S&DS 689 Homework 3

Due on Tuesday, Oct 17, 11:59 pm ET

1. (5 pts) Implement a CNN via TensorFlow/PyTorch/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.
   - Crossentropy loss can be computed via tf.keras.losses.SparseCategoricalCrossentropy and torch.nn.CrossEntropyLoss.

2. (5 pts) Implement a RNN to learn the cosine function

$$f(x) = \cos(2\pi x).$$

   The training dataset is the sequence of $f(x)$ for $x \in [0, 10]$ and $\Delta x = 0.05$, and you need to add a random Gaussian noise (mean 0 and standard deviation 0.02) to each data point. The testing dataset is the sequence of $f(x)$ for $x \in [10, 12]$ and $\Delta x = 0.05$. Specifically, You need to

   (a) Implement a RNN (LSTM).

   (b) Train your RNN using the training dataset.

   (c) After training, predict $f(x)$ for $x \in [10, 12]$, compare with the ground truth, and compute the L^2 relative error.

   (d) Tune your hyperparameters to achieve L^2 relative error <2%.

   (e) Plot the training data, testing data, and your prediction in the same figure. Report your final error and how you achieve it.

   Hint:

   - The RNN input is a sequence of length 10, $[f(x), f(x+\Delta x), \ldots, f(x+9\Delta x)]$. The output is $f(x + 10\Delta x)$.