Machine Learning and Computational Physics
Fall 2020

Assignment 2

Due: 09/21/2020, 11:59:59 PM PDT

Regularization of networks

In this assignment, you will train a deep network to approximate a scalar function. The primary objective is to observe that neural networks can over-fit the training data, thus deteriorating their capacity to generalize.

1. Consider the function developed in Assignment 1 to define a neural network. Extend it so that:

   (a) It additionally accepts the regularization parameter (a scalar) as input.

   (b) The `tanh` activation function is used in each hidden layer, with no activation in the output layer.

   (c) L2 regularization is applied on the weights (but not on the biases).

   (d) The is model compiled using the `Adam` optimizer with a learning rate `lr = 1e-3`, using the `mse` loss function and the `mse` metric.

   The following TensorFlow tools might be useful

   - `keras.regularizers.l2()`
   - `keras.optimizers.Adam()`
   - `keras.model.compile()`

2. Create a dataset using the function

$$f(x) = \sin(10\pi x) + \omega(x)$$

   where $\omega \sim \mathcal{N}(0, 0.2)$ is an additive Gaussian noise at each evaluation point. Evaluate the function at 125 uniformly spaced points between [0,1]. Use every 5th point to create the validation set, and keep the rest as the training set. Thus, the training set will have 100 samples and the validation set will have 50 samples. **Note:** Fix a random generator seed (say to 1), to ensure the same dataset is generated each time the script is run.

3. Consider the regularization parameter values $\{0.0, 1e{-}4, 1e{-}3, 1e{-}2\}$. For each parameter value, create a network with `width = 15`, `depth = 10` and `input_dim = output_dim = 1`. With each configuration, train a network 3 times with different weight initialization with the above defined training and validation data, maximum epochs 5000, batch size 25 and with shuffling on. You will need to use the `model.fit()` command to train.

   At the end of each training, save the history of the metric evaluated on training set (`mse`) and the validation set (`val_mse`). Also, print the final value of these metrics (at the final epoch) for each training). Furthermore, evaluate and save the predictions of the network on the training and validation set.

4. Plot the training and validation metrics for for each network configuration (make sure to plot the history of all 3 re-runs of a configuration on a single plot). Also plot the predictions on the training and validation set.

5. Now answer the following questions:

   (a) In the absence of any regularization, how does the network perform (across all 3 re-trains) on the two datasets?

   (b) How do the results change when regularization is used?

   (c) Is it useful to use a large regularization? Why or why not?

Instructions:

- You need to submit your work as a single notebook saved as `A2_FirstName_LastName.ipnyb` (for example `A2_Tommy_Trojan.ipnyb`). You can create this notebook locally (on your computer using Jupyter notebook) or on cloud using Google Colab (which we recommend). If you are using Google Colab, then please make sure that you are signed in to your USC Google account before starting. This will make sharing your saved work little easier.
- At the very beginning of your notebook insert a text cell and write your name.
- For all parts of Question 5, use individual text cell.
- Make sure that your entire notebook runs successfully on Google Colab before submitting it. It is your responsibility to ensure this.
- Once you finish the assignment save it and share it with `dhruvvpa@usc.edu`. (If you are using Google Colab, then the notebook will automatically be saved to your Google Drive. Once you locate it in your Google Drive, right click on it and share it with `dhruvvpa@usc.edu`). While sharing make sure that you enable "editor" option, so that we can run your notebook on our end while grading it.
- Submit you notebook only once. Resubmissions will not be considered.