The Chinese University of Hong Kong, Shenzhen
School of Data Science

DDA 3020 · Homework 3
Due: 23:59, April 16th, 2024

Instructions:

- This assignment accounts for 15/100 of the final score.
- You must independently complete each assignment.
- Late submission will get discounted score: 20 percent discount on $(0, 24]$ hours late; 50 percent discount on $(24, 120]$ hours late; no score on late submission of more than 120 hours.

---

1 Written Part (50 pts.)

Problem 1 (12pts) Decision Tree Construction
You want to figure out what makes people click on online advertisements, so you try a variety of different advertisements, and record user behavior in the following table, which is our training set.

Advertisement	Animated?	Popup?	Colorful?	Clicked
1	Yes	Yes	No	No
2	No	Yes	Yes	No
3	No	Yes	No	No
4	No	No	Yes	Yes
5	Yes	No	Yes	Yes
6	Yes	No	No	Yes
7	No	No	Yes	Yes
8	No	No	No	No

The middle three columns ("Animated", "Popup", "Colorful") are the features of each advertisement. The last column, "Clicked", is the label, specifying whether or not a user clicked on the advertisement. You want to predict the "Clicked" label for other advertisements not in this training set.

(a) Which feature should you split on at the root of the decision tree to maximize the information gain? Write an expression for the information gain of the best split. (Your expression can contain logarithms and fractions).

(b) Draw the decision tree that maximizes information gain at each split. Your drawing should include the leaves and the training points they store. (Do not include entropies or information gains). Stop splitting at pure nodes.

Problem 2 (15pts) Computational Graph(CG) and Backpropagation
Consider the following classification MLP with one hidden layer:

$$\begin{aligned}
\boldsymbol{x} &= \text{ input } \in \mathbb{R}^D \\
\boldsymbol{z} &= \mathbf{W}\boldsymbol{x} + \boldsymbol{b}_1 \in \mathbb{R}^K \\
\boldsymbol{h} &= \text{ReLU}(\boldsymbol{z}) \in \mathbb{R}^K \\
\boldsymbol{a} &= \mathbf{V}\boldsymbol{h} + \boldsymbol{b}_2 \in \mathbb{R}^C \\
\mathcal{L} &= \text{CrossEntropy}(\boldsymbol{y}, \text{softmax}(\boldsymbol{a})) \in \mathbb{R}
\end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{b}_1 \in \mathbb{R}^K, \mathbf{W} \in \mathbb{R}^{K\times D}, \boldsymbol{b}_2 \in \mathbb{R}^C, \mathbf{V} \in \mathbb{R}^{C\times K}$, where D is the size of the input, $\boldsymbol{y}$ is the groundtruth, K is the number of hidden units, and C is the number of classes.
hint : $\text{ReLU}(a) = \max(a, 0) = a\mathbb{I}(a > 0)$, where $\mathbb{I}(e)$ is the indicator function.

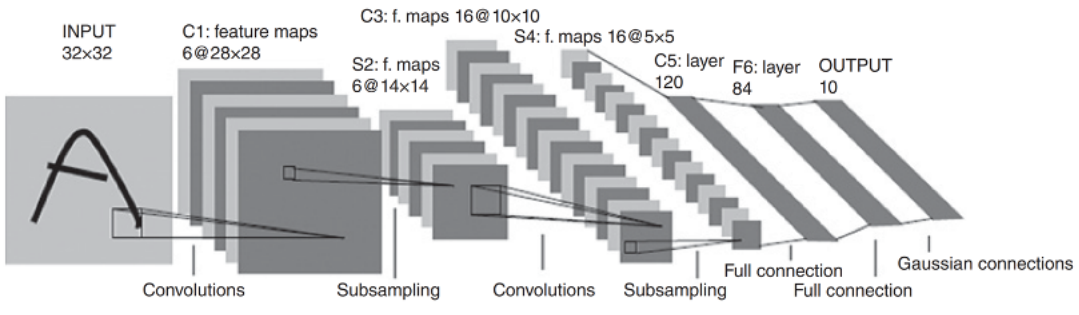
$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

You can use the fact that $\text{ReLU}'(a) = H(a)$, where $H(a)$ is the Heaviside step function $H(a) = \mathbb{I}(a > 0)$.

