Assignment #3
ADL, Spring 2019
Due March 14th

Experiment with hyperparameters

About

In this assignment, you will design and run several short experiments to compare the effect of different learning rates, activation functions, optimizers, and weight initialization strategies. You will produce plots using TensorBoard to compare the results. The goal is to gain familiarity with these choices and their impact.

Submission instructions

Please submit this assignment on CourseWorks by uploading one or more Jupyter notebooks.

- You may submit a single notebook for the entire assignment, or you may submit a zip including one notebook for each part.

- Please be sure your notebooks include saved output that shows the results of training your models.

- Please include a zip of **screenshots** with your submission that show the results of your experiments in TensorBoard.

Code references

Please write your code using the TensorFlow 2.0 preview. The most current references are:

- TensorBoard tutorials (please skip everything not in this directory, most other tutorials are written against older versions of TensorFlow).

- Tutorials and guides (for the TensorFlow 2.0 preview).

Related reading

- Please see the lecture slides (reading is always given on the last slide).

Parts 1 -- 4 (25 points each)

Using any dataset you like (MNIST or similar is recommended so you can iterate quickly), design and run experiments to compare:

1. Three different learning rates.
2. Three different activations.

3. Three different optimizers.
4. Three different weight initialization strategies.

For each experiment, please:

- Visualize your results with TensorBoard (and include a screenshot with your submission).

- Include a brief write up (two paragraphs give or take) summarizing your findings. Are the results what you expected? Why or why not?

Extra credit
You may complete some of all of these problems, in any order.

EC1: Demonstrate the vanishing gradient problem. Write a DNN (say, 10 layers deep) and train it on a simple dataset like MNIST. Choose activation functions, initialization strategies, and an optimizer that are likely to cause this behavior. Produce histograms of the activations and gradients at various during training. What do you see?

- Visualize gradients and activations in Matplotlib or TensorBoard overtime (note: doing so in TensorBoard may require digging deeper into the docs, or borrowing from 1.x code).

- Run a second experiment where you take action to correct this behavior. Visualize and compare the results.

EC2: Provide your own implementation of two activation functions published within the last 18 months, and run experiments to compare the results against a built in method like ReLU. What do you see? Did your results match expectations from the papers?