MACHINE LEARNING 2018

Midterm Exam – Solutions

Feb 8, 2019

- This exam is due at 2 PM, January 5, 2019.
- This exam covers all the materials up to the Lecture 12 - Neural Networks.
- Please submit the exam via Google Form (Link will be sent out shortly). Code for programming problems should be submitted as .py (Python) or

.ipynb (Jupyter Notebook) files. Please clear all the output of Jupyter Notebooks before submission.

- You can discuss exam problems with the instructor, TAs, classmates, or others, but the work you submit must be your own work.
- You may write your answers in Vietnamese or English or a mix of both languages.
- You may consult textbooks and print and online materials.
- Please show all of your work. Answers without appropriate justification will receive very little credit. For programming questions, please submit all the code.

Problem 1. (*5 points*)

A a symmetric $n \times n$ real matrix M is said to be positive definite if we have

$$\mathbf{z}^T M \mathbf{z} > 0 \quad \forall \mathbf{z} \in \mathbb{R}^n,\ \mathbf{z} \neq \mathbf{0} \tag{1}$$

where we use $\mathbb{R}^n$ to denote the the set of n-dimensional real-valued vectors ($\mathbf{z}$ is a is non-zero $n \times 1$ vector). Recall that an eigenvalue λ_i of matrix M is a scalar that satisfies

$$M\mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{2}$$

where $\mathbf{u}_i$ is an eigenvector of M. Show that a necessary and sufficient condition for M to be positive definite is that all of the eigenvalues λ_i of M are positive.

Solution.

From Appendix C – Properties of matrices (Bishop [3]), the eigenvectors of a real symmetric matrix can be chosen to be an orthonormal basis of the vector space $\mathbb{R}^n$. Recall that an orthonormal basis consists of unit vectors (i.e. vectors with unit Euclidean norm) that are orthogonal to each other. That is,

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3}$$

where $i, j = 1, 2, \ldots, n$. Thus any vector $\mathbf{z} \in \mathbb{R}^n$ can be written as

$$\mathbf{z} = \hat{z}_1 \mathbf{u}_1 + \hat{z}_2 \mathbf{u}_2 + \ldots + \hat{z_n} \mathbf{u}_n \tag{4}$$

Note that the basis $\mathbf{u}_i$ is not necessarily the same as the standard basis of $\mathbb{R}^n$, so $\hat{z}_i$'s are not necessarily the same as z_i's, the coordinates of $\mathbf{z}$ in the standard basis. Using Equation (4), $\mathbf{z}^T M \mathbf{z}$ can be written as

$$\mathbf{z}^T M \mathbf{z} = \left(\hat{z}_1 \mathbf{u}_1^T + \hat{z}_2 \mathbf{u}_2^T + \ldots + \hat{z_n} \mathbf{u}_n^T\right) M \left(\hat{z}_1 \mathbf{u}_1 + \hat{z}_2 \mathbf{u}_2 + \ldots + \hat{z_n} \mathbf{u}_n\right)$$

Using Definition (2), it follows that

$$\mathbf{z}^T M \mathbf{z} = \left(\hat{z}_1 \mathbf{u}_1^T + \hat{z}_2 \mathbf{u}_2^T + \ldots + \hat{z_n} \mathbf{u}_n^T\right) \left(\hat{z}_1 \lambda_1 \mathbf{u}_1 + \hat{z}_2 \lambda_2 \mathbf{u}_2 + \ldots + \hat{z_n} \lambda_n \mathbf{u}_n\right)$$

Using Property (3), we then have

$$\mathbf{z}^T M \mathbf{z} = \lambda_1 \hat{z}_1^2 + \lambda_2 \hat{z}_2^2 + \ldots + \lambda_n \hat{z}_n^2$$

If $\lambda_i > 0,\ 1 \leq i \leq n$, it follows that $\mathbf{z}^T M \mathbf{z} > 0,\ \forall \mathbf{z} \in \mathbb{R}^n,\ \mathbf{z} \neq \mathbf{0}$. Thus M is a positive definite matrix. Vice versa, if M is a positive definite matrix, i.e.,

$\mathbf{z}^T M\mathbf{z} > 0,\ \forall \mathbf{z} \in \mathbb{R}^n,\ \mathbf{z} \neq \mathbf{0}$, then $\lambda_i > 0,\ 1 \leq i \leq n$. Otherwise, if $\exists \lambda_i \leq 0$, we can choose $\mathbf{z} = \mathbf{u}_i$, then we have

$$\begin{aligned} \mathbf{z}^T M\mathbf{z} &= \mathbf{u}_i^T M\mathbf{u}_i \\ &= \mathbf{u}_i^T \lambda_i \mathbf{u}_i \\ &= \lambda_i \\ &\leq 0 \end{aligned}$$

which is contradictory to our assumption that M is a positive definite matrix. Thus we have proved the necessary and sufficient condition.

