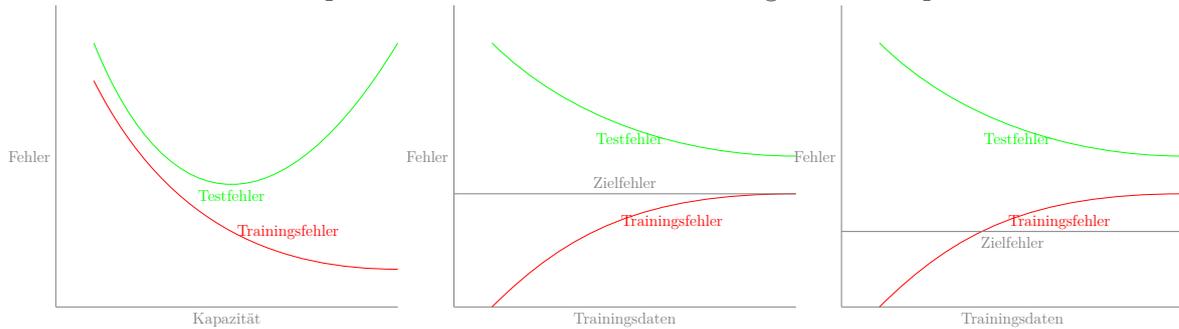


5. Übungsblatt

Aufgabe 20 Bias und Varianz

- a) Erklären Sie die Begriffe Bias und Varianz.
- b) Erläutern Sie die Verbindung von Bias und Varianz zu Overfitting und Underfitting.
- c) Erläutern Sie die Konzepte von Bias und Varianz an folgenden Graphen.



Aufgabe 21 Gewichtsregularisierung

Während der Optimierung gilt für eine quadratische Funktion mit einer Normregularisierung, dass

$$\min_{\theta \in \mathbb{R}^2} \|m - \theta\|_2^2 + \lambda \|\theta\|_\alpha$$

äquivalent zu

$$\min_{\theta \in \mathbb{R}^2} \|m - \theta\|_2^2$$

$$\text{mit } \|\theta\|_\alpha \leq \mu$$

im Sinne, dass für alle λ ein μ existiert.

- a) Skizzieren Sie für $\alpha \in \{1, 2, \infty\}$ und $\mu = 1$, wo sich die optimalen Lösungen θ des Optimierungsproblems für beliebig m befinden.
- b) Erklären sie anhand der Skizzen, welche Normen zu dünn besetzten Lösungen führen.

Hinweis : Die zweite Formel bedeutet, dass die Lösung genommen wird, die im Einheitskreis bezüglich der Norm liegt und $\|m - \theta\|_2^2$ minimiert.

Aufgabe 22 Dropout

- a) Angenommen Sie verwenden ein 10-Schichtiges Neuronales Netz mit je 10 Neuronen pro Schicht. Des Weiteren seien die Schichten untereinander voll verbunden. Wie viele Gewichte und Schwellenwerte müssen in diesem Netz insgesamt trainiert werden?
Hinweis: Bei den 10 Schichten zählt die Eingabeschicht nicht hinzu.

- b) Für den Trainingsprozess soll nun die Dropout-Methode verwendet werden. Wie viele Gewichte müssen pro Trainingsdurchlauf beachtet werden, wenn durch den Dropout (durchschnittlich) 5 Neuronen pro Schicht deaktiviert werden?
- c) Leiten Sie die Abhängigkeit der Anzahl zu lernender Gewichte zur durchschnittlichen Dropout-Rate her.
- d) Erklären Sie die Verbindung zwischen Dropout und Bagging.

Hinweis : In der Ausgabeschicht werden keine Neuronen ausgelassen. Verwenden Sie folgende Bezeichnungen für Variablen

- n_i Anzahl der Neuronen in Schicht i
- d Anteil an Neuronen, die in jeder Schicht durch Dropout deaktiviert wird
- L Anzahl der Schichten

Aufgabe 23 **Optional: Implementation eines einfachen Neuronales Netzes**

In dieser Aufgabe geht es darum ein einfaches Neuronales Netz zu implementieren und zu trainieren. Das Ziel dabei ist es, dass das Netz eine Sinusfunktion im Interval $[0, 2\pi]$ berechnet. Sie können dabei wie folgt vorgehen:

- a) Schreiben Sie ein Model, das einen Wert abhängig von Gewichten und einer Eingabe berechnen.
- b) Schreiben Sie eine Funktion, die den Fehler des Models abhängig von Trainingsdaten und Gewichten berechnet
- c) Implementieren Sie eine Funktion die den Gradienten bezüglich des Fehlers mithilfe von Vorwärtsdifferenzenquotienten abhängig von den Gewichten und einem Trainingsbeispiel berechnet.
- d) Trainieren Sie ihr Netz mithilfe von Gradientenabstieg, wobei Sie Trainingswerte zufällig von der tatsächlichen Sinusfunktion im Interval $[0, 2\pi]$ ziehen können.
- e) Visualisieren Sie das Lernergebnis indem sie die Sinusfunktion im Interval $[0, 2\pi]$ und die Ausgabe ihres Models plotten.

Hinweis: Als Programmiersprache empfehlen wir Python, aber es ist auch möglich andere Sprachen zu verwenden.