SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Homework #5

Course: *Machine Learning (CS405)* – Professor: *Qi Hao*
Due date: *November 22th, 2019*

Homework Submission Instructions. Please write up your responses to the following problems clearly and concisely. We require you to write up your responses with A4 paper. You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups. However, *each student must write down the solution independently*. You must understand the solution well enough in order to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

- **Written Homeworks.** All calculation problems **MUST** be written on single-sided A4 paper. You should bring and hand in it before class on the day of the deadline. Submitting the scan or photo version on Sakai will **NOT** be accepted.
- **Coding Homeworks.** All coding assignments will be done in Jupyter Notebooks. We will provide a `.ipynb` template for each assignment. Your final submission will be a `.ipynb` file with your answers and explanations (you should know how to write in Markdown or LaTeX). Make sure that all packages you need are imported at the beginning of the program, and your `.ipynb` file should **work step-by-step without any error**.

Question 1

Consider a regression problem involving multiple target variables in which it is assumed that the distribution of the targets, conditioned on the input vector x, is a Gaussian of the form

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{x},\mathbf{w}),\mathbf{\Sigma})$$

where $\mathbf{y}(\mathbf{x},\mathbf{w})$ is the output of a neural network with input vector $\mathbf{x}$ and wight vector $\mathbf{w}$, and $\mathbf{\Sigma}$ is the covariance of the assumed Gaussian noise on the targets.

(*a*) Given a set of independent observations of $\mathbf{x}$ and $\mathbf{t}$, write down the error function that must be minimized in order to find the maximum likelihood solution for $\mathbf{w}$, if we assume that $\mathbf{\Sigma}$ is fixed and known.

(*b*) Now assume that $\mathbf{\Sigma}$ is also to be determined from the data, and write down an expression for the maximum likelihood solution for $\mathbf{\Sigma}$. (Note: The optimizations of $\mathbf{w}$ and $\mathbf{\Sigma}$ are now coupled.)

Question 2

The error function for binary classification problems was derived for a network having a logistic-sigmoid output activation function, so that $0 \leq y(\mathbf{x}, \mathbf{w}) \leq 1$, and data having target values $t \in \{0, 1\}$. Derive the corresponding error function if we consider a network having an output $-1 \leq y(\mathbf{x}, \mathbf{w}) \leq 1$ and target values $t = 1$ for class $\mathcal{C}_1$ and $t = -1$ for class $\mathcal{C}_2$. What would be the appropriate choice of output unit activation function?

Hint. The error function is given by:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}.$$

Question 3

Verify the following results for the conditional mean and variance of the mixture density network model.

(a)

$$\mathbb{E}[\mathbf{t}|\mathbf{x}] = \int \mathbf{t} p(\mathbf{t}|\mathbf{x}) \mathrm{d}\mathbf{t} = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mu_k(\mathbf{x}).$$

(b)

$$s^2(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \{\sigma_k^2(\mathbf{x}) + \|\mu_k(\mathbf{x}) - \sum_{l=1}^{K} \pi_l(\mathbf{x}) \mu_l(\mathbf{x})\|^2\}.$$

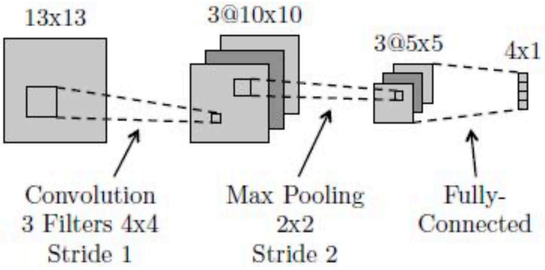
Question 4

Can you represent the following boolean function with a single logistic threshold unit (i.e., a single unit from a neural network)? If yes, show the weights. If not, explain why not in 1-2 sentences.

A	B	f(A,B)
1	1	0
0	0	0
1	0	1
0	1	0

Question 5

Below is a diagram of a small convolutional neural network that converts a 13x13 image into 4 output values. The network has the following layers/operations from input to output: convolution with 3 filters, max pooling, ReLU, and finally a fully-connected layer. For this network we will not be using any bias/offset parameters (b). Please answer the following questions about this network.



(*a*) How many weights in the convolutional layer do we need to learn?

(*b*) How many ReLU operations are performed on the forward pass?

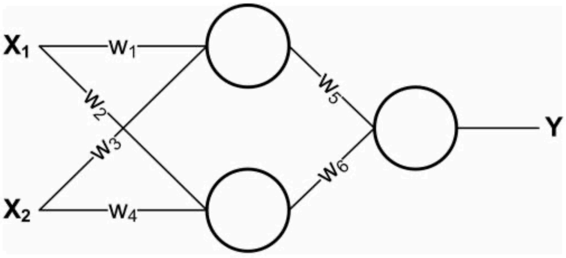
(*c*) How many weights do we need to learn for the entire network?

