Runqiu Ye
Stanford CS299
Problem Set #4
08/07/2024

Problem Set #3: EM, Deep Learning, & Reinforcement Learning

Problem 1 Neural Networks: MNIST image classification

Implement a simple convolutional neural network to classify grayscale images of handwritten digits from the MNIST dataset. The starter code splits the set of 60000 training images and labels into a sets of 59600 examples as the training set and 400 examples for the dev set. To start, implement convolutional neural network and cross entropy loss, and train it with the provided dataset. The architecture is as follows:

(a) The first layer is a convolutional layer with 2 output channels with a convolution size of 4 by 4.

(b) The second layer is a max pooling layer of stride and width 5 by 5.

(c) The third layer is a ReLU activation layer.

(d) After the four layer, the data is flattened into a single dimension.

(e) The fifth layer is a single linear layer with output size 10 (the number of classes).

(f) The sixth layer is a softmax layer that computes the probabilities for each classes.

(g) Finally, we use a cross entropy loss as our loss function.

The cross entropy loss is

$$CE(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k,$$

where $\hat{y} \in \mathbb{R}^K$ is the vector of softmax outputs from the model for the training example x, and $y \in \mathbb{R}^K$ is the ground-truth vector for the training example X such that $y = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ contains a single 1 at the position of the correct classes.
We also use mini-batch gradient descent with a batch size of 16. Normally we would iterate over the data multiple times with multiple epochs, but for this assignment we only do 400 batches to save time.

(a) Implement functions within `p01_nn.py`.

(b) Implement a function that computes the full backward pass.

■