Problem 2. (*20 points*) (Adapted from Cover & Thomas and Yurdakul)
If $p(x)$ and $q(x)$ are two probability mass functions (pmf's), the relative entropy or Kullback-Leibler divergence (or KL divergence), between $p(x)$ and $q(x)$, is defined to be

$$D(p||q) = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)} = E_p log \frac{p(x)}{q(x)} \tag{5}$$

where we can have different bases for the *log* function (the most popular bases being e and 2), and use the convention that $0\ ln\frac{0}{q} = 0$ and $p\ ln\frac{p}{0} = \infty$. X is the set of all possible values of x. The KL divergence is not considered to be a true distance between distributions as it is not symmetric and does not satisfy the triangle inequality.
(a) Prove that the KL divergence defined in Equation (5) is always non-negative and is zero if and only if $p = q$.

Let $X = \{0, 1\}$ and consider two Bernoulli distributions p and q on X. Let $p(0) = 1 - r$, $p(1) = r$, and let $q(0) = 1 - s$, $q(1) = s$.
(b) Derive $D(p||q)$ and $D(q||p)$ as functions of r and s.
(c) Verify that if r=s then $D(p||q) = D(q||p) = 0$.
(d) If $r = 1/2$ and $s = 1/4$, calculate $D(p||q)$ and $D(q||p)$ using log_2 in Equation (5).

In practice, when we have two populations $\hat{p}$ and $\hat{q}$, we normally group values of x into bins and write Equation (5) as

$$D(\hat{p}||\hat{q}) = \sum_{i=1}^{B} \hat{p}_i(x) log \frac{\hat{p}_i(x)}{\hat{q}_i(x)} \tag{6}$$

where B is the number of bins, and $\hat{p}_i$ and $\hat{q}_i$ are proportions of populations $\hat{p}$ and $\hat{q}$, respectively, in bin i. The *Population Stability Index (PSI)* is then defined as

$$PSI(\hat{p}, \hat{q}) = D(\hat{p}||\hat{q}) + D(\hat{q}||\hat{p}) \tag{7}$$

PSI is widely used to measure the difference between

- feature distributions of the training samples and samples being used for the model (the current samples) in a Machine Learning model;
- feature distributions between different points in time;
- outcome distributions.

In practice, there is a general rule of thumb: if PSI between the training samples and current samples is

- less than 10%, the model is considered appropriate;
- between 10% and 25%, we have to investigate the current samples to see why the PSI is so high;
- beyond 25%, we should retrain the model or develop a new model using more recent samples.

(e) Prove that

$$PSI(\hat{p}, \hat{q}) = \sum_{i=1}^{B} [\hat{p}_i(x) - \hat{q}_i(x)] [log(\hat{p}_i(x)) - log(\hat{q}_i(x))] \tag{8}$$

Solution.

(a) (*11 points*) We state below Jensen's inequality (Cover and Thomas [1]), which will be used for the proof.

Theorem 1. *If f is a convex function and X is a random variable, then*

$$E[f(X)] \geq f(E[X]) \tag{9}$$

Moreover, if f is strictly convex, then equality in Equation (9) implies that X is a constant.

Below is a direct corollary of Theorem 1:

Corollary 1. *If f is a concave function and X is a random variable, then*

$$E[f(X)] \leq f(E[X]) \tag{10}$$

Moreover, if f is strictly concave, then equality in Equation (10) implies that X is a constant.

In order to prove Corollary 1, we note that if f is a concave function, then $(-f)$ is a convex function. We can then apply Jensen's inequality (Theorem 1) to $(-f)$ and X.

$$\begin{aligned} -D(p||q) &= -\sum_{x \in X} p(x) log \frac{p(x)}{q(x)} \\ &= \sum_{x \in X} p(x) log \frac{q(x)}{p(x)} \\ &= E_p \left[log \frac{q(x)}{p(x)} \right] \end{aligned}$$

As $log(z)$ a strictly concave of z $\left(\frac{dlog(z)}{z} = -\frac{1}{z^2} < 0\right)$, applying Corollary 1, we have that

$$\begin{aligned} -D(p||q) &\leq log\left(E_p log\frac{q(x)}{p(x)}\right) \\ &= log\left(\sum_{x\in X} p(x)\frac{q(x)}{p(x)}\right) \\ &= log\left(\sum_{x\in X} q(x)\right) \\ &= log(1) \\ &= 0 \end{aligned}$$

Thus $D(p||q) \geq 0$ and the equality holds when $\frac{q(x)}{p(x)}$ is a constant. This is equivalent to $p(x) = q(x)\ \forall x$.

