Auswendiglernen ist kein Verstehen (Overfitting & Regularisierung)

Kapitel online

Wenn Netze auswendig lernen statt zu verstehen — und was Dropout & Co. dagegen tun (Overfitting, Regularisierung).

Fortgeschritten · Kapitel 24 von 50 · ca. 25 Min. · 250 XP

Einfach gesagt: Wer jede Übungsaufgabe auswendig lernt, scheitert in der echten Prüfung — genau das droht auch deinem Netz, wenn niemand gegensteuert.

Kennst du den Mitschüler, der vor der Arbeit alle Übungsaufgaben *auswendig* gelernt hat? In der Übung: fehlerfrei. In der Klassenarbeit mit neuen Aufgaben: Blackout. Genau das passiert neuronalen Netzen ständig — es heißt **Overfitting**: 99 % richtig auf den Trainingsdaten, 60 % auf neuen Daten. Heute lernst du die Gegenmittel, die jedes Produktiv-Modell braucht.

Direkt anknüpfend: In „Trainieren, prüfen, bestehen“ hast du gelernt, Daten in Training und Test zu trennen — und was „Schummeln“ anrichtet. Hier siehst du jetzt den Ernstfall, für den diese Trennung erfunden wurde: *Ohne Testdaten würdest du Overfitting nie bemerken* — dein Modell sähe perfekt aus und wäre draußen nutzlos.

Was du hier lernst

- Woran du **Overfitting** erkennst (Auswendiglernen statt Verstehen)
- Was **Dropout** runter zum Tensor wirklich tut: Neuronen zufällig auf 0 setzen
- Wie **L2-Regularisierung** die Gewichte klein und „bescheiden“ hält
- Warum das alles nur mit **sauber getrennten Test-Daten** überhaupt auffällt
- Das **Duell zweier Modelle** auf denselben Daten — und der Praxistest an einer neuen Schülerin

Für den Unterricht

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

Erkunden am Gerät (15 Min): die interaktiven Module auf herrki.de/240_overfitting_regularisierung ausprobieren.

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

Runter zum Tensor: Was machen Dropout und L2 wirklich?

Das „Wissen“ deines Netzes steckt in seinen **Gewichts-Tensoren** — großen Paketen gelernter Zahlen (seit „Was ist ein Neuronales Netz?“). Hier geht EINE Zahl hinein (die Lernstunden) und läuft durch 32 Neuronen [1, 32], dann durch 16 [32, 16]. Beide Gegenmittel greifen genau dort an:

1**Dropout(0.3):** In jedem Trainingsschritt werden zufällig 30 % der Neuronen kurz auf `0` gesetzt — als müsste dein Team jede Woche ohne ein paar zufällige Kollegen auskommen. So sieht das für eine Schülerin mit 15 Lernstunden aus, 6 der 32 Neuronen herausgegriffen:

```
Aktivierungen [ 0.80 1.40 0.00 2.10 0.60 1.90 ]

würfeln, 30 % raus (hier Neuron 2 und 5):
               [ 0.80 0.00 0.00 2.10 0.00 1.90 ]

Rest ×1/(1−0.3) = ×1.43, damit die Summe im Schnitt stimmt:
               [ 1.14 0.00 0.00 3.00 0.00 2.71 ]
```

Kein Neuron kann sich auf ein anderes verlassen, keins darf ein einzelnes Trainingsbeispiel „auswendig“ speichern: Das Wissen verteilt sich robust über den ganzen Tensor.

2**L2-Regularisierung:** Große Gewichte = steile, zackige Vorhersagekurven, die sich um jeden einzelnen Datenpunkt winden. L2 addiert die quadrierten Gewichte — mal 0.01 — auf den Fehler. Drei Gewichte als Kostprobe:

```
ohne L2, auswendig gelernt: [ 12.40 −9.80 7.10 ]
  Strafe = 0.01 × (12.40² + 9.80² + 7.10²)
         = 0.01 × 300.21 = 3.00

mit L2, bescheiden: [ 0.80 −0.60 0.50 ]
  Strafe = 0.01 × ( 0.80² + 0.60² + 0.50²)
         = 0.01 × 1.25 = 0.01
```

3.00 zusätzlicher Fehler tut dem Optimierer weh, 0.01 juckt ihn nicht — dieser sanfte Dauerdruck zieht die Gewichte klein und die Vorhersagekurve glatt. Das Netz darf sich also weiter anstrengen, aber es muss seine Übertreibungen bezahlen.

So teilen wir die Daten auf

Trainingsdaten (80%)
Daraus lernt das Modell
(nur diese Noten sieht es beim Üben!)

✔ Testdaten (20%)
Die „Klassenarbeit“

(beim Training komplett unsichtbar!)

Alle Datenpunkte

30 Schülerinnen und Schüler, ihre Lernstunden pro Woche und die Prüfungsnote — mit realistischem Rauschen (auch wer viel lernt, erwischt mal einen schlechten Tag). Genau dieses Rauschen ist die Falle: Ein überangepasstes Modell lernt den schlechten Tag mit auswendig.

