

EINFÜHRUNG IN DAS MASCHINELLE LERNEN – L5

BILDERVERARBEITUNG MIT PYTORCH

Bekommen in L5

Abgabe in L6

Vergewisse dich, dass du auf deiner Festplatte mehrere GB Freiraum hast.

Falls du ein Nvidia GPU hast, installiere CUDA, so dass Pytorch die GPU Acceleration benutzen kann.

Falls du ein GPU von einem anderen Vendor hast, sind andere Bibliotheken / Bindings notwendig, und der Prozess ist kompliziert und einige der Abhängigkeiten sind buggy und / oder nicht unterstützt. Falls du nicht für andere Zwecke Pytorch lokal brauchst, kannst du das überspringen und auf CPU arbeiten.

Falls du auf eine Cloud-Umgebung Zugriff hast, kannst du das Ganze überspringen. Google Collab und das von der Uni bereitgestellte Azure-Konto (login [hier](#) mit dem @stud.ubbcluj.ro-Konto) bieten **begrenzte** GPU-Laufzeiten an.

Installiere pytorch und torchvision ([link](#)). Falls du ein passendes GPU hast, überprüfe dass es anerkannt ist. Siehe [hier](#). Falls du schon Erfahrung mit tensorflow hast und eine funktionierende Installation auf deiner Maschine, kannst du es gerne anstatt Pytorch benutzen.

Erstelle ein Klassifikationsalgorithmus mit dem Flowers Dataset von [hier](#). Falls das Lesen des gesamten Datensatzes auf deine Maschine zu langsam oder gar nicht möglich ist, mache downscaling in Batches. Der Datensatz enthält keine Bounding Boxes.

Experimentiere mit 1-2 hidden layers, 0-4 convolution layers, mit / ohne Pooling, mit mehreren Anzahlen von Neuronen pro Layer, mit mehreren Aktivierungsfunktionen, Lernraten, und Werten für L1- bzw. L2-Regularisierung, mit oder ohne Batch Normalization (im Stil von Grid Search). Benutze auch Augmentation falls notwendig. Finde heraus, wie man die Gewichte und Biases, und die Größen der Vektoren / Matrizen ausgeben kann, und zeige sie für das beste Modell.

Monitorisiere die Trainingszeit. Finde ein passendes Cutoff, nach wie vielen Epochen kann der Prozess gestoppt werden?

Es ist zu erwarten, dass das Training länger dauert. Auf einem Laptop mit Intel Core i7 quad, 16 GB RAM, ohne GPU support, dauert ein Trainingsexperiment mit 60 Epochen einige Minuten. Falls du ein GPU hast, vergleiche die Laufzeit CPU vs GPU. Nachdem du die Experimente zum Laufen bringst, starte das Prozess und das Logging, und mache gerne etwas Anderes bis es läuft. Das vollständige Prozess (mit Logging der Ergebnisse und Zeiten von allen Experimenten) kann bei Bedarf auch über Nacht laufengelassen werden.

Erstelle Plots mit train / validation loss und train / validation accuracy je Epoche, sowie die Konfusionsmatrix, für alle Experimente. Interpretiere diese Plots.

Bonus × 1.5 für das Team mit dem besten macro-f1

Bonus × 0.5 für das Team mit dem zweitbesten macro-f1

