Midterm Exam

AI75823 Deep Learning, Fall 2021
School of BioMedical Convergence Engineering, PNU
Due: Oct. 22. 23:59

I. REMARK

- This is an open book exam.
- There are a total of 100 points in the exam. Each problem specifies its point total.
- You must SHOW YOUR WORK to get full credit.
- If you DISOBEY THE DEADLINE, you will get only 80 % of your full score!!!!
- If you JUST COPY YOUR COLLEAGUE'S, you will get 0 point.
- Answer using Korean or English.

II. PROBLEM SET

1) The task is to create a deep-learning algorithm to distinguish dogs from cats. Implement and show your code in the report (answer sheet). Of course, you can find and modify appropriate codes in any website. No matter what you develop or find the code, describe in detail why you adopt the functions and parameters [40 points].

    a) Go to the 'http://kaggle.com/c/dogs-vs-cats/data' . Download and unzip the file 'train.zip' ,

    b) Load the image files. Randomly select 10,000 dog figures and 10,000 cat figures. Every figure title has a label indicating either dog or cat. Use the 7,000 dog figures and 7,000 cat figures for training, and use the remain for validation.

    c) Preprocess the images for deep-learning. You can resize the image if it is too big. You can use any preprocessing method for facilitating learning process. Show and describe your work.

    d) Design your deep-learning structure. It must be fully-connected networks. (Use not CNN types!!!! If you use CNN, there will be no point !!!!) Show and describe your work. Describe in detail why you design the model, the numbers of layers and nodes, non-linear function, optimizer, loss function, and so on.

    e) How many does your network have the trainable parameters (weights)? Explain how to get the number from your network in detail.

    f) Set up the optimization algorithm, learning rate, batch size and epoch numbers. Graph the training loss and validation loss over epoch. Explain the graph results.

    g) What is your accuracy (% success rate) for the validation data set? Give a shot to obtain maximum accuracy. It must be at least 70%. Save your finally-trained-network and upload corresponding file in plato. Also, upload your code file for training and validating processes. Describe all if you conduct additional processes to improve generalization (ex, any regularization technique..)

    h) Download and unzip the file 'test1.zip'. Label 50 images ('1.jpg', '2.jpg', ... '50.jpg') by yourself. Use the images as test data and obtain classification results through loading your network file and running the network. Upload your test code file in plato so that TA or me can run the file. If it fails in running, the score might be zero. What is your accuracy? It must be also at least 70%.

2) Answer the following questions [10 points].

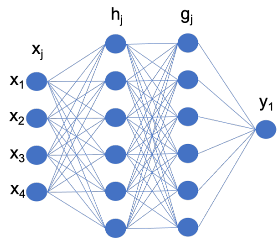
    a) In the class, we learned MSE is an optimal loss function for regression, derived from maximum likelihood estimation (MLE). Likewise, derive that the optimal function for binary classification is binary cross-entropy.

    b) Describe why a sigmoid function is required for an output layer for the classification problem.

3) Explain under-fitting and over-fitting. Assume over-fitting happens. Describe all possible causes and the strategies to prevent them. [5 points]

4) Describe a detail back-propagation algorithm for the network. Assume a simple gradient descent method is used. Also, assume that every layer has no ReLU and

every node has no bias. Assume that batch size is 1 and the loss function is MSE (between estimate y_1 and ground-truth $\bar{y}_1$). Name trainable variables (weights) by yourself. Algorithm 6.2 in Chapter 6 (p.204) in the textbook will help you. You need to describe the computed value of every derivative in detail. [10 points]



5) The task is to investigate that neural nets can compute any function. One function is given as $f(x,y) = 0.2 + x^2 + y^2 + xy + \sin(15x) + \cos(40y)$ where $0 \leq x \leq 1$ and $0 \leq y \leq 1$. [15 points]

   a) Plot the functions $f(x,y)$, $\frac{\partial f(x,y)}{\partial x}$, $\frac{\partial f(x,y)}{\partial y}$ and $\frac{\partial^2 f(x,y)}{\partial x \partial y}$.

   b) Implement deep-learning code for representing the function. The input of the network is $[x,y]^T$ using 2 nodes and the output of the network is $[f(x,y), \frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}, \frac{\partial^2 f(x,y)}{\partial x \partial y}]^T$ using 4 nodes. The design for hidden layers is up to you. It must be the type of fully connected network. The network must be updated through minimizing the error between the estimate and ground-truth. Show your code. Describe your networks and whole training procedures.

   c) For testing, input the pairs in $\{(x_i, y_j) | x_i = 0.01i, y_j = 0.01j, 0 \leq x_i \leq 1, 0 \leq y_i \leq 1\}$ where i and j are integers. Plot the error graphs on the xy domain between the estimates and ground-truths for the functions $f(x,y)$, $\frac{\partial f(x,y)}{\partial x}$, $\frac{\partial f(x,y)}{\partial y}$ and $\frac{\partial^2 f(x,y)}{\partial x \partial y}$.

6) Answer the following questions. [5 points]

   a) Compare the properties between L1 and L2 regularization.

   b) Describe how early stopping acts as regularizer.

   c) Describe bagging.

   d) Describe dropout.

7) The table describes the input-output set of the 3-input Exclusive NOR function. [15 points]

Input A	Input B	Input C	Output Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

   a) Implement the code to represent the function using linear regression.

   b) Implement the code to represent the function using neural networks. Try to make as simplest structure as possible.

   c) Compare the results of linear regression with that of neural networks.