(a) Please plot the computational graph for the loss function $\mathcal{L}$, i.e. draw the computational graph of forward pass.

(b) Derive the backward pass, i.e. calculate parameter derivative of each layer, including $\nabla_{\mathbf{V}}\mathcal{L}, \nabla_{b_2}\mathcal{L}, \nabla_{\mathrm{W}}\mathcal{L}, \nabla_{b_1}\mathcal{L}$.

Problem 3 (12pts) CNN
Consider the CNN network shown below with two convolution layers (C1 and C3) and two subsampling (max-polling) layers (S2 and S4). During convolution, we assume stride = 1 with no padding. The filter size at each layer is given, where $6@28 \times 28$ means six filter used to generate feature maps with 28×28 each. The full connection layers (C5 and F6) are specified with neuron counts. The input layer takes a 32×32 image filter to each neuron in the C1 layer. The output layer has 10 neurons.



(a) Determine the size of the convolutional filter used in the first layer C1, which equals to the number of input signals that each neuron in C1 receives. Justify your answer.

(b) How many neurons are needed in the convolutional layer C1?

(c) Given the 2×2 filter in the subsampling layer S2, what is the stride distance required for this filter? How many neurons are required in the S2 layer?

(d) What are the purposes in using the hidden layers for convolution and pooling? What are the purposes in using the fully connected layers at the output end?

Problem 4 (11pts) More CNN

(a) List two reasons we typically prefer convolutional layers instead of fully connected layers when working with image data.

(b) Consider the following 1D signal: $[1, 4, 0, -2, 3]$. After convolution with a length-3 filter, no padding, stride $= 1$, we get the following sequence: $[-2, 2, 11]$. What was the filter?

(c) Transpose convolution is an operation to help us upsample a signal (increase the resolution). For example, if our original signal were $[a, b, c]$ and we perform transpose convolution with pad $= 0$ and stride $= 2$, with the filter $[x, y, z]$, the output would be $[ax, ay, az+bx, by, bz+cx, cy, cz]$. Notice that the entries of the input are multiplied by each of the entries of the filter. Overlaps are summed. Also notice how for a fixed filtersize and stride, the dimensions of the input and output are swapped compared to standard convolution. (For example, if we did standard convolution on a length-7 sequence with filtersize of 3 and stride $= 2$, we would output a length-3 sequence).
If our 2D input is $\begin{bmatrix} -1 & 2 \\ 3 & 1 \end{bmatrix}$ and the 2D filter is $\begin{bmatrix} +1 & -1 \\ 0 & +1 \end{bmatrix}$ What is the output of transpose convolution with pad $= 0$ and stride $= 1$?

2 Coding Part (50 pts.)

In the coding part, you will be asked to implement a **Fully-connected Neural Network** and a **Convolutional Neural Network** from scratch using only the package **Numpy**. We use FashionMnist as the main dataset for this exercise. Two csv files **fashion-mnist_train.csv**; **fashion-mnist_test.csv** are provided.

In this exercise, you will use these two networks the address the FashionMnist task. This dataset consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255 . The training and test data sets have 785 columns. The first column consists of the class label, and represents the article of clothing. The

rest of the columns contain the pixel-values of the associated image.

For Fully-Connected Neural Network, you need to implement both the **forward and backward propagation process**, along with defining the **loss function, activation function, and the whole training process**.

For Convolutional Neural Network, you only need to implement the **forward propagation process, including conv_forward, max_pooling and linear_forward**. Codes of other parts are provided. You can train the CNN once you finish the forward part.

Please refer to the **assignment3.ipynb** for more details. Noted that no written report is needed for this coding assignment. We will mark on your effort for implementing the two networks.

Submission Format

- one **pdf file** - containing all your written problem solutions
- one **ipynb file** - containing your code **and the running output** of each step (numbers, plots, etc.). If your notebook has only code but no output results, you will get a discounted score.