Project proposal:

Replication of examination of Hyperparameters

**Group name:** Team Jönsson.

**Team Members**: Ola Roos, Khushdeep Singh, Andreas Jönsson

**A brief description of the problem:**

Replicate the experiments in (1) ‘*A Disciplined Approach To Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay’* on increasingly advanced networks.

This papers experiments were only evaluated on Image Classification Networks, if there is the time we want to run the same experiments on U-net and/or other image-classification networks to examine if there is a difference in behaviour.

**Training Data:**

We will use the CIFAR-10 library.

**Software Package:**

PyTorch, Jupyter Notebook, TensorBoard for PyTorch, OverLeaf, GitHub, 2080 ti.

**How much of the software implementation your group will write self:**

We will try to write as much of model structure ourselves. We will try to build smaller versions of the classical networks used in the paper [1] – models that are possible to train in a reasonable time.

**The initial set of experiments we will run:**

Download the main models used in the paper: Resnet, Densenet, etc with pytorch and experiment-train the network and get estimate on how long training will take.

Use this information to estimate how deep we should build the smaller replicas of these models in order to have time to do all experiments we want.

Write code to save plots and models in a comprehensive way to be easily analysed and put into the report.

Work top down examining the hyperparameters in the paper. Starting with cyclic-learning rate, then investigating batch size, following up with cyclical momentum, and lastly adding weight decay.

Start with shallow 3-layer network; set up all the hyperparameter experiments on it. Change network and run the same experiments. Replicate top down experiments in paper [1]. Then Resnet, Densenet, if we have time U-net.

**How will we measure the success of our project:**

Compare the results for our hyperparameters (for our compressed models) to the results in the paper[1].

**What we want to learn:**

**Ola:** I want solid knowledge of go-to methods for tuning basic hyperparameters and their relation to each other. PyTorch experience to build more advanced models for my master thesis. Couple TensorBoard to PyTorch for analyzing the experimental results.

**Khushdeep:** I am interested to gain insights and develop multi-layered neural networks. Also, knowing about the methods for tuning hyperparameters and how they affect each other. This would help me to acquire knowledge about training deep neural networks to incubate with mechanical robots in the future.

**Andreas:** Besides getting hands-on experience in working with industry standard software for deep learning applications (PyTorch) I am also looking forward to learning about different ways of selecting hyperparameters. Before taking courses that touched on neural networks, I read that selecting the best parameters was so difficult that it could as well be black magic. I don’t think that sounds very scientific, so investigating in detail methods and/or heuristics for selecting parameters makes sense to me, both for my academic knowledge, as well as my future career, whatever it may hold.

Project proposal:

Replication of examination of Hyperparameters

**Team Members**: Ola Roos, Khushdeep Singh, Andreas Jönsson

We will replicate the experiments in (1) ‘*A Disciplined Approach To Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay’* on increasingly advanced networks. Using PyTorch, Jupyter Notebook and TensorBoard for PyTorch we will try to write as much of model structure ourselves. We will try to build smaller versions of the classical networks used in the paper [1] – models that are possible to train in a reasonable time. We will download the main models used in the paper: Resnet, Densenet, etc with pytorch and experiment-train the network and get estimate on how long training will take. Using this information to estimate how deep we should build the smaller replicas of these models in order to have time to do all experiments we want. Write code to save plots and models in a comprehensive way to be easily analysed and put into the report.

We will work top down examining the hyperparameters in the paper. Starting with cyclic-learning rate, then investigating batch size, following up with cyclical momentum, and lastly adding weight decay. Starting with shallow 3-layer network, we will set up all the hyperparameter experiments on it. Change network and run the same experiments. After successfully recorded the effect of the hyperparameters on that simple network, we will do the same for a resnet that has 18 layers. We will then do the experiments on a densenet. If we have time, we will also run the experiments on a U-net that we might have to implement ourselves.

In order to evaluate the success of the report, we will compare our results to the original report. We will collaborate using OverLeaf and GitHub. If feasible, we will try to train using a 2080 ti GPU.

**What we want to learn:**

**Ola:** I want solid knowledge of go-to methods for tuning basic hyperparameters and their relation to each other. PyTorch experience to build more advanced models for my master thesis. Couple TensorBoard to PyTorch for analyzing the experimental results.

**Khushdeep:** I am interested to gain insights and develop multi-layered neural networks. Also, knowing about the methods for tuning hyperparameters and how they affect each other. This would help me to acquire knowledge about training deep neural networks to incubate with mechanical robots in the future.

**Andreas:** Besides getting hands-on experience in working with industry standard software for deep learning applications (PyTorch) I am also looking forward to learning about different ways of selecting hyperparameters. Before taking courses that touched on neural networks, I read that selecting the best parameters was the most difficult part of machine learning, so investigating in detail methods and/or heuristics for selecting parameters makes sense to me. Both for my academic knowledge, as well as my future career, whatever it may hold.