September 9, 2022

UNIVERSITY OF ARKANSAS

CSCE 5563: Introduction to Deep Learning

Homework Assignment #1

Submission Deadline: 11:59PM, 10/03/2022

What to submit:

- **1 pdf file** contains your solution and your name. Please don't submit the screenshot/scan of your handwritten solution. Exception is made for hand-drawn graphs. Grading rule: If it's unreadable, I won't grade. Confusing answer gets no point.
- **source code folder** that contains your code. The source code contains readme file on how to run the code.
- Put all the to-be-submitted files into a folder with the name format as:

  {*LASTNAME*}_{*FIRSTNAME*}_homework1

What else to note:

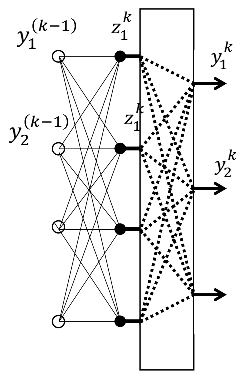
A.. If you don't understand the question, ask the TA immediately at khoavoho@uark.edu or visit AICV Lab (JBHT #447) on Tuesday 1:45PM-2:45PM.

B.. You can discuss with your classmates, DO NOT COPY and please submit your own work.

C.. Solution for this homework is provided within 1-2 weeks after submission deadline or extended deadline, whichever comes later.

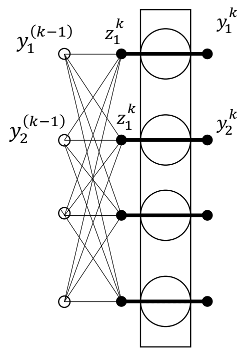
Question 1. (10 points): Derivatives

Given $y^{(0)} = x$, $z^{(k)} = W^{(k)y^{(k-1)}} + b^{(k)}$, $y^{(k)} = f(z^{(k)})$, as follows. Knowing that the number of outputs $y^{(k)}$ maybe different from the number of inputs $z^{(k)}$, i.e. $y^{(k)} \in R^M$ and $z^{(k)} \in R^D$.



a. (5pts) What is Jacobian matrix $J_y^{(k)}(z^{(k)})$?

b. (5pts) If function f is a scalar function as follows, what is Jacobian matrix $J_y^{(k)}(z^{(k)})$?



Question 2. (10 points): Multivariate Quadratic

Given a multivariate quadratic:

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c$$

Where $\mathbf{w} = [w_1, w_2, ..., w_N]$ and $\mathbf{A}$ is diagonal.

Parameters are updated: $\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta\nabla_{\mathbf{w}}E$.

What is the optimal learning rate η ?

Question 3. (15 points): Gradient Descent

a. (5 points) What is Gradient Descent ? What are the pros and cons of Gradient Descent ?

b. (5 points) What is Stochastic Gradient Descent (SGD) ? What are the pros and cons of SGD ?

c. (5 points) What is Mini-Batch Gradient Descent ? What are the pros and cons of Mini-Batch Gradient Descent ?

Question 4. (15 points): Regularization

What is each of the following regularization and how does it help deep neural network ?

a. (3 points) Batch normalization.

b. (3 points) Drop out.

c. (3 points) L1/L2 regularization.

d. (3 points) Data augmentation.

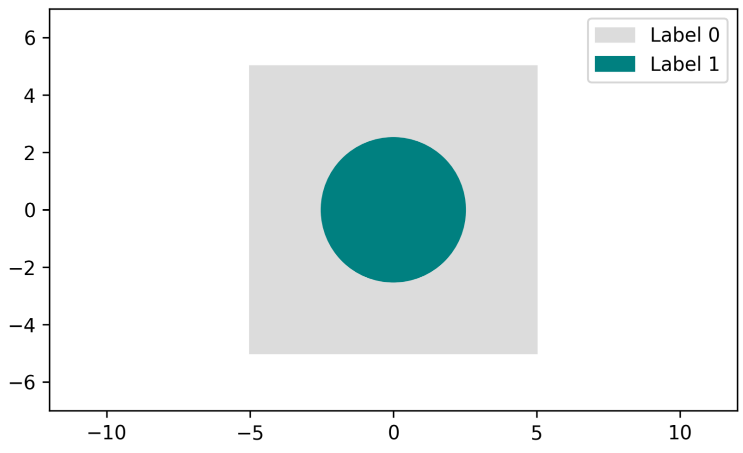
e. (3 points) Early stop.

Question 5. (30 points): Implementation

+ Building a MLP to perform classification task with the following data description:

- Training data: Training data is a set of tuples $\mathcal{D}_{\text{train}} = \{(x_i, y_i) | i \in \mathbb{N} \cap [1, 100\,000]\}$ with the size of $|\mathcal{D}_{\text{train}}| = 100\,000$. Every sample $x_i = (a_i, b_i)$ is a point in 2-dimensional space, which is drawn from a uniform distribution with a range of $a_i, b_i \in [-5, 5]$. Every x_i corresponds to a binary label $y_i \in \{0, 1\}$. $y_i = 1$ if and only if x_i is inside a circle centered by $(0, 0)$ with a radius of 2.5, otherwise, $y_i = 0$ if x_i is outside that circle. See the below figure for a demonstration of training data.
- Validation data: Validation data $\mathcal{D}_{\text{val}}$ follows the exact criterion of $\mathcal{D}_{\text{train}}$, but its size is $|\mathcal{D}_{\text{val}}| = 20\,000$.
- Testing data (X, Y): Testing data $\mathcal{D}_{\text{test}}$ also follows the exact criterion of $\mathcal{D}_{\text{train}}$, but its size is $|\mathcal{D}_{\text{test}}| = 20\,000$.

