CSE455/555 - Intro to Pattern Recognition
Problem Set 4: Neural Networks

Due Date: Sunday, April 24, 2022 11:59PM

In this problem set, you will train a neural network to identify the digit on a image in the MNIST data set http://yann.lecun.com/exdb/mnist/ using Tensorflow/Keras or pyTorch. This neural network has 10 softmax output nodes generating $\log p(t = m|x; w)$ where $m = 0,1,\ldots,9$. Let $x_n \in R^{28\times28}$ be the 28 × 28 images, t_n be the label of the image x_n, w be the synaptic weights of the neural network, and n be the index of an image in the training data set.

1 Task-1(Theoretical)

(a) Demonstrate that a neural network to maximize the log likelihood of label is one that has softmax output nodes and minimizes the criterion function of the negative log probability of training data set: $J_0(w) = -\log p(\{(x_n, t_n): n = 1,2,\ldots\}; w) = -\log \prod_n \prod_{m=0,\ldots,9} p(t_n = m|x_n; w)$

(b) Demonstrate that a neural network to maximize the posterior probability of observing the training data given a Gaussian prior of the weight distribution $p(w; \alpha) = N(0, \alpha I)$ is one that minimizes the criterion function with $L2$ regularization $J(w) = J_0(w) - \log p(w; \alpha - 1)$.

2 Task-2(Coding)

(a) Build a neural network (without regularization) with 1 hidden layer of 30 sigmoid nodes, and an output layer of 10 softmax nodes. Train the neural network with 1000 training images (100 images per digit) for 30 complete epochs, using mini-batches of 10 training examples at a time and a learning rate of η = 0.1. Plot the training error, testing error, criterion function/loss on training data set, criterion function/loss on testing data set of a separate 1000 testing images (100 images per digit), and the learning speed of the hidden layer (the average absolute changes of weights divided by the values of the weights). If you are confused about loss and error, check this link: https://stackoverflow.com/questions/38303348/what-is-the-difference-between-the-train-loss-and-train-error

(b) Repeat Task-2(a) with 2 hidden layers of 30 sigmoid nodes each, 3 hidden layers of30 sigmoid nodes each, and with and without $L2$ regularization $\lambda|w|^2$ and λ = 5. (i.e., you will repeat Task-2(a) for 5 times: one for 2 hidden layer network without regularization; one for 3 hidden layer network without regularization; and one times each for 1, 2, 3 hidden layers with regularization.) Note that it is okay to have low accuracies in this part. The purpose is to see how hyperparameter can affect the performance.

(c) First, construct and train convolutional neural network without regularization for MNIST classification. Then, regularize the training of the neural network through dropout (randomly sets elements to zero to prevent overfitting). Finally, regularize the training of neural network through augmenting your selection of 1000 images by rotating them for 1-3 degrees clockwise and counter clockwise, and shifting them for 3 pixels in 8 different.

You can find many tutorials on those techniques, and our emphasize is that we understand those techniques.
Tutorials on dropout:https://www.tensorflow.org/api_docs/python/tf/nn/dropout
Tutorials on image augmentation:
https://machinelearningmastery.com/image-augmentation-deep-learning-keras/
https://keras.io/api/preprocessing/image/
https://opensource.com/article/19/3/python-image-manipulation-tools

Submission

Submit your solutions as one ipynb file through UBlearn. Use Google Colab:
https://colab.research.google.com/notebooks/intro.ipynb
https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c. The ipynb file should include your code, execution results, any explanations and answers to the questions. Use text cells to answer questions and add explanations. Markdown guide for text cells:
https://colab.research.google.com/notebooks/markdown_guide.ipynb#scrollTo=Lhfnlq1Surtk
https://colab.research.google.com/notebooks/basic_features_overview.ipynb#scrollTo=4hfV37gxpP_c
You can also add math to text cells using LaTeX. Just place the statement within a pair of $ signs. Please typeset your mathematics. Do not upload pictures of handwriting math formulas. Math typesetting help: https://www.codecogs.com/latex/eqneditor.php

Libraries

- Allowed:

Tensorflow/Keras/Pytorch are allowed.
Basic libraries are allowed, such as gzip, pickle, math, numpy, scipy, matplotlib, etc. Image manipulation tools are allowed, such as the tools in this link: https://opensource.com/article/19/3/python-image-manipulation-tools

- Not allowed:

Do not use any Python libraries/toolboxes, built-in functions, or external tools/libraries that directly perform classification.

Do not use libraries like sklearn/scikit-learn to construct the neural network. (However, scikit-image is ALLOWED for image manipulation.)

Rubric

Total: 10 points

Task-1: 2 points: 1 point for part (a), 1 point for part (b).
Task-2: 8 points: 2.5 points for part (a), 2.5 for part (b), 3 point for part (c)

Academic Integrity

Academic integrity is a fundamental university value. Any violation will be reported to the University and will result in penalties in grades. Do not share your answers with other students. This is an individual assignment. You are not allowed to work in groups. Working in groups and submitting similar answers is considered a violation of academic integrity. Do not plagiarize someone else's words, ideas, or data you find online. Always cite your sources.