Exercise 4

Ilya Loshchilov
ILYA.LOSHCHILOV@gmail.com

Due: 23.12.2016 14:00

In this exercise we will study a useful algorithm for hyperparameter optimization of deep neural networks called Hyperband. More specifically, we will compare it to random search when training on i) an artificial benchmark problem, ii) MNIST classification task and iii) CIFAR-10 classification task (this is optional).

Hyperband is a simple yet efficient optimization algorithm. Its easy-to-follow description and Python pseudo-code is available at

https://people.eecs.berkeley.edu/~kjamieson/hyperband.html

A more formal description and theoretical results are available at

https://arxiv.org/abs/1603.06560

The source code that implements Hyperband specifically for this exercise is available at

http://aad.informatik.uni-freiburg.de/teaching/ws2016/dl_course/index.html

We also provide a MatLab script to plot results for Scenarios #1, 2 and 3 given below. Very much like in the previous lab exercise, you may use your own script instead of our MatLab plotting script. Have a look at the latter when in doubt.

Please send us your report, source code and raw data so that we can reproduce your figures.

Scenario #1. An artificial benchmark.

We define a simple one-dimensional loss function of a single hyperparameter $x \in [0,1]$:

$$f(x) = \left(\sqrt{|x_{opt} - x|} + \frac{0.5}{n_e}\right)(1 + |\mathbb{N}(0, \sigma^2)|), \quad (1)$$

$$x_{opt} = x^* - \frac{x_{shift}}{\sqrt{n_e}},$$

where $x^* \in [0,1]$ is a location of the true optimum, x_{shift} is the maximum shift of the optimum which is controlled by n_e, the number of epochs or training points or computational budget used to train your model.

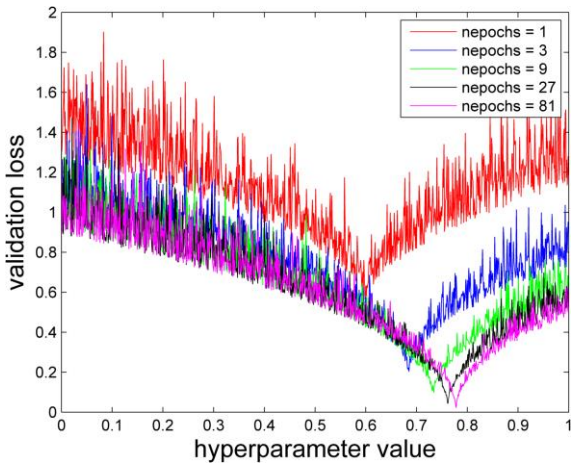
Note that $(1 + |\mathbb{N}(0, \sigma^2)|)$ defines a multiplicative truncated noise term controlled by σ.

Note that $\frac{0.5}{n_e}$ suggests that the lower bound of $f(x)$ decreases with n_e.

The test function is implemented in

def run_then_return_val_loss(nepochs, hyperparameters, noiselevel)

The provided Python code of scenario #1 evaluates 1001 equispaced x values for different number of epochs $n_e \in [1; 3; 9; 27; 81]$. Make a plot which shows how $f(x)$ depends on x for all considered values of n_e. An example plot for $x_{shift} = 0.2$ is given below.

