

Deep Learning

- Sheet 2-

On this sheet, we will analyze the effect of (i) the optimizer, (ii), the learning rate, and (iii) the batch size on learning progress and runtimes for the CIFAR-10 dataset, which is a part of both libraries Keras and PyTorch and can be downloaded through any of these libraries; using the keras library for this is recommended. The data must be flattened into 2D format.

The only deliverable of this sheet in Teams is a zip file with the GIF animations from exercise 2.

Exercise 1 (6 Points) Implement the `train_util.py` file and upload it to gradescope. More specifically:

1. Implement the function `make_model(framework, seed)` that will create a new neural network with input and output layers suitable for CIFAR-10, and 4 hidden layers with 1024, 512, 256, 128 neurons; use ReLU as activation function. Make sure that the weights are always the same for a framework if the seed is the same. It must work for a value of `framework` in `sklearn`, `keras`, and `torch`.

2. Implement the function

```
train_network(network, X, y, optimizer, learning_rate, batch_size, timeout)
```

that will train a freshly generated network with the provided optimizer (given as string `sgd` or `adam`) and the given learning rate and batch size on the data (X, y) for `timeout` seconds. The data is provided as numpy array (X is 2D and y 1D). Make sure that you will not exceed the allowed runtime by more than 10s; otherwise you will not receive points at gradescope.

The function must return a *list* of tuples (t, l) , one for each *update* on the network parameters. Here, t is the elapsed time since training start (in seconds, rounded to 4 places after the comma), and l is the cross entropy observed for the batch.

Hint: Use `isinstance` to check the type of `network` so that you know which type of training routine you need to apply.

Hint for Keras: Check the `train_on_batch` function and the `SparseCategoricalCrossentropy`.

Exercise 2 (3 points) For each of the frameworks, test performance on (at least) batch sizes 16, 64, 256, 2048 for both SGD and Adam under at least two different learning rates, one of them being 0.001). Apply a runtime of 10 minutes.

Then create a GIF where you show the performance of each framework over time, in each configuration.

Hint: The overall runtime for the minimal setup here is $3 \cdot 4 \cdot 2 \cdot 2 \cdot 10 = 480$ minutes, i.e. 8 hours. So it is highly recommended to create CSV files for each setup and to create figures only post hoc based on these results.

Exercise 3 (1 Point) Participate in the competition at gradescope for this sheet. Use the best configuration (framework + optimizer + learning rate + batch size) you were able to find in the previous exercise.