

PRG-PR

Milestone 3 (Σ 300P.) - WS 2019/2020

Prof. Dr. I. Kisel, G. Kozlov, A. Belousov
Softwareentwicklung für Hochleistungsrechner
J. W. Goethe Universität, Frankfurt am Main

Deadline: 22.01.2020, 23:59

In diesem Milestone geht es um die Erweiterung des bereits bestehenden neuronalen Netzes durch ein weiteres Feature: dem Convolutional Neural Network (CNN) für die QGP-Erkennung. Dafür soll ein Feedforward-CNN implementiert werden, bei dem die Topologie frei wählbar ist. Arbeiten Sie mit Ihrem neuronalen Netz weiter und implementieren Sie Milestone 3 in das Milestone 2 Projekt - betten Sie es entsprechend in das grafische Interface ein.

1 CNN: Channels, Kernels (Σ 40P.)

Implementieren Sie...

- a) eine Klasse "Channel" für Conv3D (verschiedene Initialisierungen) (10P.)
- b) Get-, Set-, Add- und Clear-Funktionen für Channel (10P.)
- c) die Aktivierungsfunktion LReLU für Channel (5P.)
- d) Eine Klasse "Kernel" für Conv3D. Der Kernel soll mit zufälligen Gewichten initialisiert werden. (10P.)
- e) Eine Funktion mit der die Kernel-Gewichte resettet werden können. (5P.)

2 Neural Network (Σ 260P.)

Implementieren Sie...

- a) eine Funktion mit der das CNN initialisiert werden kann. Darunter fällt die Initialisierung von CNN-Layern (inkl. Channel für Conv3D und Layer für MaxPool3D) und die Initialisierung von Kernels. (25P.)
- b) eine Funktion mit der man das CNN mit Input-Daten füllen kann. (30P.)
- c) eine Clear-Funktion für das gesamte CNN. (25P.)
- d) eine Funktion die Conv3D von $28 \times 20 \times 20 \times 20$ nach $32 \times 20 \times 20 \times 20$ durchführt. Dabei soll LReLU als Aktivierungsfunktion genutzt werden. (30P.)
- e) eine Funktion die MaxPool3D für $32 \times 20 \times 20 \times 20$ durchführt. (30P.)
- f) eine Funktion die Conv3D von $32 \times 10 \times 10 \times 10$ nach $64 \times 10 \times 10 \times 10$ durchführt. Dabei soll LReLU als Aktivierungsfunktion genutzt werden. (10P.)
- g) eine Funktion die MaxPool3D für $64 \times 10 \times 10 \times 10$ durchführt. (10P.)
- h) eine Funktion die Backpropagation für Conv3D durchführt (von $32 \times 10 \times 10 \times 10$ nach $64 \times 10 \times 10 \times 10$). Darunter fällt auch die Berechnung von neuen Gewichten und des Bias in den Kernels. (50P.)
- i) eine Funktion die Backpropagation für Conv3D durchführt (von $28 \times 20 \times 20 \times 20$ nach $32 \times 20 \times 20 \times 20$). Darunter fällt auch die Berechnung von neuen Gewichten und des Bias in den Kernels. (10P.)
- j) eine Funktion die Backpropagation für MaxPool3D ($64 \times 10 \times 10 \times 10$) durchführt. (30P.)
- k) eine Funktion die Backpropagation für MaxPool3D ($32 \times 20 \times 20 \times 20$) durchführt. (10P.)