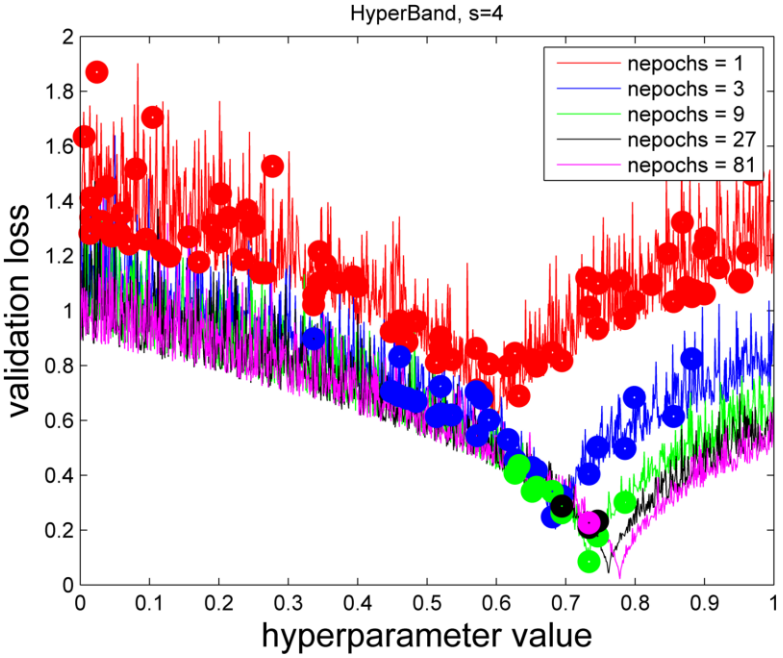


Please make 4 plots for $x_{shift} \in [0;\ 0.2; 0.4; 0.8]$.
Please try to answer the following questions:
What happens with the location of the optimum when different x_{shift} and n_e are considered?
What implications for the hyperparameter search (take into account n_e) might be expected when different x_{shift} are considered?

The source code of scenario #1 also generates some results for random search. They will be used in scenario #4.

Scenario #2. Hyperband.

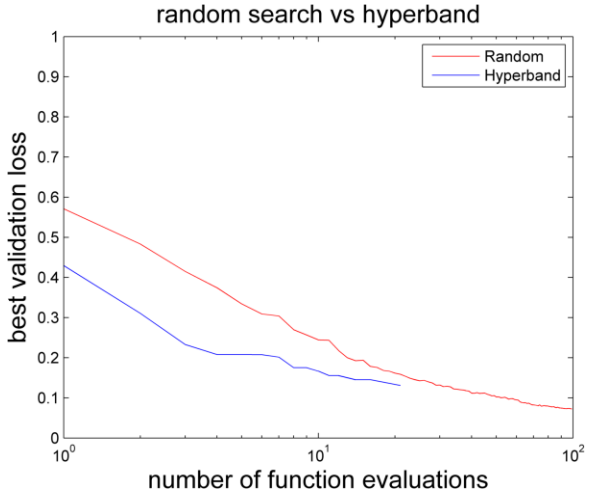
The provided source code runs Hyperband on (1). The algorithm is described in detail at https://people.eecs.berkeley.edu/~kjamieson/hyperband.html
Please investigate / debug the code to understand how it works.
After running Hyperband on (1), make a plot where solutions generated by Hyperband *during the first successive halving / inner loop* are shown on top of random solutions, e.g., as shown below.



Run Hyperband for $x_{shift} \in [0;\ 0.2; 0.4; 0.8]$. Insert the corresponding figures in your report and describe what you observe.

Scenario #3. Hyperband vs Random search.

Note that the provided source code saves solutions generated by random search (see scenario #1) and hyperband (see scenario#2). Use these solutions to plot best validation errors (y-axis) suggested by both algorithms over total computational budget (x-axis; e.g., count 81 epochs in total as one function evaluation). Make a plot showing medians of 100 runs, e.g., see plot below



Make four plots for $x_{shift} \in [0;\ 0.2; 0.4; 0.8]$.
When Hyperband outperforms random search?
When random search outperforms Hyperband?
Explain your results.

Scenario #4. Hyperband on MNIST.

Scenario#5 of the previous lab exercise suggested to run random search on the MNIST dataset. Run Hyperband instead of random search in the same settings and compare the two, i.e., produce a plot showing validation error vs number of function evaluations. Limit each function evaluation in the same way as in Scenario#5 of the previous lab exercise. Note that n_e does not necessarily define the number of epochs, it can also define time in seconds. Note that you can use parallel evaluations on two GPUs as implemented in Scenario#6 of the previous lab exercise.
Does Hyperband outperform random search?
Explain you results.

Scenario #5 (optional). Hyperband on CIFAR-10.

Consider any publicly available implementation / source code which allows to run experiments on the CIFAR-10 dataset, e.g., https://github.com/Lasagne/Recipes/tree/master/papers/densenet . Parameterize the search space and perform hyperparameter search using Hyperband. Please limit the maximum computational budget per function evaluation to, e.g., 1 hour. Compare random search and Hyperband.