(b) (*2 points*) Using the definition of K-L divergence in Equation (5), we have that

$$\begin{aligned} D(p||q) &= \sum_{x\in X} p(x)log\frac{p(x)}{q(x)} \\ &= (1-r)log\frac{1-r}{1-s} + r\ log\frac{r}{s} \end{aligned}$$

$$\begin{aligned} D(q||p) &= \sum_{x\in X} q(x)log\frac{q(x)}{p(x)} \\ &= (1-s)log\frac{1-s}{1-r} + s\ log\frac{s}{r} \end{aligned}$$

(c) (*1 points*) When $r = s$, we have that $log\frac{1-r}{1-s} = log\frac{r}{s} = log\frac{1-s}{1-r} = log\frac{s}{r} = log1 = 0$. Thus $D(p||q) = D(q||p) = 0$.

(d) (*4 points*) If $r = 1/2$ and $s = 1/4$, using log base 2, $D(p||q)$ and $D(q||p)$ can

be calculated as follows.

$$\begin{aligned} D(p||q) &= (1-r)log\frac{1-r}{1-s} + r\ log\frac{r}{s} \\ &= \frac{1}{2}log_2\frac{1/2}{3/4} + \frac{1}{2}log_2\frac{1/2}{1/4} \\ &= \frac{1}{2}log_2\frac{2}{3} + \frac{1}{2}log_2 2 \\ &= 1 - \frac{1}{2}log_2 3 \\ &\approx 0.2075 \text{ bits} \end{aligned}$$

$$\begin{aligned} D(q||p) &= (1-s)log\frac{1-s}{1-r} + s\ log\frac{s}{r} \\ &= \frac{3}{4}log_2\frac{3/4}{1/2} + \frac{1}{4}log_2\frac{1/4}{1/2} \\ &= \frac{3}{4}log_2\frac{3}{2} + \frac{1}{4}log_2\frac{1}{2} \\ &= \frac{3}{4}log_2 3 - 1 \\ &\approx 0.1887 \text{ bits} \end{aligned}$$

Notes

- If we use natural logarithms, the unit of K-L divergence will be *nats*.

(e) (*2 points*) We have that

$$\begin{aligned} PSI(\hat{p},\hat{q}) &= D(\hat{p}||\hat{q}) + D(\hat{q}||\hat{p}) \\ &= \sum_{i=1}^{B} \hat{p}_i(x)\ log\frac{\hat{p}_i(x)}{\hat{q}_i(x)} + \sum_{i=1}^{B} \hat{q}_i(x)\ log\frac{\hat{q}_i(x)}{\hat{p}_i(x)} \\ &= \sum_{i=1}^{B} \hat{p}_i(x)\,[log(\hat{p}_i(x)) - log(\hat{q}_i(x))] + \\ &\qquad \hat{q}_i(x)\,[log(\hat{q}_i(x)) - log(\hat{p}_i(x))] \\ &= \sum_{i=1}^{B} [\hat{p}_i(x) - \hat{q}_i(x)]\,[log(\hat{p}_i(x)) - log(\hat{q}_i(x))] \end{aligned}$$

Problem 3. (*5 points*) (Adapted from Bishop) It was mentioned in class that we can use *'tanh'* as an activation function for neural networks. It was also mentioned that

$$tanh(a) = 2\sigma(2a) - 1 \tag{11}$$

Using this equation, show that a general linear combination of logistic sigmoid functions of the form

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x - \mu_j}{s}\right) \tag{12}$$

is equivalent to a linear combination of 'tanh' functions of the form

$$y(x, \mathbf{u}) = u_0 + \sum_{j=1}^{M} u_j tanh\left(\frac{x - \mu_j}{2s}\right) \tag{13}$$

Find expressions to relate the new parameters $\{u_0, u_1, \ldots, u_M\}$ to the original parameters $\{w_0, w_1, \ldots, w_M\}$.

Solution.

For the sake of completeness, we also prove identity (11) here (Students can use this identity without giving a proof).