+ (5 points) Data generation.



+ (25 points) Report training loss, validation accuracy at every epoch by a graph and report the testing accuracy of the final model of the MLP classifier is configured under the following settings:

a. (5 points) MLP with one hidden layer, 3 perceptrons, L2 loss, SGD optimizer, no dropout, 10 epochs.
b. (5 points) MLP with one hidden layer, 3 perceptrons, L2 loss, SGD optimizer, no dropout, 500 epochs.
c. (5 points) MLP with one hidden layer, 128 perceptrons, Cross Entropy loss, Adam optimizer, no dropout, 100 epochs.
d. (5 points) MLP with three hidden layers, (32, 64, 32) perceptrons, Cross Entropy loss, Adam optimizer, no dropout, 100 epochs.
e. (5 points) MLP with three hidden layers, (32, 64, 32) perceptrons, Cross Entropy loss, Adam optimizer, 20% dropout, 100 epochs.

Question 6. (20 points) Hand-written digits classification.

In this question, you will learn a complete process to train various types of deep neural networks on classifying hand-written digits.

Firstly, please download the below ipython notebook file:

`https://colab.research.google.com/drive/1KNBbRfu3qRhVETs5mJhVar9o08bLm1T0?usp=sharing`

Then, upload it to `colab.research.google.com` so that you can edit its source code and run it with GPU support. In order to enable GPU utilization, go to ***Runtime*** > ***Change runtime type*** and select '*GPU*' in the option box under **Hardware accelerator**.

All instructions and descriptions of every sub-task in this question can be found in the ipython notebook file stated in the above URL. The sub-tasks are summarized as follows:

- **TODO 1** (3 points): Complete the function of 'def __init__(root, transform)' by filling missing codes after TODO1 to store contents of MNIST dataset into two lists, i.e., 'self.images' that contains all images of MNIST, and 'self.labels' that contains the corresponding label of each image in 'self.images'.
- **TODO 2** (2 points): Complete the function of 'def __getitem__(index)' by replacing 'pass' after TODO2 with the missing codes to draw the sample at 'index' and its corresponding label. This function should return a tuple (X, y), where X is the image (numpy ndarray of shape 1x28x28) and y is a scalar from 0 to 9 representing X's label.
- **TODO 3** (3 points): During the training process, save best performed model using its averaged accuracy computed on validation set, please fill missing code after TODO3.
  Hint: use 'torch.save(model.state_dict(), PATH)' to save model weights into a file specified by 'PATH'.
- **TODO 4** (3 points): After you finsished training, load the best performed model saved by the above **TODO3**, and evaluate it on testing set to obtain testing loss and testing accuracy, please fill missing code after TODO4.
  Hint: use 'checkpoint = torch.load(PATH)' to load content of file specified in 'PATH' into 'checkpoint', then, use 'model.load_state_dict(checkpoint)' to load parameters saved in 'checkpoint' into 'model'.

- **TODO 5** (3 points): Define ***LeNet*** network and train them using Cross Entropy loss, SGD optimizer, learning rate of 0.001, and in 100 epochs. Saving best performed model on validation subset during training process, and finally evaluate its performance (loss, accuracy) on testing set.
- **TODO 6** (3 points): Define ***VGG16*** network and train them using Cross Entropy loss, SGD optimizer, learning rate of 0.001, and in 100 epochs. Saving best performed model on validation subset during training process, and finally evaluate its performance (loss, accuracy) on testing set.
- **TODO 7** (3 points): Define ***ResNet18*** network and train them using Cross Entropy loss, SGD optimizer, learning rate of 0.001, and in 100 epochs. Saving best performed model on validation subset during training process, and finally evaluate its performance (loss, accuracy) on testing set.

Hint 1: Finish TODO's 1 through 4 before starting TODO's 5 through 7.

Hint 2: An implementation of LeNet can be found at this github repo.

Hint 3: Pytorch's Torchvision library includes very standard implementations of VGG16 and ResNet18.

Important Notice

1. You are expected to submit the ipython notebook for **Question 6**.

2. Please re-name the ipython notebook following 'Q6_⟨LAST NAME⟩_⟨UofA ID⟩.ipynb' before submitting.

3. All TODO's of **Question 6**, especially 3 through 7, are scored by both your source code and their outputs that are executed by you. Please make sure that you already saved their outputs of your source code before you submit the ipython notebook to avoid deducted points.

4. If you have any question, please feel free to contact TA (Khoa Vo) through email khoavoho@uark.edu, or visit AICV Lab (JBHT #447) on Tuesday 1:45PM-2:45PM.