(*d*) True or false: A fully-connected neural network with the same size layers as the above network ($13 \times 13 \to 3 \times 10 \times 10 \to 3 \times 5 \times 5 \to 4 \times 1$) can represent any classifier?

(*e*) What is the disadvantage of a fully-connected neural network compared to a convolutional neural network with the same size layers?

Question 6

The neural networks shown in class used logistic units: that is, for a given unit U, if A is the vector of activations of units that send their output to U, and W is the weight vector corresponding to these outputs, then the activation of U will be $(1 + \exp(W^T A))^{-1}$. However, activation functions could be anything. In this exercise we will explore some others. Consider the following neural network, consisting of two input units, a single hidden layer containing two units, and one output unit:



(a) Say that the network is using linear units: that is, defining W and and A as above, the output of a unit is $C * W^T A$ for some fixed constant C. Let the weight values w_i be fixed. Re-design the neural network to compute the same function without using any hidden units. Express the new weights in terms of the old weights and the constant C.

(b) Is it always possible to express a neural network made up of only linear units without a hidden layer? Give a one-sentence justification.

(c) Another common activation function is a theshold, where the activation is $t(W_T A)$ where $t(x)$ is 1 if $x > 0$ and 0 otherwise. Let the hidden units use sigmoid activation functions and let the output unit use a threshold activation function. Find weights which cause this network to compute the XOR of X_1 and X_2 for binary-valued X_1 and X_2. Keep in mind that there is no bias term for these units.

pre-Program Question

Answer the following questions in `.ipynb` file before beginning your program questions.

(*a*) You have an input volume that is $63 \times 63 \times 16$, and convolve it with 32 filters that are each 7×7, using a stride of 2 and no padding. What is the output volume? (hint : the third dimension of each filter spans across the whole third dimension of the input)

(*b*) Suppose your input is a 300×300 color (RGB) image, and you are not using a convolutional network. If the first hidden layer has 100 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)?

(*c*) Using the following toy example, lets compute by hand exactly how convolutional layer works.

$$\textit{input image} = \begin{bmatrix} 4 & 4 & 1 & 3 & 2 \\ 2 & 2 & 4 & 1 & 2 \\ 5 & 1 & 2 & 5 & 1 \\ 2 & 1 & 5 & 2 & 4 \\ 4 & 3 & 4 & 5 & 1 \end{bmatrix}$$

$$\textit{filter 1} = \begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 2 \\ 1 & 1 & 2 \end{bmatrix}$$

$$\textit{filter 2} = \begin{bmatrix} 2 & -2 & 1 \\ -1 & 0 & 2 \\ 3 & -2 & 0 \end{bmatrix}$$

Here we have a $5 \times 5 \times 1$ input image, and we are going to use 2 different filters with size $3 \times 3 \times 1$ and stride 1 with no padding as our first convolutional layer. Compute the outputs and complete table. (hint: the output dimension is $3 \times 3 \times 2$)

Row	**Column**	**Filter**	**Value**
1	1	1	-
1	1	2	-
1	2	1	-
2	1	1	-

(*d*) Let's train a fully-connected neural network with 9 hidden layers, each with 20 hidden units. The input is 30-dimensional and the output is a scalar. What is the total number of trainable parameters in your network?

(*e*) State two advantages of convolutional neural networks over fully connected networks.

Program Question

For this question, refer to the Jupyter Notebook. You will be using PyTorch to implement a convolutional neural network the notebook will have detailed instructions. We will be using the fashion MNIST dataset for a classification task.

1. **Convolutional Neural Network**
2. **Network Architecture and Implementation**

   This table describes the baseline architecture for the CNN. Please implement this architecture. You are, however, free to change the architecture as long as you beat the accuracy of this baseline.

Layers	Hyperparameters
Convolution 1	Kernel = (5, 5, 16); Stride=1; Padding=2
ReLU 1	-
Maxpool 1	Kernel size=2
Convolution 2	Kernel = (5, 5, 32); Stride=1; Padding=2
ReLU 2	-
Max pool 2	Kernel size=2
Dropout	Probability=0.5
Fully Connected Layer	Output Channels=10; followed by Softmax

   If you are using your own network architecture, give an explanation for your choice of network and how it may be better than the baseline network.
3. **Accuracy**

   Report the overall accuracy and the per-class accuracy. Identify the problematic classes and list the possible reasons as to why these classes may have significantly lower accuracy compared to other classes.

Reference. These questions are from websites, CMU and UPenn.