PRG-PR

Milestone 2 - WS 2019/2020

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

Deadline: 11.12.2019, 23:59

In diesem Milestone sollen die Grundlagen eines "Fully Connected Neural Network" (FC NN) erlernt werden. Dafür soll ein Feedforward-Netz erstellt werden, dessen Topologie frei wählbar ist. Das Training des neuronalen Netzes soll über Backpropagation realisiert werden. Achten Sie daher darauf, für neue Klassen neue Header- und CPP-Files anzulegen. Testen Sie Ihr Projekt entsprechend und sichern Sie es gegen ungültige Eingaben ab.

1 Neuron (90 Punkte)

Implementieren Sie...

a) ... eine Klasse "Neuron" zur Erstellung von Neuronen mitsamt Aktivierungsfunktion. Dabei soll die Aktivierungsfunktion variabel bleiben. (10P)

b) ... die Aktivierungsfunktion $\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$ und ihre Ableitung, wobei K die Größe des Vektors ist (siehe Softmax-Funktion). Diese sollte als Aktivierungsfunktion genutzt werden können. (5P)

c) ... eine Verbindung zwischen Neuronen von Layer i und Layer $i + 1$, mitsamt einer Möglichkeit die Gewichte zu hinterlegen. (30P)

d) ... eine Funktion, mit der zum einen die Output-Gradienten für Output-Layer-Neuronen, zum anderen die Hidden-Gradienten für alle Hidden-Layer berechnet werden können. (35P)

e) ... eine Funktion, mit der die Gewichte abgefragt und gesetzt werden können, die ein Neuron auf alle Neuronen des Folgelayers hat. (5P)

f) ... eine Funktion zur Berechnung von Output-Gradienten, Hidden-Gradienten, sowie eine Möglichkeit zum Updaten von Gewichten für kleinere Batches. (5P)

(Tipp: **std::vector** kann sehr hilfreich sein)

2 Neurales Netz (165 Punkte)

Implementieren Sie...

a) ... eine Klasse für ein neuronales Netz. Das Netz sollte auf einer Topologie basieren, die wiederum angibt, wie viele Neuronen je Layer existieren. (25P)

b) ... eine Funktion, mit welcher der Output des neuronalen Netzes abgefragt bzw. ausgegeben werden kann. (20P)

c) ... eine Funktion, die den Output für eine bestimmte Eingabe des neuronalen Netzes berechnet. (40P)

d) ... eine Funktion, um die Trainingsdaten auszulesen, dabei sollen sowohl einzelne Dateien als auch einzelne Batches geladen werden können. Achten Sie dabei darauf, nicht alle Dateien zeitgleich in den Arbeitsspeicher zu laden. (30P)

e) ... eine Funktion für die Backpropagation, die mit gegebenem Output verschiedene Sachen berücksichtigt: Batch-Normalisierung, Gradienten, Dropout und Gewichtsupdates. Dies soll unter Berücksichtigung verschiedener Aktivierungsfunktionen geschehen. (25P)

f) ... eine Funktion, die die Daten analysiert und verschiedene mathematische Verteilungen berechnen kann. (25P)

3 Benutzerinterface (60 Punkte)

Implementieren Sie eine grafische Benutzeroberfläche, die es dem Anwender erlaubt...

a) ... das neuronale Netz mit beliebigen Trainingsdaten zu nutzen (Dateien/Ordner sollten auswählbar sein). Während des Trainings sollte das Benutzerinterface nicht einfrieren. (15P)

b) ... den Fortschritt des Trainings zu verfolgen, während eine Trainingssession läuft. (5P)

c) ... die Anzahl der genutzten Trainingsdaten / Validierungsdaten festzulegen. (5P)

d) ... die Topologie des neuronalen Netzes zu ändern (z.B. FC NN ohne Hidden-Layer, mit einem Hidden-Layer, mit zwei, etc.) (20P)

e) ... die grafischen Anzeigen zu ändern. Dabei soll es beispielsweise möglich sein, die analysierten Daten als Histogramm grafisch auszugeben oder die Erfolgsrate im Verlauf grafisch darzustellen oder Verteilungen zu plotten. Siehe dazu auch Aufgabe 4. (15P)

4 Zeichenfläche für Graphen (85 Punkte)

a) Erstellen Sie ein Widget mit einer Zeichenfläche, auf der die Graphen dargestellt werden können. Dabei sollen die Achsenbeschriftungen variable einstellbar sein (z.B. Effizienz je Epoche). Sie dürfen dazu die *QCustomPlot* Library nutzen. (30P)

b) Erstellen Sie eine Möglichkeit auf der GUI ebenfalls einen weiteren kleinen Graphen mit der Loss-Rate je Epoche darzustellen (parallel neben den anderen Graphen). (30P)

c) Implementieren Sie die Möglichkeit ebenso Informationen der Inputdaten grafisch darzustellen, wie φ, θ und Momentum. (25P)