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

Homework #5

Course: *Machine Learning (CS405)* – Professor: *Qi Hao*
Due date: *October 7th, 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 Guassian 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.

Answer. In this case, the likelihood function becomes

$$p(\mathbf{T}|\mathbf{X},\mathbf{w},\mathbf{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{t_n}|\mathbf{y}(\mathbf{x_n},\mathbf{w}),\mathbf{\Sigma}),$$

with the corresponding log-likelihood function

$$\ln p(\mathbf{T}|\mathbf{X},\mathbf{w},\mathbf{\Sigma}) = -\frac{N}{2}(\ln|\mathbf{\Sigma}| + K\ln(2\pi)) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t_n}-\mathbf{y_n})^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{t_n}-\mathbf{y_n}),$$

where $\mathbf{y_n} = \mathbf{y}(\mathbf{x_n},\mathbf{w})$ and K is the dimensionality of $\mathbf{y}$ and $\mathbf{t}$.
If we first treat $\mathbf{\Sigma}$ as fixed and known, we can drop terms that are independent of $\mathbf{w}$, and by changing the sign we get the error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t_n}-\mathbf{y_n})^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{t_n}-\mathbf{y_n}).$$

By rewriting the second term we get

$$-\frac{N}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\mathrm{Tr}[\mathbf{\Sigma}^{-1}\sum_{n=1}^{N}(\mathbf{t_n}-\mathbf{y_n})(\mathbf{t_n}-\mathbf{y_n})^{T}].$$

Thus the optimal value for $\mathbf{\Sigma}$ depends on $\mathbf{w}$ through $\mathbf{y}_n$.
A possible way to address this mutual dependency between $\mathbf{w}$ and $\mathbf{\Sigma}$ when it comes to optimization, is to adopt an iterative scheme, alternating between updates of $\mathbf{w}$ and $\mathbf{\Sigma}$ until some convergence criterion is reached.

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:

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

Answer. This simply corresponds to a scaling and shifting of the binary outputs, which directly gives the activation function, in the form

$$y = 2\sigma(a) - 1.$$

The corresponding error function can be constructed by applying the inverse transform to y_n and y_n, yielding

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N} \frac{1+t_n}{2} \ln \frac{1+y_n}{2} + (1-\frac{1+t_n}{2}) \ln(1-\frac{1+y_n}{2}) \\ &= -\frac{1}{2} \sum_{n=1}^{N} \{(1+t_n)\ln(1+y_n) + (1-t_n)\ln(1-y_n)\} + N\ln 2 \end{aligned}$$

where the last term can be dropped, since it is independent of $\mathbf{w}$.
To find the corresponding activation function we simply apply the linear transformation to the logistic sigmoid, which gives

$$\begin{aligned} y(a) &= 2\sigma(a) - 1 = \frac{2}{1+e^{-1}} - 1 \\ &= \frac{1-e^{-a}}{1+e^{-a}} = \frac{e^{a/2} - e^{-a/2}}{e^{a/2} + e^{-a/2}} \\ &= \tanh(a/2). \end{aligned}$$

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)

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

Answer.

$$\begin{aligned} \mathbb{E}[\mathbf{t}|\mathbf{x}] &= \int \mathbf{t} p(\mathbf{t}|\mathbf{x}) \mathrm{d}\mathbf{t} \\ &= \int \mathbf{t} \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathcal{N}(\mathbf{t}|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \mathrm{d}\mathbf{t} \\ &= \sum_{k=1}^{K} \pi_k(\mathbf{x}) \int \mathbf{t} \mathcal{N}(\mathbf{t}|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \mathrm{d}\mathbf{t} \\ &= \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mu_k(\mathbf{x}). \end{aligned}$$

We now introduce the shorthand notation

$$\bar{\mathbf{t}}_k = \mu_k(\mathbf{x}) \qquad \textit{and} \qquad \bar{\mathbf{t}} = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \bar{\mathbf{t}}_k.$$

