Technische Hochschule
Brandenburg
University of
Applied Sciences
Fachbereich
Informatik und Medien

Deep Learning zur Objektdetektion in Bildern mit Region-based Convolutional Neural Networks und GPU-Computing

Jonas Preckwinkel

Masterarbeit • Studiengang Informatik • Fachbereich Informatik und Medien • 14.03.2018

Aufgabenstellung

Ziel der Arbeit ist die Erarbeitung von Implementierungen verschiedener Region-based Objektdetektionssysteme, um ein generelles Verständnis dieser zu erlangen. Durch das Verändern von Hyperparametern beim Training dieser Implementierungen und durch Evaluierung der resultierenden Modelle sollen Erkenntnisse für die Wahl von bedeutenden Hyperparameterkonfigurationen in ähnlichen Anwendungssituationen gewonnen werden.

Deep Learning

Deep Learning ist eine Variante des maschinellen Lernens mit der auch generalisierungsfähige Modelle mit Viel-Dimensionalen Daten (z.B. bei Bildern zählt jeder Pixel als Dimension) erlernt werden können. Die Modelle bestehen dabei aus vielen miteinander verbundenen Schichten von künstlichen Neuronen, deren Parameter durch Konzepte des Deep Learnings angepasst werden, um hierarchisch Merkmale der zugrundeliegenden Datenmenge zu erlernen (die vorderen Schichten lernen generelle Merkmale während hintere Schichten spezielle Merkmale erlernen).

Detektion von Objektklassen in Bildern

Im Forschungsfeld „Computer Vision" werden Algorithmen entwickelt, um die Fähigkeiten der visuellen Wahrnehmung des Menschen im Computer zu verwirklichen. Dazu gehören die Detektion und Klassifikation von Objekten.
Detektionsalgorithmen müssen in der Lage sein, ein oder mehrere Objekte zu erkennen, deren Position im Bild und ihre Klasse zu bestimmen und bauen daher auf der Klassifikation auf.

R-CNN, Fast-RCNN, Faster-RCNN [1]

In der Arbeit wurden diese aufeinander aufbauenden Objektdetektionssysteme untersucht. Sie reduzieren das Problem der Detektion auf die Klassifikation, indem am Anfang Vorschläge für mögliche Objekte im Bild gesucht werden (sog. Regions of Interest RoI), welche daraufhin klassifiziert werden. In R-CNN werden 2000 RoIs pro Bild einzeln durch die convolutional Layer und den Klassifikator propagiert.
Mit Fast-RCNN wird das ganze Bild bloß einmal durch die conv. Layer propagiert. Anhand der extrahierten Merkmale des letzten conv. Layers werden die RoIs von einem speziellen Pooling Layer in eine feste Größe umgewandelt, da die darauf folgenden fully-connected Layer dies erwarten.
In Faster-RCNN werden die RoIs mit einem sog. Region Proposal Netzwerk (RPN) generiert und daraufhin wie in Fast-RCNN klassifiziert.
Das **RPN** (Abb. 1) klassifiziert jede Stelle des Bildes als Objekt oder Hintergrund und verfeinert die Position einer Bounding Box für ein mögliches Objekt an dieser Stelle im Bild. Als Ergebnis liefert das RPN pro Stelle im Bild k Bboxen und zu jeder Bbox einen „Objectness Score". In einem Bild mit 960×640 Pixeln werden etwa 2400×k Bboxen berechnet, die daraufhin gefiltert werden, sodass nur die besten RoIs klassifiziert werden.

System und Implementierung

Das Training von künstlichen neuronalen Netzen profitiert enorm von der Verwendung von Grafikkarten. Für das Training der Modelle standen daher ein System mit einer Geforce GTX 1060 (6GB) und eines mit einer Geforce GTX 1080 zur Verfügung. Tensorflow und die high-level API Keras ermöglichen einen einfachen Einstieg zur Erstellung von künstlichen neuronalen Netzen. Daher wurde eine Keras Implementierung von Faster-RCNN gewählt, mit der die Modelle trainiert wurden.

Training und Evaluation

Trainiert wurden die Modelle auf den Daten der Pascal VOC Challenge 2012 [2] und evaluiert auf den annotierten Testdaten der VOC Challenge 2007. Die Modelle wurden auf 80% der Trainingsdaten trainiert und auf 20% validiert. Indem der Validationsfehler beobachtet wird, kann das Training abgebrochen werden, falls das Modell anfängt, die Trainingsdaten auswendig zu lernen. Nach erfolgtem Training und Abbruch des Trainings durch Earlystopping Methoden wurden die Modelle evaluiert, indem die „mean Average Precision" (mAP) des Modells auf den Testdaten berechnet wurde.

Ergebnisse und Fazit

Die Architektur der künstlichen neuronalen Netze entsprach der VGG16 Architektur [3]. Die besten Ergebnisse lieferten Modelle, die im Training die originalen Gewichte des VGG16 Netzes verwendeten und diese finetuned haben. Außerdem konnten mit der Optimierungsfunktion „Adam" gegenüber SGD bessere Modelle erlernt werden, was auf die adaptive Lernrate von Adam zurückzuführen ist.
Das Problem der Objektdetektion konnte mit einer Implementierung von Faster-RCNN nachvollzogen werden. Es wurden Modelle mit verschiedenen Hyperparameterkonfigurationen trainiert, aus deren Trainingsverlauf und Evaluationsergebnissen Schlüsse auf die Wahl der Hyperparameter gezogen werden konnten.

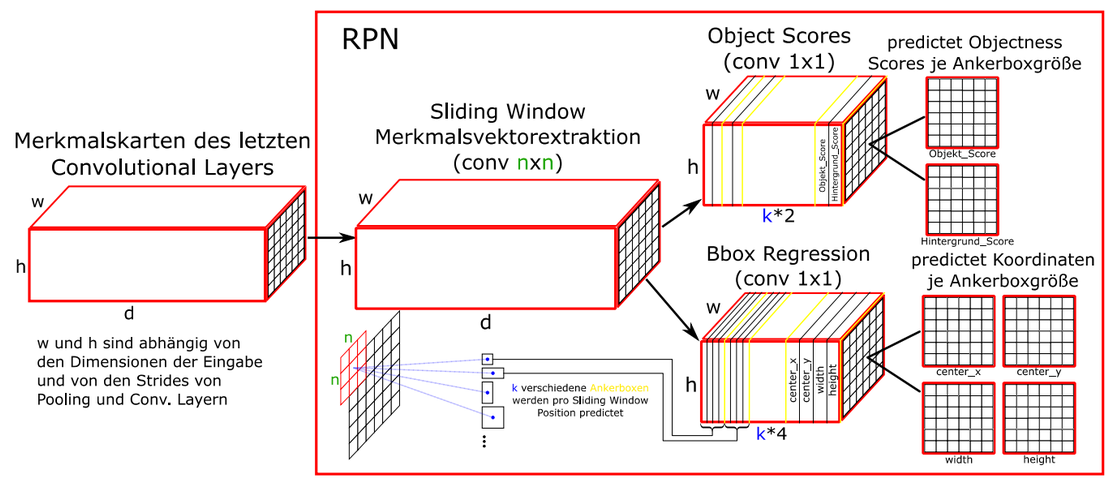

Abbildung 1: Region Proposal Netzwerk

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems (NIPS), 2015.
[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for largescale image recognition," Available: http://arxiv.org/abs/1409.1556

Betreuung Prof. Dr. Sven Buchholz • Technische Hochschule Brandenburg • Dipl.-Inf. Ingo Boersch • Technische Hochschule Brandenburg