

Assigned: 28 October

Homework #6

EE 541: Fall 2024

Due: Friday, 08 November at 22:00. Submission instructions will follow separately on BrightSpace.

1. Use PyTorch to train a logistic-classifier for the MNIST dataset of handwritten digits. Use the provided PyTorch DataSet to read from the earlier MNIST data files — `mnist_traindata.hdf5` and `mnist_testdata.hdf5`. Create a `nn.Sequential` network with a single fully-connected (*i.e.*, linear) layer. Choose dimensions to match the multi-class classifier from Homework 3. *Omit* the final soft-max activation (PyTorch implicitly computes this when calculating the loss).

Train your model using a multi-category cross-entropy loss. PyTorch provides the `CrossEntropyLoss` to numerically stabilize the combined `SoftMax-NLLikelihood` calculation. Optimize with (standard) stochastic gradient descent (SGD) and with a mini-batch size of 100. Experiment with ℓ_1 - and/or ℓ_2 -regularization to stabilize training and improve generalization. You may need to experiment the learning rate to improve performance.

Record the log-loss and accuracy of your model on the training set and test set at the end of each epoch. Plot log-loss (*i.e.*, learning curve) of the training set and test set on the same figure. On a separate figure plot the accuracy of your model on the training set and test set. Plot each as a function of the epoch number.

Evaluate the fully trained model on the test data and generate a **confusion matrix** – *i.e.*, find the classification rate conditioned on the true class. Element (i, j) of the confusion matrix is the rate at which the network decides class j when class i is the correct label (ground truth). Use seaborn or similar python package to generate a heatmap showing the confusion matrix.

2. Train two models on Fashion MNIST for 40 epochs:

- (1). One hidden layer with ReLU activation, 128 nodes. No regularization, no dropout.
- (2). One hidden layer with ReLU activation, 48 nodes. L2 regularization with coefficient $\lambda = 0.0001$ and dropout with rate 0.2 at the hidden layer.

You may use the demo scripts from lecture and discussion. Produce histograms for the weights of these two networks – a separate histogram for the two layers (the input and hidden layers) in each case. Describe the qualitative differences between these histograms. What effect does regularization have on the distribution of weights?

3. Train an MLP for CIFAR-10 (<https://www.cs.toronto.edu/~kriz/cifar.html>). The CIFAR-10 dataset consists of 60000 32×32 colour images in 10 classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each 32×32 image consists of three color channels (Red, Green, and Blue) for a total of $3 \times 32 \times 32 = 3072$ pixels. There are 50000 training images and 10000 test images.

Use two hidden layers and ReLU activations. Use 256 nodes in the first hidden layer and 128 nodes in the second hidden layer. Use a dropout layer for each hidden layer output with 30% dropout rate and use an L2 regularizer with coefficient $\lambda = 0.0001$. You may use the demo scripts from lecture and discussion.

Evaluate this trained model on the test data and compute a confusion matrix. Use seaborn or similar python package to generate a heatmap showing the confusion matrix.

- (a) Consider class m . List the class most likely confused for class m for each object type.
- (b) Which two classes (object types) are most likely to be confused overall?