



Summer 2 2024, Khoury College, Northeastern University

## Problem Set + Programming Assignment 3

### Instructions

- Hand-drawn figures are acceptable only where specified in the question, and provided all labels are clear and legible. If we can't read your text, we can't assign points. Please scan and insert any such figures in the final PDF document.
- Any use of GenAI that is **solely** intended to improve writing clarity or sentence structure is acceptable in the written sections, but must be accompanied by an appendix containing your original, unedited answers. If you use generative AI in this manner, start a new page titled 'Appendix' at the end of the written submission, and paste your original answers here with the corresponding question numbers clearly indicated.
- If you discuss this assignment with one or more classmates, all parties must declare collaborators in their individual submissions within a code comment. Such discussions must be kept at a conceptual level, and no sharing of actual code is permitted.
- You may use any generative AI tool available to you for the programming section, as long as it is appropriately cited in a code comment. **However, for this assignment I strongly urge you not to rely on GenAI tools, but rather write your own code based on PyTorch documentation for a meaningful learning experience.**

### Deadlines

- Submissions should be uploaded to Gradescope by 6:00 PM on 8/11/2024. Please upload your completed `fashionmnist.py` file, the saved model (namely the `fmnist.pth` file) and a PDF containing your solutions to the written section.
- Gradescope will show a 'late' deadline of 8/14. This is intended solely for any students who may wish to invoke the freebie, outlined in the course policies.
- Any submissions received after 6:00 PM on 8/11 will be considered late, and will automatically invoke the use of your freebie. If you have used your freebie on a previous assignment, your submission will not be accepted for credit.
- Regrade requests must be submitted on Gradescope **before 6:00 pm on 8/17**, in order to help me meet faculty grading deadlines.

## Grading Split

The written portion of this assignment contributes 30 points towards the part of your grade based on Problem Sets. The code section amounts to 35 points towards the part of your grade based on Programming Assignments.

## Reach Out!

If at any point you feel stuck with the assignment, please reach out to the TAs or the instructor, and do so early on! This lets us guide you in the right direction in a timely fashion and will help you make the most of your assignment.

---

## Loss, Gradients, and PyTorch

### 1.1 Written Portion - answer within PDF

**Preliminaries:** Recall our discussion of loss functions from class. For a multi-class classification problem, given a prediction model  $f$ , a common loss function is the Categorical Cross Entropy loss, which is calculated as follows:

$$L(\mathbf{x}_i, \mathbf{y}_i) = -\mathbf{y}_i \cdot \log f(\mathbf{x}_i)$$

Here,  $\mathbf{y}_i$  is a one-hot vector (i.e. a vector of length  $m$ , where  $m$  is the number of classes) with an entry of 1 in the position corresponding to the true class and 0 elsewhere. Similarly, the model output  $f(\mathbf{x}_i)$  is a vector, containing the probabilities that the input belongs to each of the classes.

**Q1.** Consider a logistic regression ( $\hat{y}_i = \sigma(\mathbf{w} \cdot \mathbf{x}_i + b)$ ) with two classes (0 and 1). Given the input example,  $\mathbf{x}_i = [6, 4, 2, 3, 1]$ , model weights  $\mathbf{w} = [0.5, 0.1, -0.8, 0.9, 0.0]$ , bias term,  $b = 0.05$  and the true label encoded as  $\mathbf{y}_i = [0, 1]$ , compute the cross entropy loss and show every step of your calculation. (Hint: a logistic regression gives you the probability of the '1' class.) (4)

**Q2.** The Rectified Linear Unit (ReLU) activation function is defined as:

$$ReLU(x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases}$$

Now consider the neural network in the figure below, with 3 possible classes instead. Assume that the hidden layer uses the ReLU activation function instead of the sigmoid function discussed in class. The final layer uses the sigmoid (logistic function) activation function, followed by a [Softmax](#) operation to ensure that the outputs add up to 1.