Student	Lernstunden/Woche	Prüfungsnote (1-6)	Daten-Set

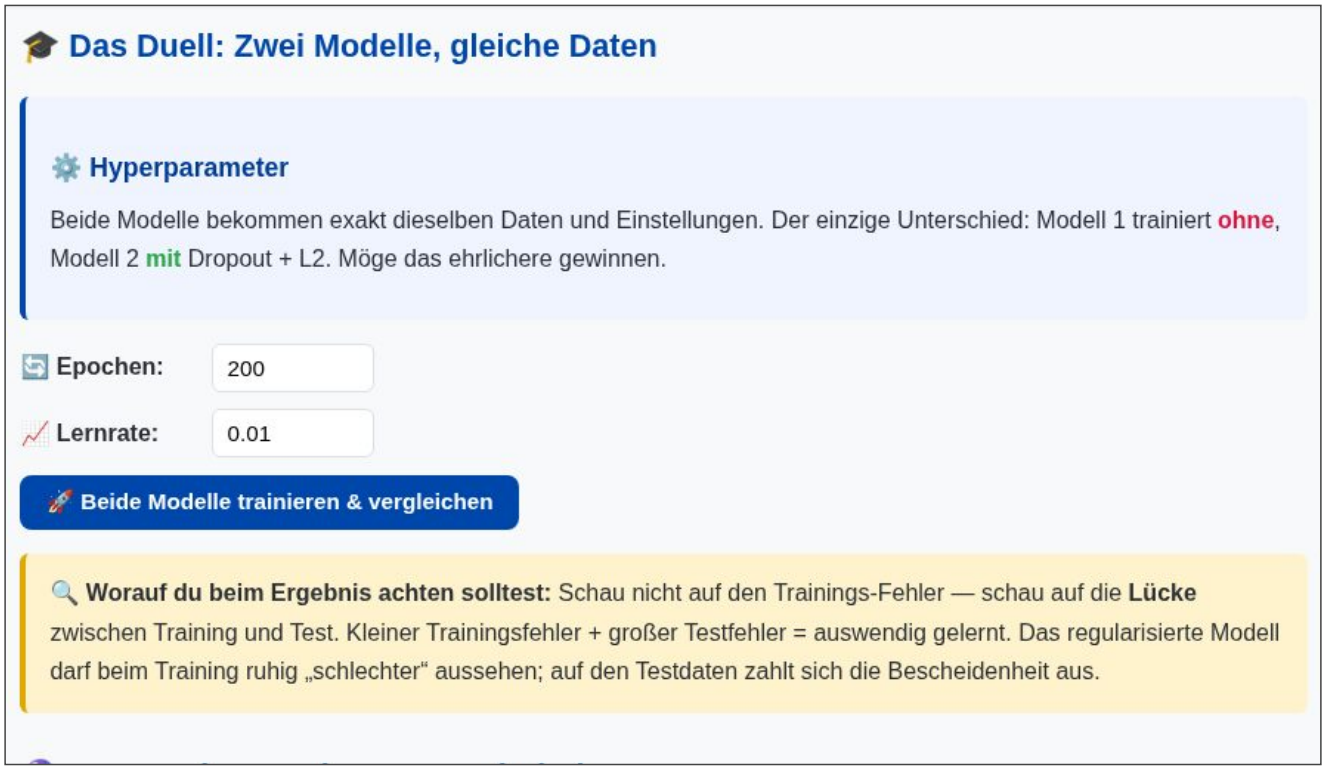
Das Duell: Zwei Modelle, gleiche Daten

Hyperparameter

Beide Modelle bekommen exakt dieselben Daten und Einstellungen. Der einzige Unterschied: Modell 1 trainiert **ohne**, Modell 2 **mit** Dropout + L2. Möge das ehrlichere gewinnen.

Am Gerät ausprobieren: herrki.de/240_overfitting_regularisierung#abschnitt-4

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

0%

Trainiere Modell OHNE Regularisierung...

0%

Trainiere Modell MIT Regularisierung...

Worauf du beim Ergebnis achten solltest: Schau nicht auf den Trainings-Fehler — schau auf die **Lücke** zwischen Training und Test. Kleiner Trainingsfehler + großer Testfehler = auswendig gelernt. Das regularisierte Modell darf beim Training ruhig „schlechter“ aussehen; auf den Testdaten zahlt sich die Bescheidenheit aus.

Der Praxistest: eine neue Schülerin

> Gib Lernstunden ein, die *nicht* in den Daten vorkommen (z. B. 15,5) — also ein Fall, den beide Modelle noch nie gesehen haben. Welches tippt plausibler?

