Deep Learning Exercise 2 Report

Kai Haase, 4118916

November 20, 2017

1 Task 1: Implementing a deep CNN in Tensorflow

See jupyter file.

2 Task 2: Changing the learning rate

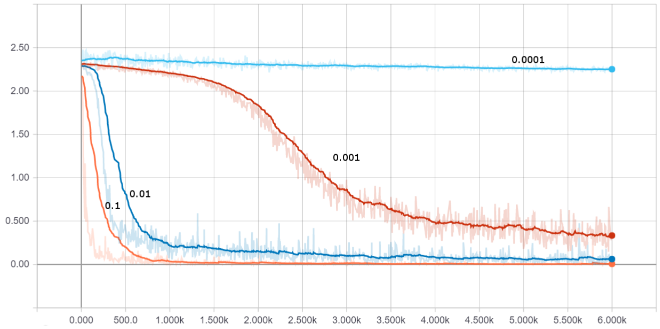

Figure 1: Loss development with different learning rates

In the figure we can see that high learning rates tend to find an semi optimal solution in relatively quick time, but also will vary a lot when found one. Low learning rates are steadily improving into the right direction with less variation, but a lot slower. I could imagine high learning rates being useful for testing concepts, and low learning rates for building a best possible solution in longer terms.

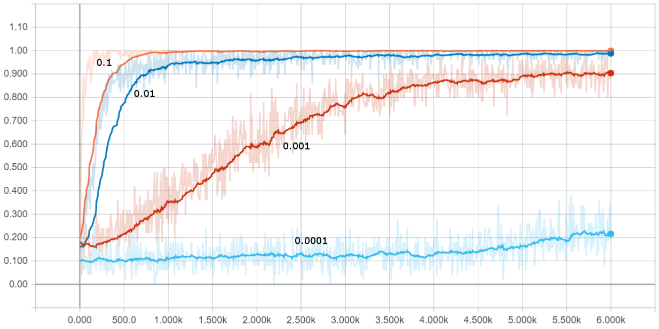

Figure 2: Accuracy development with different learning rates

3 Task 3: Runtime

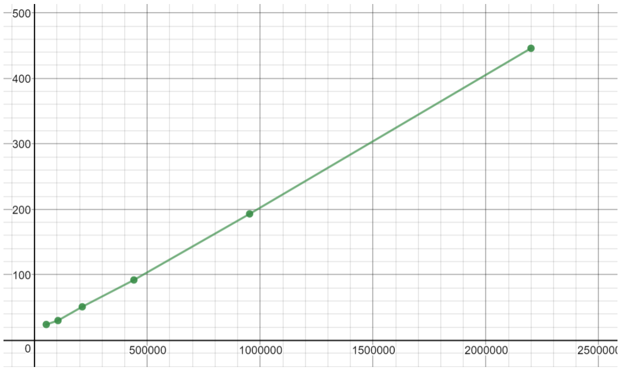

Figure 3: Runtime depending on amount of trainable parameters when using gpu

Here we can see that gpu is not not only a lot faster than gpu training in general, but especially after a threshold, when the amount of trainable parameters can only be handled by effective multithreading. From here on, the cpu has no chance performing against the gpu.

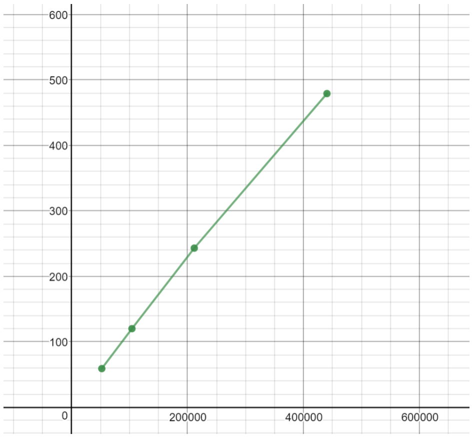

Figure 4: Runtime depending on amount of trainable parameters when using cpu