Optimierung „Orange Cam“

Studiengang: BSc in Elektrotechnik und Informationstechnologie | Vertiefung: Communication Technologies
Betreuer: Prof. Dr. Elham Firouzi
Industriepartner: Schleuniger AG, Thun

Neuronale Netze zur semantischen Segmentierung haben heutzutage viele Anwendungen: Ein selbstfahrendes Auto erkennt Hindernisse und die Fahrbahn, eine industriell eingesetzte Maschine erkennt, welches Objekt ein Roboter vom Band nehmen sollte und viele weitere Anwendungen. Je nach Komplexität kann die Ausführung des Netzes mehrere Sekunden dauern, was je nach Anwendung zu viel ist. In dieser Arbeit geht es um die Performance Optimierung eines solchen Netzes.

Einsatzgebiet

Schleuniger stellt modular aufgebaute Maschinen für Industriekunden her. Die Maschine (siehe Abbildung 1) crimpt ein Kabel mit verschieden möglichen Anschlüssen, Schnitten etc.
Im Projekt geht es um ein Modul, welches im Bereich der Qualitätssicherung eingesetzt werden wird. Das Kabelende wird fotografiert und dieses Foto wird anschließend mit Hilfe von einem Neuronalen Netz analysiert, sodass auf entsprechende Fehler reagiert werden kann.

Ziel der Arbeit

Das Kamerasystem „Orange Cam“ besteht aus vier Kameras, von denen drei das Kabel in 120° versetzten Positionen abfotografieren und einer Kamera, die frontal das Kabelende fotografiert. Die Kameras werden von einem Nvidia Jetson AGX Xavier gesteuert und die Fotos mit Hilfe des auf Tensorflow basierenden Neuronalen Netzes nacheinander semantisch segmentiert. Da dieses Kamerasystem keine lange Zeit zwischen zwei Fotosätzen hat, muss die semantische Segmentierung performanceoptimiert werden, damit die Segmentierung fertig ist, bevor neue Fotos gemacht werden.

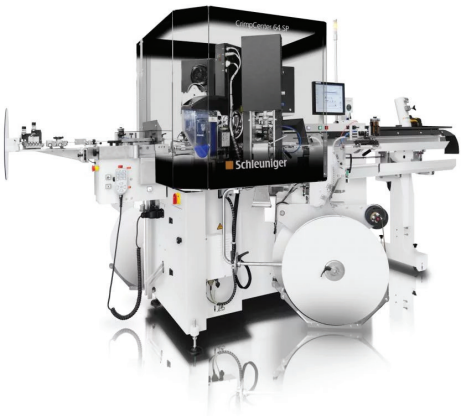

Abbildung 1 CrimpCenter 64 SP

Optimierungsmethoden

Pruning

Die Performance der Bildsegmentierung konnte mit Hilfe von Pruning optimiert werden, also der Löschung einiger Knoten (beispielsweise die oberen beiden Knoten in Layer h_n in Abbildung 2), die nur wenig Einfluss auf das Gesamtsystem hatten. Der Verlust an Genauigkeit ist hierbei zwar spürbar, aber das Ergebnis liegt im Rahmen des Nutzbaren.

Änderung der Genauigkeit

Der Versuch, zusätzlich Performance durch Verringerung der allgemeinen Netzgenauigkeit konnte nicht, wie ursprünglich geplant von Float32 komplett auf Int8 durchgeführt werden. Obwohl die Performance von Int8-basierten Netzen ein Vielfaches der von genaueren Netzen ist, reicht die Genauigkeit in der Praxis nicht aus. Float16 wurde als guter Kompromiss zwischen Performancegewinn und Genauigkeit als Basis gewählt.

Fazit

Aus Sicht des aktuellen Standes konnte eine Optimierung mittels Pruning um etwa 20% Rechenzeit erzielt werden. Verschiedene anderen Methoden konnten als Mittel zur Optimierung ausgeschlossen werden. Betrachtet man den allgemeinen Performancegewinn des optimierten Systems im Vergleich zum ursprünglichen Netz, sind die Erwartungen erfüllt.

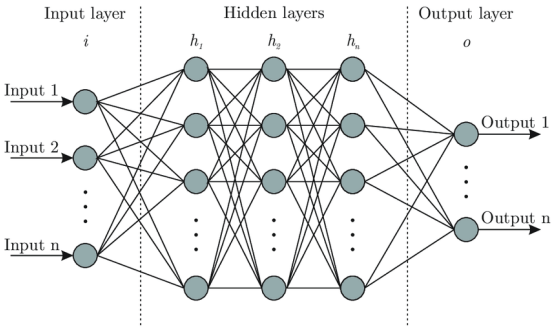

Abbildung 2 Allgemeine Architektur eines neuronalen Netzes

Frederik Phillip Wolf