APMA 2070 & ENGN 2912V Deep Learning for Scientists and Engineers
Homework 06
Due Date: 04-19-2024, 11:59 pm (E.T.)

Lecture 1.6

1. Implement a CNN via TensorFlow/PyTorch or JAX to learn the MNIST database of handwritten digits. There are different ways to download the dataset. Here, we use MNIST in CSV in a easy-to-use CSV format. Specifically, you need to

   (a) Implement a CNN.

   (b) Train your CNN using the training dataset `mnist_train.csv`.

   (c) After training, test your CNN in the testing dataset `mnist_test.csv` and compute the testing accuracy. It is helpful to check the accuracy during training (e.g., every 100 epochs).

   (d) Tune your hyperparameters to achieve >98% testing accuracy.

   (e) Report your final accuracy and how you achieve it.

   Hint:

   - Normalize the pixel values to $[0, 1]$ by dividing 255.
   - Reshape the input from 1D array of size 784 to a 2D image of size $(28, 28)$, so that you can use 2D convolutional layers.
   - Cross-entropy loss can be computed via tf.keras.losses.SparseCategoricalCrossentropy and torch.nn.CrossEntropyLoss.

2. Write a code in PyTorch or TensorFlow or JAX to approximate the Runge function using deep (e.g., 8 layers) and narrrow (4 neurons) feed forward neural network using ReLU activation function. Please write your observations. (Hint: consult with the Lecture 5 on Neural Network Architecture discussing dying ReLU.)

3. Write a TensorFlow/PyTorch or JAX code to learn the oscillatory function

$$f(x) = \begin{cases} 5 + \sum_{k=1}^{4} \sin(kx), & x \in [-\pi, 0) \\ \cos(10x), & x \in [0, \pi] \end{cases} \tag{1}$$

using a ReLU network. Use a training dataset of 80 uniform data points in $[-\pi, \pi]$, and add a Gaussian noise with mean zero and standard deviation 0.1 to each training point. Train your network with

(a) L^1 regularization,

(b) L^2 regularization,

(c) dropout regularization,

to achieve $< 5\%$ L^2 relative error on a testing dataset of 1000 uniform data points in $[-\pi, \pi]$. Explain what you have done to achieve good accuracy, and write your observations.

4. Approximate the function in (1) using Xavier Normal, Xavier Uniform, He Normal and He Uniform initialization. Please vary the number of layers from 8-16 and observe the learning.

5. 2D dynamical systems are represented by a sequence of images, making the resulting dataset high-dimensional. Learning the inherent patterns from high dimensional datasets can be tricky. Hence, researchers have resorted to dimensionality reduction techniques for obtaining a low dimensional (l_d) representation. However, this comes with a cost – reconstruction error (ϵ), defined as,

$$\epsilon = ||x - \tilde{x}||_2^2,$$

where, x is the true sample, and $\tilde{x}$ is the reconstructed sample from the low dimensional representation.

Consider each image as a sample and split the given dataset randomly to train and test sets in the ratio 80:20. For the given dataset (link), do the following tasks.

(a) Perform Principal Component Analaysis (PCA) and record the reconstruction error on both train and test datasets for different values of the latent low dimension, l_d.

(b) Use an autoencoder for dimensionality reduction and record the reconstruction error on both train and test datasets for the same set of l_d values used in last task.

(c) Plot the reconstruction loss for PCA and autoencoder, against l_d. Provide your observations and the corresponding reasons for those observations.

6. Train a WGAN-GP to generate a uniform distribution in $[-1, 1]^5$. Use a Gaussian distribution of mean 0 and standard deviation 1 as the input noise, and try different dimensionality of the noise.

7. Train a WGAN-GP to generate a uniform distribution in $[-1, 1]^{20}$. Use a Gaussian distribution of mean 0 and standard deviation 1 as the input noise, and try different dimensionality of the noise.

8. Compare the computational cost of 6 and 7 to achieve 5% accuracy of the generated distribution.