EML10EX03 - Leandro Borzyk, Niels Kissner, Florian Schrittwieser

3.1 PyTorch: Automatic differentiation

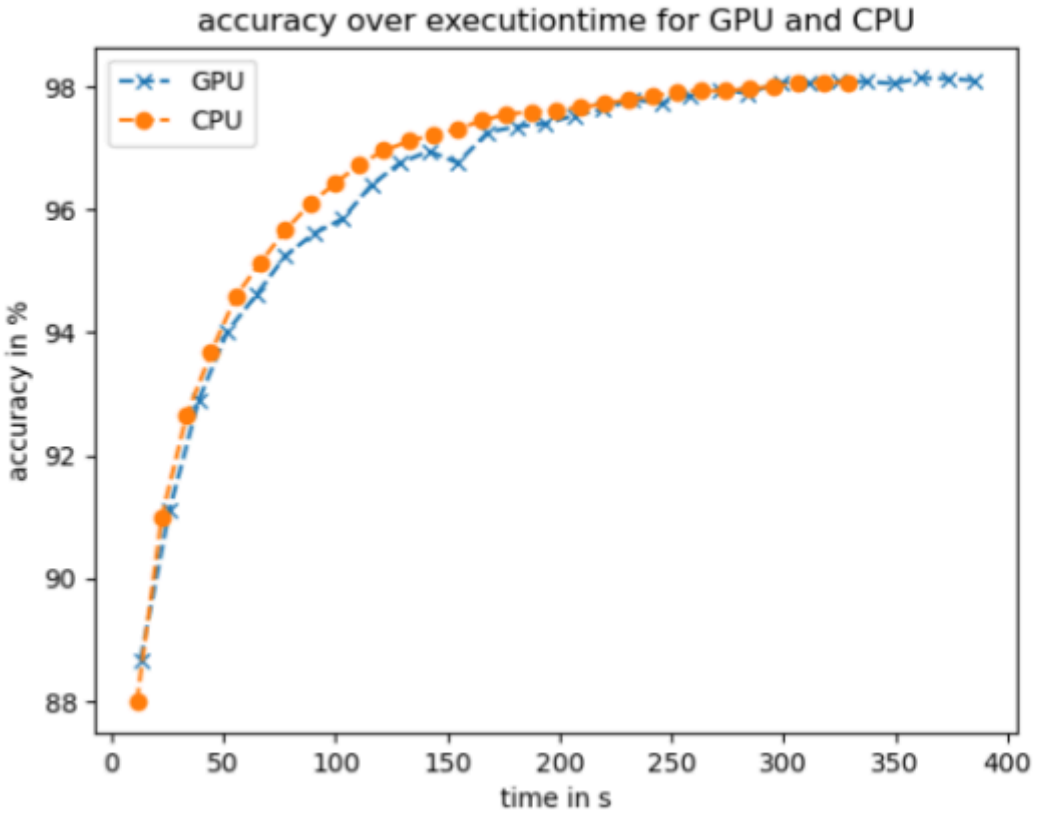

Fig.1

Figure 1 shows a comparison of runtime between a GPU and a CPU. It can be noticed that the CPU application is still faster than the GPU, both applications used the same dataset, epochs and model. Since the MNIST dataset and the MLP represents a rather small problem a bigger advantage can be assumed for a bigger problem.
For comparison between CPU and GPU a hardware comparison is reasonable.
CPU: csg-brook01 Arch=x86_64 CoresPerSocket=16, but of these 16 cores only one is utilized, by setting pytorch Flags more cores can be used.
GPU: RTX2080TI 544 Tensor Cores especially used for deep learning

