DS5220: Supervised Machine Learning and Learning Theory Fall 2022

Problem Set 4

Instructor: Hongyang Ryan Zhang *Due:* ***November 23, 2022, 11:59pm***

Instructions:

- You are expected to write up the solution on your own. Discussions and collaborations are encouraged; remember to mention any fellow students you discussed with when you turn in the solution.
- There are up to three late days for all the problem sets and project submissions. Use them wisely. After that, the grade depreciates by 20% for every extra day. Late submissions are considered case by case. Please reach out to the instructor if you cannot meet the deadline.
- Submit your written solutions to Gradescope and upload your code to Canvas. You are recommended to write up the solution in LaTeX.

Problem 1 (10 points)

(a) (2 points) Explain how convolutional neural networks capture invariant properties of an input image including translation and rotation.

(b) (3 points) What is batch normalization and why is it useful for training deep nets?

(c) (3 points) What happens if the training and testing distributions are different in deep neural networks? How can we address such difference?

(d) (2 points) Why is ReLU activation a good function to be in the hidden layers of deep neural networks compared with sigmoid activation?

Problem 2 (50 points). In this problem, we will set up a two-layer neural network and use the network to train a classifier for recognizing hand-written digits. We consider the MNIST dataset, which consists of 50,000 hand-written digits in the training dataset and 10,000 hand-written digits in the test dataset. Given the training dataset, we construct a two-layer neural network with rectified linear unit activations and use the cross-entropy loss (softmax plus negative log-likelihood) to evaluate the predicted result.

(a) **[15 points]** Let W_1 and W_2 denote the weight matrices of each layer, respectively. Given a sample with feature vector $x \in \mathbb{R}^{28\times 28}$ and label $y \in \{0, 1, 2, \ldots, 9\}$, what is the predicted output y^{pred} of the two-layer neural network? Write down the final prediction y^{pred} for input x as a function of the first layer W_1, b_1 and the second layer W_2, b_2. W_i and b_i refer to the weight matrix and the bias vector of each layer, for $i = 1, 2$. (Hint: Write y^{pred} after applying the softmax operation over the real-valued prediction for every digit type.)

(b) **[15 points]** Let $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ be the training dataset, where $m = 50,000$, $x_i \in \mathbb{R}^{28\times 28}$, and $y_i \in \{0, 1, 2, \ldots, 9\}$ for every $i = 1, 2, \ldots, m$. Write down the training loss of the two-layer neural network as a function of W_1, b_1 and W_2, b_2. (Hint: Write the averaged cross-entropy loss over the training data set; The cross-entropy loss is applied over the prediction y^{pred} and the true label y.)

(c) **[20 points]** Open the notebook for problem 2. Implement the above two-layer neural network using PyTorch. Follow the instructions in the notebook. Report the loss and the prediction accuracy on both the training and test data sets.

Problem 3 (40 points). This problem continues with Problem 2 and instead uses convolutional neural networks to train a classifier for recognizing hand-written digits.

(a) **[15 points]** Recall that a convolutional neural networks involves the following components. Briefly describe the meaning of each component: (1) number of layers; (2) layer type (convolution and fully-connected) (3) filter size for a convolution layer and the max-pooling operation; (4) number of hidden units and activation function for a fully-connected layer. Next, specify 3 common configurations of each component and explain their use cases.

(b) **[20 points]** Open the notebook for problem 3. Implement a convolutional neural network with two convolutional layers and one fully-connected layer in PyTorch. The configuration of each layer is specified below in a sequence from the input to the output. Write the implementation in `CNN(torch.nn.Module)` in the handout. [Hint: The following are recommended layer sizes but feel free to define your own layer sizes too.]

(i) A 2d convolutional layer with 10 filters of size 5x5 with stride 1, zero padding, followed by a ReLU activation, then a 2d max pooling operation with size 2x2.

(ii) A 2d convolutional layer with 20 filters of size 5x5 with stride 1, zero padding, followed by a ReLU activation, then a 2d max pooling operation with size 2x2.

(iii) Fully-connected layer followed by a ReLU activation.

After getting the network output, apply softmax to convert the real-valued prediction to a probability distribution over $\{0, 1, \ldots, 9\}$. Use the negative loss likelihood as the loss function

and SGD the optimizer. Report the loss and the prediction accuracy on both the training and test data sets. Compare the test accuracy of feedforward neural networks to convolutional neural networks.

(c) **[5 points]** Calculate the number of parameters used in both neural networks from problems 2 and 3. Which one is more parameter-efficient? Comment on your findings.