

# Parallel Programmeren: Classificatie met Neurale Netwerken

Pieter Dilien, Waldo Wautelet

May 20, 2022

## 1 Introductie

In dit verslag zal het paralleliseren van een programma dat een afbeelding kan classificeren besproken worden. Eerst zal het originele niet-geparalleliseerde programma bekeken worden. Vervolgens zal er ingegaan worden op de verbeteringen die er aan het originele programma aangebracht zijn om het parallel te laten functioneren. Ten slotte zullen de resultaten van de beide programma's vergeleken worden.

## 2 Origineel

Eerst en vooral werd bestudeerd hoe het programma juist werkte. Vervolgens bekeken we welke layers het meeste computing time innamen. Dit om te bepalen welke layers nog het meest ruimte voor verbetering hadden. De layertijden van het originele programma zijn te zien op figuur 2.

Het is duidelijk dat layers 1, 3, 6, 10, 14, 18, 19, 21 en 22 reeds erg snel zijn. Deze hebben dus minder marge voor verbetering. Hierdoor keken we naar de overgebleven layers. Het werd snel duidelijk dat deze allemaal gebruikmaken van de `convolution_layer` functie. Het is dan ook deze functie dat gekozen werd om te paralleliseren.

## 3 Verbeterd

## 4 Resultaten

Het verbeterde programma is duidelijk sneller dan het originele, dit is zichtbaar in figuur 1. De totale computing time werd verlaagd van gemiddeld 17 seconden naar gemiddeld 2.5 seconden.

Daarnaast is duidelijk te zien in figuur 2 dat het opzet om layers 2, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16 en 17 - die allemaal gebruikmaken van de `convolution_layer` functie - te versnellen geslaagd is.

Het is ook goed te zien dat al de andere layers die nog steeds volledig op de CPU uitgevoerd worden en dus niet geparalleliseerd zijn nog steeds dezelfde tijd hebben.

Ten slotte is het belangrijk om aan te geven dat ondanks het versnellen van het programma, we wel nog een fout resultaat krijgen. We zouden "tabby\_cat" moeten krijgen, maar krijgen in plaats daarvan steeds het resultaat "buckle". We hebben

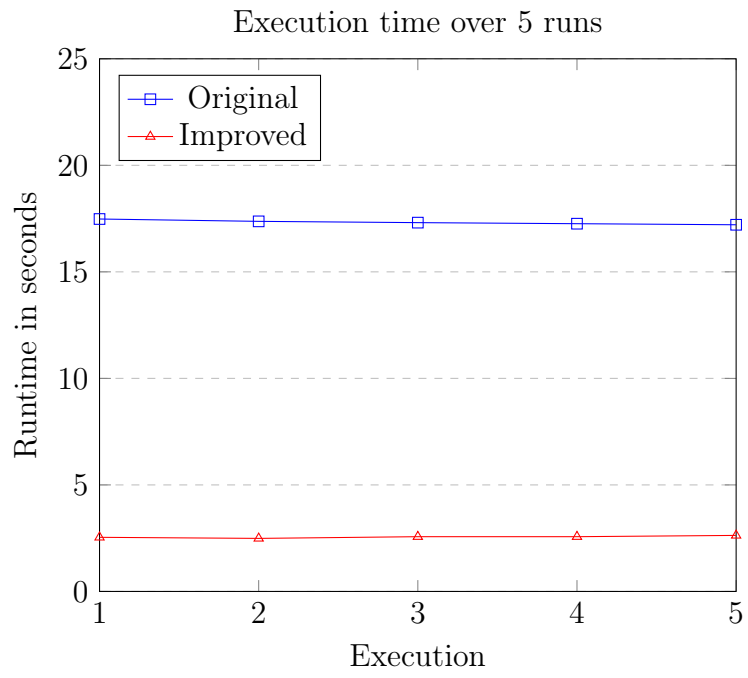


Figure 1: Execution time of original and improved program

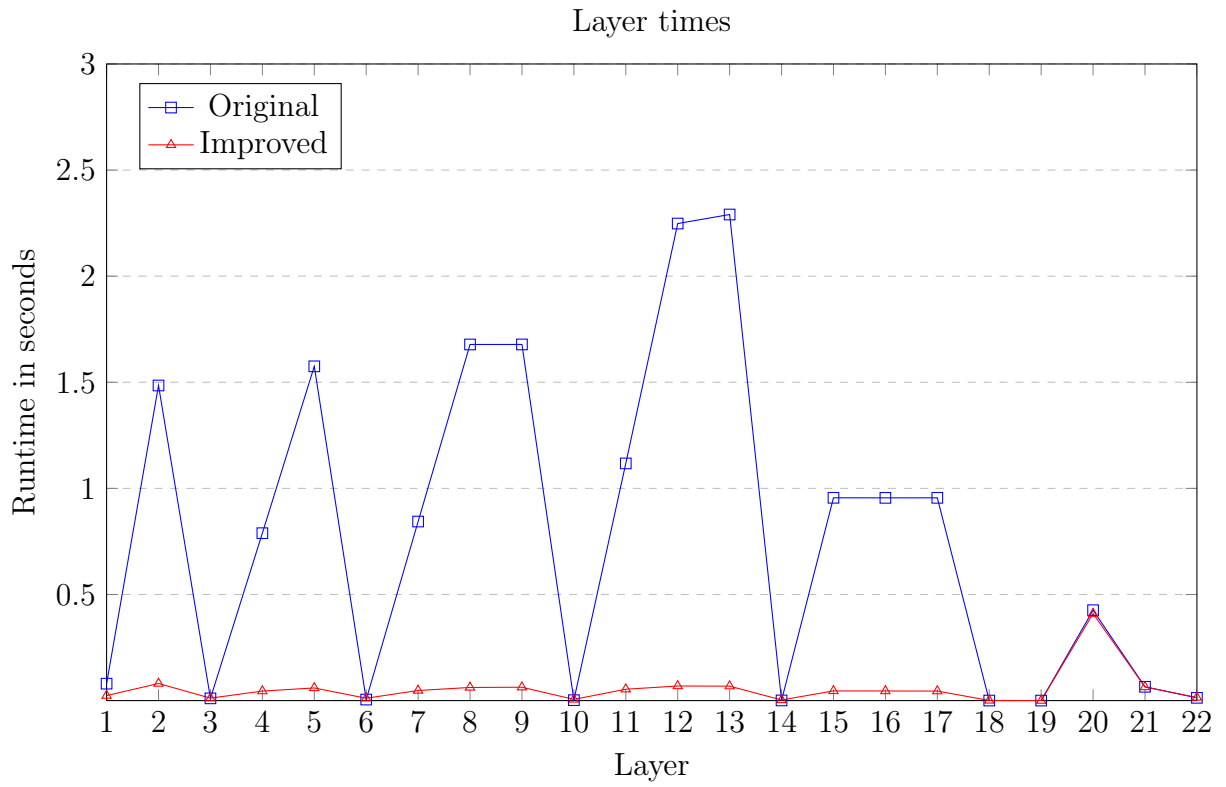


Figure 2: Layer times of original and improved program