3.2 MLP and CNN on CIFAR-10

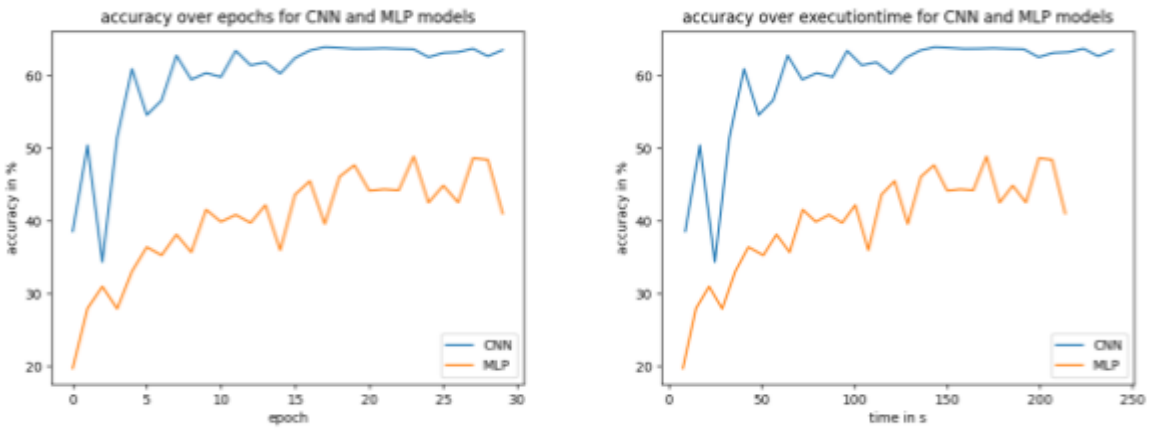

Fig.2

Figure 2 depicts accuracy comparisons between the MLP and CNN network architectures. It shows that the CNN drastically outperforms the MLP on the CIFAR-10 dataset. Comparing the accuracy over epochs, the CNN has consistently a more than 10 % higher accuracy (<50% for the MLP vs >60% for the CNN).
While the CNN takes more time to train (per epoch), the accuracy over time chart (also Fig. 2) shows that the better performance more than compensates for that, with the CNN having a better accuracy at every training time.

3.3 Optimizers

Figures 3 and 4 show a comparison of multiple different optimizers and different learning rates. It can be noted that all optimizers have fundamental different characteristics for this CNN application. For the optimizers Adadelta and SGD a higher learning rate is more beneficial, whereas for Adagrad, Adam, Rprop and RAdam a rather low learning rate is better.