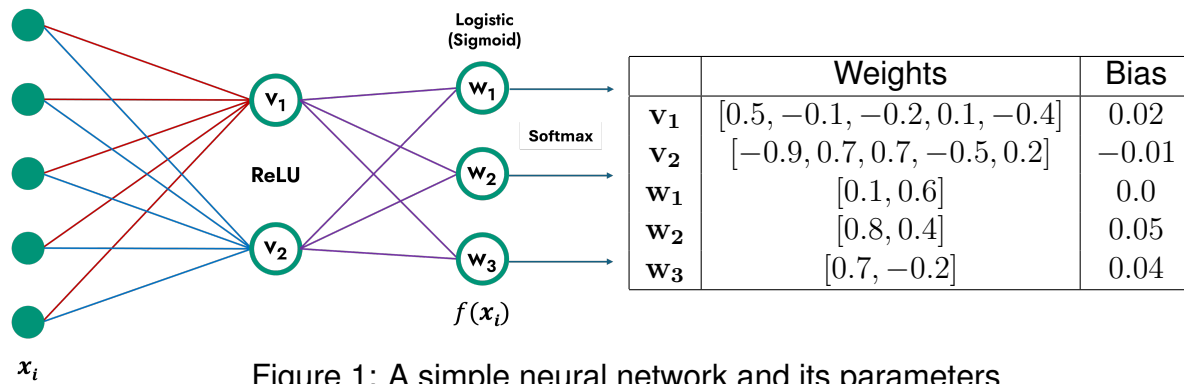


Figure 1: A simple neural network and its parameters

For  $x_i = [5, 7, 4, 3, 2]$ , and the true class label 2 (where the possible classes are 0, 1, and 2), use the model from the diagram, and the given weights table (including nonzero biases this time) to compute the cross entropy loss. You may use NumPy/SciPy operations (but not the loss function methods from these packages) to perform the calculations for this question - if doing so, paste a screenshot of your function(s) and the final result into your written submission. You may find the SciPy expit function useful: [\[link\]](#). (6)

**Q3.** Using a computation graph (of the kind we discussed in class), compute the gradient of the cross entropy loss with respect to the model parameters in the above network. Repeatedly split the loss function by operators like we did in class, and ignore all bias terms. You do not need to split vector/matrix operators into elementwise/row-wise operations. For this question alone, you may scan a hand-drawn figure. Any writing must be perfectly legible. Finally, use NumPy to compute the value of the gradient of the loss function with respect to all parameters ( $w_1, w_2, w_3, v_1, v_2$ ) for the given input and true label. Note that the  $\log$  operator refers to the natural logarithm (base  $\exp$ ). Report your final values and a screenshot of your code. HINT: [\[Derivative of Softmax\]](#) (5 + 5)

## Fashion-MNIST Classification

### 2.1 Implement and train a Neural Network

(30)

**Relevant file:** `fashionmnist.py`

In this section, you will be working with a dataset called Fashion-MNIST. Your task is to design and train a neural network that classifies each image of the dataset correctly. Data is downloaded directly from within the script (using PyTorch). The actual architecture of the neural network is up to you, and convolutional layers are optional.

Fashion-MNIST contains grayscale images of 28 x 28 pixels representing images of clothing. The dataset has 60000 training images, and 10000 testing images, and each image comes with an associated label (e.g. t-shirt, coat, bag, etc.). There are 10 classes, just like the MNIST handwritten digits dataset, so that it may serve as a direct drop-in replacement to test neural networks. Read the full details about this dataset [at the repository](#).

The starter code for this part of the assignment is called "fashionmnist.py". The skeleton code serves as a general guide. There are some sections without any guidelines where you will be expected to research and experiment with various techniques to find a good approach. You must use PyTorch in this section, which is well-documented online.

**Your accuracy on the testing dataset must be greater or equal to 80%.**

## 2.2 Written Portion - answer within PDF

(The points from Q4 and Q5 will be added to the programming portion of the grade for this assignment.)

**Q4.** Submit two figures: A figure containing an image that is classified incorrectly by your model. Include a clear label in this figure that indicates the *predicted* class from your model and the *true* class the image belongs to (both human-readable labels, not just the class number). The second figure should be a single image classified correctly by your model and its corresponding class label. (3)

**Q5.** Submit a plot showing your training loss over time. (2)

**Q6.** Assume both the testing and training datasets have twice as many coat images as shirt images, and answer the following questions: (6)

- a) Would accuracy still be a good metric?
- b) How would you modify your accuracy metric to account for data imbalance?
- c) What other metrics can you use to evaluate model performance?

**Q7.** Calculate the total number of tunable parameters your model has. Show your work, and don't forget the bias terms. Do not use code output to answer this question. (4)