

1 Question: How would you create your model to classify the digits? (Note: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in detail.

2
3 Answers: The following steps are used to create a model that is able to recognize handwritten digits.

- 4 1. Load data and labels from mat files (data: 784x2500, labels: 1x2500).
- 5 2. Reshape data and labels (data:2500x784, labels:2500)
- 6 3. Perform K-Fold Cross Validation to split the dataset into 15 folds.
- 7 4. For each fold,...
- 8 4.1. Split the train dataset into a train dataset and a validation dataset.
- 9 Keep the test dataset the same.
- 10 4.2. Load all three datasets into Dataset classes to convert all values to Tensor.
- 11 4.3. Load train_dataset into a Dataloader class to perform batch training.
- 12 4.4. Define the model, the loss function, and the optimizer.
- 13 4.5. At each epoch, ...
- 14 4.5.1. Feed data from train_dataloader into the model to produce predicted labels.
- 15 4.5.2. Use the loss function to calculate a loss between predicted labels and truth labels.
- 16 4.5.3. Calculate gradients of each model's weights with respect to the loss.
- 17 4.5.4. Use the optimizer to update a model's weights.
- 18 4.5.5. Calculate train_loss and train_accuracy.
- 19 4.5.6. Validate the model with val_dataset and calculate val_loss and val_accuracy.
- 20 4.5.7. If val_loss has not decreased in the last 10 epochs, end the training process.
- 21 4.5.8. Else, train the model until the final epoch.
- 22 4.6. Test the model with test_dataset and calculate test_loss and test_accuracy.
- 23 5. Calculate the average loss and the average accuracy for all three datasets across all folds.

24 Design details:

- 25 • Model: Fully-connected Model
- 26 Given: in_features, num_classes
- 27 Model(
- 28 Linear(in_features=in_features, out_features=32)
- 29 ReLU()
- 30 Linear(in_features=32, out_features=32)
- 31 ReLU()
- 32 Linear(in_features=32, out_features=num_classes)
- 33)

34 Ans: As shown above, the model that is used to recognize handwritten digits is a fully connected model composed of three hidden layers. As with any fully-connected model, each node in a layer is connected to all nodes in the next layer. ReLU is the activation function for the first and the second hidden layers and it serves to introduce nonlinearity into the model.

- 35
- 36 • Loss function: Cross-Entropy Loss

37 Ans: Cross-Entropy is the loss function of choice as it is designed to measure differences between classes. The implementation of Cross-Entropy loss in PyTorch utilizes the sigmoid function which ensures all values fall between 0 and 1 and the negative log loss where loss goes to 0 as input approaches one.

- 38
- 39 • Optimizer: Adam

40 Ans: Adam is an adaptive optimizer that works well in most cases. It utilizes an adaptive learning rate where each weight has its own learning rate and momentum where

learning rates get adjusted based on the gradient direction and time.