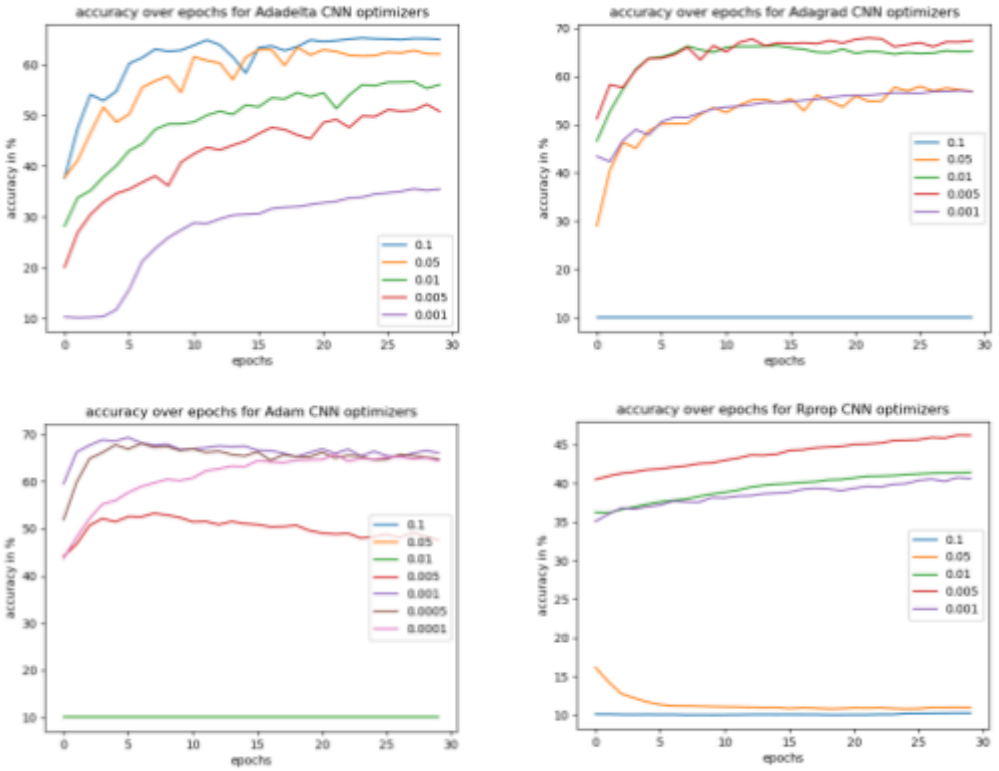

Fig. 3

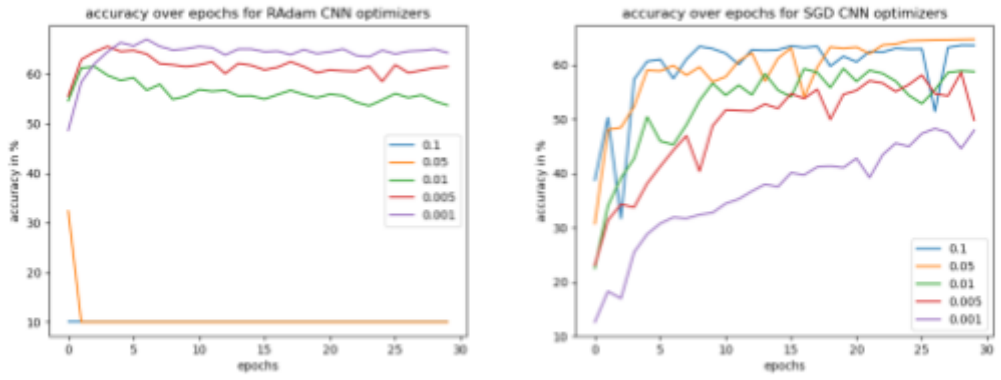

Fig 4

Figure 5 illustrates the accuracy over epochs, whereas for each optimizer the optimal learning rate was chosen. After 30 epochs Adagrad with a learning rate of 0.05 performs amongst its competitors as best.

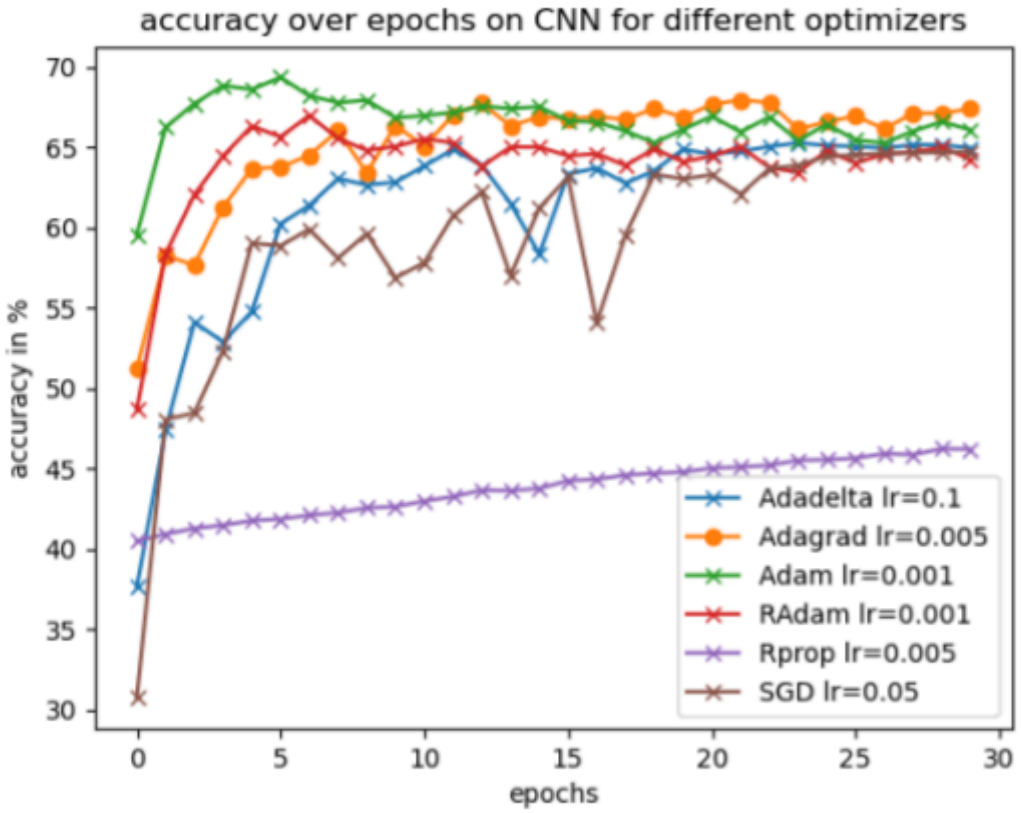

Fig 5

3.4 Willingness to Present:

Willing to present 3.1, 3.2 and 3.3.