ECE 411 Homework 4 (Fall 2023)
Instructor: Dr. Chau-Wai Wong
TA in Charge: Mr. Prasun Datta
Material Covered: Modern ML applications

Problem 1 (20 points) [Softmax Function] Given an input image, a neural network extracts a sequence of features $\mathbf{z} = (z_1, \ldots, z_K)$. The softmax output for the ith feature z_i is given by

$$\sigma_i(\mathbf{z}) = \frac{\exp(\beta z_i)}{\sum_{j=1}^{K} \exp(\beta z_j)},$$

where β is a positive integer.

a) When $\mathbf{z} = (1, 2, 3, 4, 5)$, use your favorite programming language to calculate and plot $\sigma_i(\mathbf{z})$ as a function of i in bar charts when β takes values of 0.1, 1, and 10, respectively. Based on the empirical results, could you guess what the role of β is?

b) Prove that $\Big(\sigma_1(\mathbf{z}), \ldots, \sigma_K(\mathbf{z})\Big)$ is a valid probability mass function.

c) When z_1 is the largest feature value, prove that $\sigma_1(\mathbf{z}) = 1$ as $\beta \to \infty$.

d) When z_1 is the largest feature value, prove that $\sigma_j(\mathbf{z}) = 0$ for $j = 2, \ldots, K$ as $\beta \to \infty$.

e) Show that $\sigma_j(\mathbf{z}) = 1/K$ for all $j \in [1, K]$ when $\beta = 0$.

f) How are the results in c)–e) connected to your guess about the role of β in a)?

Problem 2 (20 points) [Simple Neural Network for Data with Nonlinear Decision Boundaries] You are going to play with the code of a simple neural network that does binary classification. Open *the Colab notebook file* using Google Drive. Quickly scan through the whole document to get a high-level idea, and then sequentially run the code blocks by clicking the play button on the top left corner of each code block.

a) Draw three convergence curves in one plot for `learning_rate` $= 10^{-3}, 10^{-4},$ and 10^{-5}. The x-axis should be the iteration number and the y-axis should be the loss/training error. For smaller learning rates/step sizes, you may want to increase `num_of_iter` to allow the curve to flatten out.

b) *This Colab notebook file* will walk you through implementing a simple linear regression model $y = 3x + 1$ using a PyTorch neural network. Fill in the missing lines of code around `criterion` and `outputs` by learning the syntax from part a) and reading the PyTorch Documentation, if needed. After that, compare the regression lines under different loss functions, e.g., `nn.L1Loss()`, `nn.MSELoss()`, and under different noise variance within the range $[1.5, 5]$. Submit the plots and key lines of your source code.

Problem 3 (20 points) [Neural Network for Classifying Digits] Open *the Colab notebook file* using Google Drive. Quickly scan through the whole document to get a high-level idea, and then sequentially run the code blocks by clicking the play button on the top left corner of each code block. Examine how the following factors affect the convergence rate and test accuracy:

a) Learning rate

b) Number of epochs

c) Batch size

d) Number of hidden units

In your opinion, what is the best combination of the parameters that leads to a reasonable trade-off between accuracy and convergence time?

Problem 4 (20 points) [BERT for Sentiment Classification] In *this Colab notebook file*, you will use a pretrained BERT for feature extractor and add a softmax layer on top of BERT for binary classification of movie reviews. The softmax layer takes the final output of BERT as input and classifies sentences as either positive or negative (1 or 0, respectively). Note that in the code, the softmax layer is implemented using logistic regression. Examine the following items:

a) Write your own movie review, put it into the pipeline, and see whether the result is positive or negative.

b) Train the softmax layer/logistic regression model with different sized training datasets and evaluate the performance.

c) The last part of the code has three simple functions: `get_attentions`, `plt_attentions`, and `plt_all_attentions`, which enable us to visualize attention maps providing semantic relations of how each of the words pays attention to other words. Note that each row of the attention maps in this code is horizontally normalized by softmax functions. (This is different from the attention map in our lecture note where each column is vertically normalized by softmax functions.) Plot all the 144 attention maps for your movie review and skim through them to find some with good variations of attention weights. Pick one attention map that you believe to have good variations in the heatmap and submit it with your own interpretation/explanation.