Then we get

$$
\begin{aligned}
s^2(\mathbf{x}) &= \mathbb{E}[\|\mathbf{t} - \mathbb{E}[\mathbf{t}|\mathbf{x}]\|^2|\mathbf{x}] = \int \|\mathbf{t} - \bar{\mathbf{t}}\|^2 p(\mathbf{t}|\mathbf{x})\mathrm{d}\mathbf{t} \\
&= \int (\mathbf{t}^\mathbf{T}\mathbf{t} - \mathbf{t}^\mathbf{T}\bar{\mathbf{t}} - \bar{\mathbf{t}}^\mathbf{T}\mathbf{t} + \bar{\mathbf{t}}^\mathbf{T}\bar{\mathbf{t}}) \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{t}|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x}))\mathrm{d}\mathbf{t} \\
&= \sum_{k=1}^{K} \pi_k(\mathbf{x})\{\sigma_k^2 + \bar{\mathbf{t}}_\mathbf{k}^\mathbf{T}\bar{\mathbf{t}}_\mathbf{k} - \bar{\mathbf{t}}_\mathbf{k}^\mathbf{T}\bar{\mathbf{t}} - \bar{\mathbf{t}}^\mathbf{T}\bar{\mathbf{t}}_\mathbf{k} + \bar{\mathbf{t}}^\mathbf{T}\bar{\mathbf{t}}\} \\
&= \sum_{k=1}^{K} \pi_k(\mathbf{x})\{\sigma_k^2 + \|\bar{\mathbf{t}}_\mathbf{k} - \bar{\mathbf{t}}\|^2\} \\
&= \sum_{k=1}^{K} \pi_k(\mathbf{x})\{\sigma_k^2 + \|\mu_k(\mathbf{x}) - \sum_{l}^{K} \pi_l \mu_l(\mathbf{x})\|^2\}.
\end{aligned}
$$

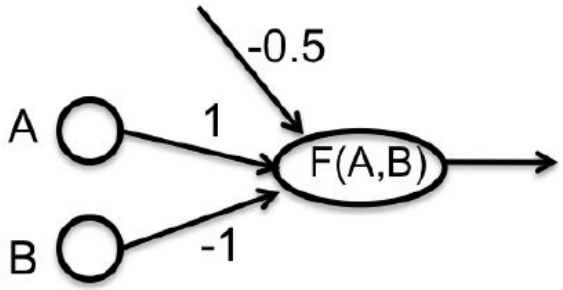
Question 4

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

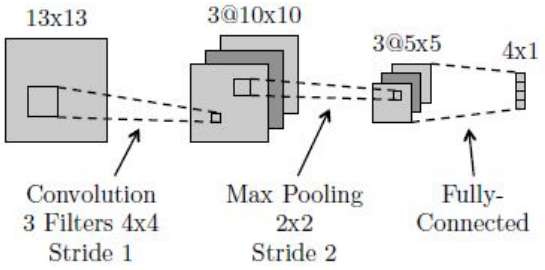
Answer. Yes, you can represent this function with a single logistic threshold unit, since it is linearly separable. Here is one example.

$$F(A, B) = 1\{A - B - 0.5 > 0\}$$



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 (13x13→3x10x10→3x5x5→4x1) can represent any classifier What is the disadvantage of a fully-connected neural network compared to a convolutional neural network with the same size layers?

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

Answer. (a) 48 weights. Three lters with 4x4=16 weights each.

(b) 75 ReLu operations. ReLu is performed after the pooling step. ReLu is performed on each pixel of the three 5x5 feature images.

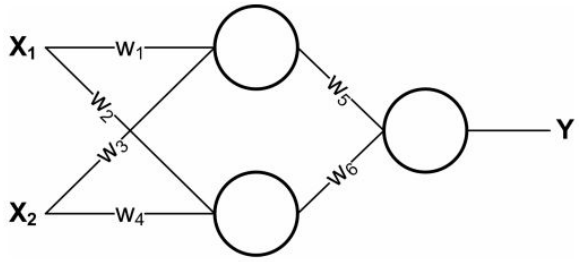
(c) 348 weights. 48 for the convolutional layer. Fully-connected has 3x5x5=75 pixels each connected to four outputs, which is 300 weights. Pooling layer does not have any weights.

(d) True

(e) Too many weights to effectively learn.

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.

Answer. (a) Connect the input for X_1 to the output unit with a weight $C * (w_5 * w_1 + w_6 * w_2)$, and connect the input for X_2 to the output unit with weight $C * (w_5 * w_3 + w_6 * w_4)$. Then the output unit can use the same activation function it used originally

(b) This is true. Each layer can be thought of as performing a matrix multiply to find its representation given the representation on the layer that it receives input from. Thus the entire netowrk just performs a chain of matrix multiplies, and therefore we can simply multiply the matrices together to find the weights to represent the function with a single layer.

(c) One solution: $w_1 = w_3 = 10, w_2 = w_4 = 1, w_5 = 5$, and $w_6 = 6$. The intuition here is that we can decompose A XOR B into (A OR B) AND NOT (A AND B). We make the upper hidden unit behave like an OR by making it saturate when either of the input units are 1. It isnt possible to make a hidden unit that behaves exactly like AND, but we can at least make the lower hidden unit continue to increase in activation after the upper one has saturated.