Am Gerät ausprobieren: herrki.de/240_overfitting_regularisierung#abschnitt-5

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Zurück zur Anwendung

Denk noch einmal an den Mitschüler vom Anfang, der alle Übungsaufgaben auswendig konnte und in der Klassenarbeit trotzdem einbrach. Genau seine Krankheit hast du eben bei einem Modell gesehen — und du hast die zwei Gegenmittel selbst ausprobiert. Ab jetzt schaust du bei jeder Genauigkeits-Angabe zuerst auf die Lücke zwischen Training und Test, denn nur die verrät, ob ein Modell verstanden oder auswendig gelernt hat.

Damit hast du den Praxis-Werkzeugkasten komplett: Daten zu Tensoren machen, Modelle trainieren, ehrlich prüfen und robust halten. Jetzt wird der Werkzeugkasten breiter: In „Sehen lernen: Faltung, Filter, Pooling“ bekommt dein Netz Augen — es sucht Muster im Bild, statt Pixel einzeln anzustarren. In „Der Flaschenhals“ presst ein Netz Daten durch eine enge Stelle und baut sie daraus wieder auf. Und in „Zeitreihen: fortschreiben oder klassifizieren“ geht es um Daten, bei denen die Reihenfolge zählt. Danach wird es aktuell: Im Bereich „Aktuell“ wartet z. B. der Latent Space — die geheime Landkarte, auf der moderne KI denkt.

Zum Weiterlesen

- **Die Original-Arbeit zu Dropout:** Nitish Srivastava, Geoffrey Hinton u. a., *„Dropout: A Simple Way to Prevent Neural Networks from Overfitting“*, JMLR 2014 — jmlr.org/papers/v15/srivastava14a.html
- **Regularisierung im Überblick:** Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Kapitel 7 „Regularization for Deep Learning“ (frei online) — deeplearningbook.org/contents/regularization.html
- **Overfitting kompakt und mit Grafiken:** Google Machine Learning Crash Course, *Overfitting* — developers.google.com → Overfitting (developers.google.com/machine-learning/crash-course/overfitting)

Kurzer Check – mit Lösung

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

Frage 1: Beim Training sinkt der Trainings-Fehler weiter, aber der Test-Fehler steigt. Was passiert hier?

- Das Modell wird einfach immer besser.
- ✔ **Overfitting: Das Modell lernt die Trainingsbeispiele auswendig, statt das Muster zu verstehen.**
- Die Lernrate ist zu klein eingestellt.

Das ist DER Warnblick dieses Kapitels: Schau auf die Lücke zwischen Training und Test, nicht auf den Trainingsfehler. Wächst die Lücke, lernt dein Modell auswendig — wie der Mitschüler mit den alten Klausuren.

Frage 2: Was macht Dropout während des Trainings?

- Es löscht dauerhaft die schlechtesten Trainingsdaten.
- ✔ **Es schaltet in jedem Schritt zufällig Neuronen ab (setzt sie auf 0), damit sich das Netz nicht auf einzelne Spezialisten verlässt.**
- Es verkleinert das Netz für immer auf die Hälfte.

Zufällige Nullen im Team zwingen alle Neuronen, robuste, gemeinsam getragene Muster zu lernen. Bei der Vorhersage sind dann wieder alle an Bord — nur schlauer verteilt.

Frage 3: Modell A: Training 1 % Fehler, Test 12 %. Modell B: Training 5 %, Test 6 %. Welches nimmst du mit in die echte Welt?

- Modell A — 1 % Trainingsfehler ist unschlagbar.
- ✔ **Modell B — kleine Lücke heißt: Es generalisiert, und auf neuen Daten zählt nur der Testfehler.**
- Beide gleich — im Schnitt sind sie ähnlich gut.

In der echten Welt sieht dein Modell NUR neue Daten — also ist der Testfehler die einzige Währung, die zählt. Bescheidene 5 % im Training sind der faire Preis für ehrliche 6 % draußen.

Das nimmst du mit

- **Overfitting heißt auswendig lernen.** Der Trainings-Fehler sinkt weiter, der Test-Fehler steigt — die Lücke zwischen beiden ist der Warnblick.
- **Dropout schaltet zufällig Neuronen ab.** 30 % der Aktivierungen auf 0, der Rest ×1,43 hochskaliert — so verteilt sich das Wissen über den ganzen Tensor.
- **L2 macht große Gewichte teuer.** 0,01 × Summe der Quadrate: 3,00 Strafe tut dem Optimierer weh, 0,01 juckt ihn nicht — die Kurve wird glatt statt zackig.
- **Ohne sauber getrennte Testdaten merkst du gar nichts.** Das Modell sähe perfekt aus und wäre draußen nutzlos.
- **Es zählt nur der Testfehler.** Modell B mit 5 %/6 % schlägt Modell A mit 1 %/12 % — bescheiden und ehrlich gewinnt gegen brillant und auswendig.

So zitierst du dieses Kapitel

HerrKI: Auswendiglernen ist kein Verstehen (Overfitting & Regularisierung). herrki.de/240_overfitting_regularisierung, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.