

Übungsblatt: 2

Abgabe bis spätestens 02. Juni 2025 23:59:59 GMT+1

Aufgabe 1: Ihr eigenes kleines Deep Learning Framework

Ziel dieser Übung ist das Hinzufügen zusätzlicher Layer in das Framework. Dabei soll insbesondere die Convolution Layer implementiert werden. Für ihre Implementierung sind die numpy Funktionen correlate und convolve, sowie ähnliche Funktionen nicht zulässig. Andere numpy Funktionen sind erlaubt.

(a) Erstellen Sie eine Klasse Conv2DLayer mit:

Einem Filter F (3-Dimensional), sowie der Anzahl an Filtern als Parameter. Implementieren Sie dazu die Methoden:

1. $\text{forward}(\dots) \rightarrow Y = X * F + \text{bias}_F$
2. $\text{backward}(\dots) \rightarrow \delta X = \delta Y *_{\text{rot}_{x,y}^{180}}(\text{trans}_{0,1,3,2}(F))$
3. $\text{calculateDeltaWeights}(\dots) \rightarrow \frac{\partial L}{\partial f} = X *_{ch} \delta Y, \frac{\partial L}{\partial \text{bias}_f} = \sum_i \delta y_i, f=c$

(b) (Optional) Erweitern Sie die Klasse Conv2DLayer um:

1. Einen Parameter **stride**, ein 2-Dimensionaler Shape, der angibt, um wieviel der Filter verschoben wird.
2. Einen Parameter **dilation**, der das Field of View des Filters durch eingeschobene nullen erweitert.

(c) Convolution Layer werden meist in Kombination mit Pooling Layern eingesetzt, um die Anzahl der trainierbaren Parameter im Netz zu verringern. Implementieren Sie eine Klasse Pooling2D mit:

- den Optionen **MAX** bzw. **AVERAGE** (optional), um zwischen Max-Pooling und Average Pooling zu unterscheiden.
- Dem Parameter **axis**, der angibt, in welcher Ebene der Pooling Filter verschoben wird.
- den zugehörigen $\text{forward}()$ und $\text{backward}()$ Methoden
- den Parametern **kernel_size** und **stride**

(d) Laden und Speichern

Implementieren Sie das Laden und Speichern für den Convolution Layer, wie Sie es bereits für den Fully Connected Layer getan haben.

Aufgabe 2: MNIST

(a) Verwenden Sie ein CNN zur Klassifikation der Daten von MNIST. Lassen Sie Ihre Implementierung für 10-20 Epochen laufen und notieren Sie für jede Epoche die Epochennummer, die Laufzeit der Epoche und den durchschnittlichen Loss der Epoche. Notieren Sie außerdem die Accuracy des trainierten Netzwerks auf den Testdaten. Ihre Implementierung sollte eine Accuracy von $> 95\%$ erreichen. Welche Beobachtungen stellen Sie fest? Können Sie eine bessere Qualität erreichen als mit ausschließlich Fully Connected Layern? Schreiben Sie Ihre Notizen

und Beobachtungen in die README.md-Datei und committen Sie sie zusammen mit den gespeicherten Netzwerk-Gewichten in das GitLab-Repository. Ihre Implementierung sollte so effizient sein, dass ein CNN in akzeptabler Zeit auf dem MNIST Datensatz trainiert werden kann.