

Übung 3 zur Vorlesung Deep Learning

Aufgabenstellung

- Erlaubte Python-Pakete sind die Python-Standardbibliothek, NumPy, Theano und Lasagne:
 - Theano: <http://deeplearning.net/software/theano/>
 - Lasagne: <https://lasagne.readthedocs.io/en/stable/user/installation.html>
 - Empfehlenswert ist die Installation der stabilen Versionen per Python-Pip.
 - Anaconda im Raum F035 sollte Theano und Lasagne vorinstalliert besitzen.
- Zu verwendender Datensatz mit Beschreibung: <http://yann.lecun.com/exdb/mnist/>
- Ziel ist die Implementierung und Training eines Convolutional Neural Networks mit Lasagne und Theano um einen Erkenner für den MNIST-Datensatz zu erhalten.
- Implementieren Sie ein CNN mit mindestens zwei Convolutional- und zwei Pooling-Layern. Wählen Sie eine geeignete Topologie und Aktivierungsfunktionen des CNN selbst.
- Letzter Layer des CNN ist ein vollverknüpfter Feedforward-Layer mit 10 Neuronen und logistischem Sigmoid als Aktivierungsfunktion.
- Eingabe in das CNN sind Grauwertbilder mit jeweils einer handgeschriebenen Ziffer aus dem MNIST-Datensatz.
- Ausgabe des CNN sind 10 Wahrscheinlichkeiten für die verschiedenen Ziffern.
- Zielfunktion ist weiter der Mean Squared Error (MSE).
- Ausgabe des Mean Squared Errors nach jeder Epoche (ein vollständiger Durchlauf durch den Trainingsdatensatz) auf der Kommandozeile.
- Training des CNN's mit konfigurierbarer Mini-Batch-Größe, sowie auswählbar Gradientenabstieg mit Momentum, RMSProp oder AdaDelta. „Konfiguration“ per Variablen am Beginn der Python-Datei. Wählen und evaluieren Sie geeignete Parameter für das Optimierungsverfahren.
- Evaluation der verschiedenen Mini-Batch-Größen und Optimierungsverfahren (MSE über Trainingsepochen) in einer Datei, z.B. Log-Ausgabe der Kommandozeile oder in CSV-Format gespeichert. Wie verhalten sich die unterschiedlichen Optimierungsverfahren in ihrem CNN?

Fehlermaß

Gezählt wird im MNIST der prozentuale Anteil der falsch klassifizierten Beispiele. Falsch klassifiziert ist ein Beispiel dann, wenn die maximale vom CNN prädizierte Wahrscheinlichkeit nicht der korrekten Klasse/Zeichen entspricht.

Abgabekriterien

- Lösung in einer einzelnen Python-Datei.
- Zufällige Initialisierung der Parameter des neuronalen Netzes.
- Zähler der Epoche und Mean Squared Error des Trainingsdatensatzes werden auf der Kommandozeile ausgegeben.
- Ausgabe der Fehlerrate auf dem Trainings- und Testdatensatz am Ende des Trainings: Prozentualer Anteil der falsch klassifizierten (maximale Wahrscheinlichkeit entspricht nicht der wahren Ziffer) Beispiele.
- Fehlerrate auf Testdatensatz kleiner 5% falsch klassifizierte Beispiele mit mindestens einem CNN und Optimierungsverfahren.
- Dateien mit Evaluationen der verschiedenen Optimierungsverfahren aufbewahren!
- Abnahme durch E-Mail mit einer Python-Datei und Evaluation der Optimierungsverfahren an martin.schall@htwg-konstanz.de und Besprechung in der nächsten Übung mit Marc-Peter Schambach oder Martin Schall.

Abgabetermin:

04.05.2017