

KI_Lernziele

Regression, Lernarten & neuronale Netze

Informatik

Übergreifend (für Regression/Klassifikation)

- erklären, warum Modelle nicht „richtig/falsch“ sind, sondern über **Fehlermaße** bzw. **Metriken** bewertet werden.

Lernarten des maschinellen Lernens

Ich kann - Beispiele jeweils der passenden Lernart zuordnen (z.B. Lernen mit Lösung vs. Muster finden vs. Lernen durch Belohnung), - beschreiben, welche Rolle **Trainingsdaten** und **Labels** bei Supervised Learning spielen, - in einfachen Beispielen erklären, was „gut gelernt“ bedeutet (z.B. wenige Fehler bei Supervised Learning).

- die Bedeutung von y (Ist-Wert) und \hat{y} (Vorhersage) erklären,
- ein lineares Modell $\hat{y} = ax + b$ **grafisch darstellen**,
- Residuen $y - \hat{y}$ bestimmen und interpretieren,
- den **Mean Squared Error (MSE)** von Hand berechnen,
- zwei lineare Modelle anhand ihres MSE **vergleichen** und begründet auswählen.

Logistische Regression (binäre Klassifikation)

Ich kann

- erklären, wofür man logistische Regression nutzt (0/1-Entscheidungen als **Wahrscheinlichkeiten**),
- aus einer linearen Kombination $z = ax + b$ mit der **Sigmoid-Funktion** $\sigma(z)$ eine Wahrscheinlichkeit $p \in (0, 1)$ berechnen,
- die **Entscheidungsgrenze** (z.B. Schwelle 0.5) erklären und anwenden,
- aus Wahrscheinlichkeiten und einer Schwelle Vorhersagen $\hat{y} \in \{0, 1\}$ ableiten,
- die Metrik **Accuracy** berechnen und in einfachen Fällen interpretieren,
- begründen, wie/warum man die Schwelle verändert (strenger vs. großzügiger klassifizieren).

Perzeptron (Step-Funktion, Lernregel)

Ich kann

- den Aufbau eines Perzeptrons beschreiben (**Inputs, Gewichte, Bias, Aktivierung**),
- einen Vorwärtsthroughlauf rechnen: gewichtete Summe z und Step-Funktion $g(z)$,
- erklären, wie der **Bias** die Trennlinie verschiebt,
- einfache logische Gatter (z.B. AND/OR) als lineare Trennprobleme verstehen,
- die **Perzeptron-Lernregel** $w_i = w_i + \eta (y - \hat{y}) x_i$ auf ein Trainingsbeispiel anwenden,
- die Rolle der **Lernrate** η erklären (zu groß/z zu klein),
- erklären, warum XOR mit einem einzelnen Perzeptron nicht lösbar ist (**nicht linear separabel**).

Mehrschichtige neuronale Netze (Hidden Layer, Idee von Backpropagation)

Ich kann

- den Aufbau eines mehrschichtigen Netzes erklären (Schichten, Gewichte, Bias; Hidden Layer als Zwischenschritt),
- begründen, warum eine versteckte Schicht komplexere Entscheidungsgrenzen ermöglicht als nur eine Gerade,
- einen einfachen Vorwärtsdurchlauf in einem kleinen Netz mit Step-Funktion nachvollziehen,
- erklären, wie XOR durch „Zwischenschritte“ (z.B. OR und NAND in der Hidden Layer) konstruiert werden kann,
- die Grundidee von **Backpropagation** erklären: Fehler bestimmen und Gewichte schrittweise korrigieren,
- ein einfaches Gewichtsupdate in der Output-Schicht mit $v_i^{neu} = v_i^{alt} + \eta (y - \hat{y}) x_i$ rechnen und interpretieren,
- erklären, warum kleine Updates (und passende Lernraten) zu stabilerem Lernen führen.