Assignment 6

Welcome to the sixth assignment of the lecture *Visual Computing*. **Please read all instructions carefully!** This assignment covers deep neural networks and transfer learning. Submission is due on Wednesday, Jule 2nd, 2021 at 8pm. Please note that late assignments will receive zero (0) marks, so you are strongly encouraged to start the assignment early. Please submit your solutions via `read.mi.hs-rm.de`.

Exercise 1 (6 points)**.** Create a deep neural network with five hidden layers of 100 neurons, Xavier initialization and RELU activation functions using the software library *PyTorch*.

1. (3 points) Use Adam optimization and L_2-regularization to train the network on digits 0 to 4 of the MNIST dataset (`http://yann.lecun.com/exdb/mnist/`). *Only use digits* 0 *to* 4 *as you should use transfer learning for digits* 5 *to* 9 *in exercise 2*. You will need a *softmax* output layer with five neurons. Save checkpoints at regular intervals as well as the final model so you can reuse it later.

2. (1 point) Tune the hyperparameters using *cross-validation* and see what precision you can achieve.

3. (1 point) Add *Batch Normalization* and compare the learning curves. Is it converging faster than before? Does it produce a better model?

4. (1 point) Is the model overfitting the training set? Try adding *dropout* to every layer. Does it help?

Exercise 2 (6 points)**.** Create a deep neural network that reuses all the pretrained hidden layers of the model obtained from exercise 1, freezes them and replaces the *softmax* output layer with a new one.

1. (3 points) Train the new network on digits 5 to 9, using only 100 images per digit, and time how long it takes. Despite this small number of examples, can you achieve high precision?

2. (1 point) Try caching the frozen layers, and train the model again. How much faster is it now?

3. (1 point) Train the model again reusing just four hidden layers instead of five. Can you achieve a higher precision?

4. (1 point) Unfreeze the top two hidden layers and continue training. Can you get the model to perform even better?