein Netz etwas ganz anderes: **Entscheidungen treffen** — weiter zur binären Klassifikation.

So zitierst du dieses Kapitel

HerrKI: Wachstum vorhersagen (Exponentielle Regression). herrki.de/210_exponentielle_regression, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.