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

Now we start with Equation (12),

$$\begin{aligned} y(x, \mathbf{w}) &= w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x - \mu_j}{s}\right) \\ &= w_0 + \sum_{j=1}^{M} w_j \frac{\tanh\left(\frac{x-\mu_j}{2s}\right) + 1}{2} \\ &= w_0 + \sum_{j=1}^{M} \frac{w_j}{2} + \sum_{j=1}^{M} \frac{w_j}{2} \tanh\left(\frac{x - \mu_j}{2s}\right) \end{aligned}$$

Thus $u_0 = w_0 + \sum_{j=1}^{M} \frac{w_j}{2}$, and $u_j = \frac{w_j}{2},\ j = 1, 2, \ldots, M$.

Problem 4. (*10 points*) (Please look under Section 'Exam', https://piazza.com/vef_academy/winter2018/ml101/resources for the dataset 'uber_usage.csv'). In a survey of Uber customers, a marketing agency interviewed 400 customers that had had at least 2 Uber rides per month in the past 12 months. As can be seen in the file 'uber_usage.csv', they obtained the age (column 'Age') and the annual salary (in USD, column 'AnnualSalary') for each customer (column 'CustomerID'). Based on how often each customer had used Uber, they also labeled the customer as a regular rider (value 1 in column 'RegularRider') or non-regular rider (value 0 in column 'RegularRider'). We want to apply Logistic Regression to classify customers into two classes: Regular Riders and Non-regular riders based on the age and the annual salary.

(a) Split the dataset to training set and test set using the ratio training set : test set = 7 : 3. The utility *model_selection.train_test_split* can be used to split the dataset.

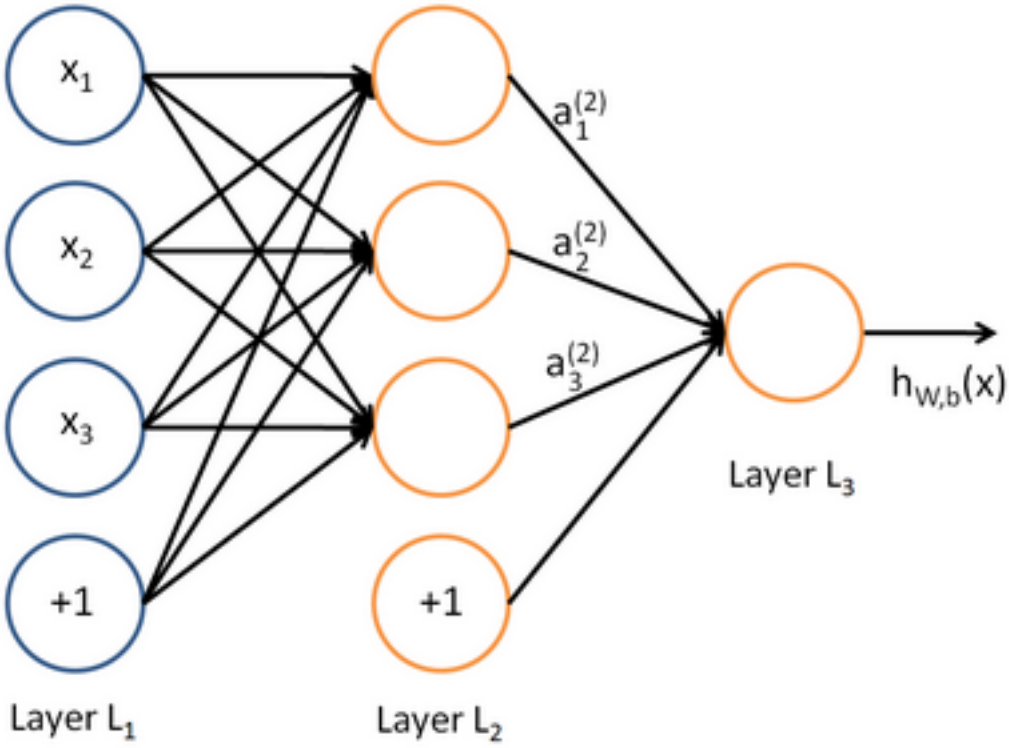
(b) Use scikit-learn's *StandardScaler*, fit and transform each feature (age, annual salary) into a standard normal distribution. Then use the same parameters of the distributions to transform the test set.

(c) Use scikit-learn's Logistic Regression model *linear_model.LogisticRegression* to train a model on the training set and apply it to the test set.

(d) Calculate the TP rate, FP rate, FN rate, Precision, Recall, F1 score, and AUC on the training set and test set. Compare the AUC on the training set and test set.

(e) Visualize the feature values and labels of the training set using a scatter plot. Visualize the feature values and decisions on another scatter plot.

Problem 5. (*10 points*) Using the same dataset in Problem 4, also split the dataset with the same ratio and standardize the feature values with StandardScaler.
(a) Use scikit-learn's SVM model *sklearn.svm.SVC* to train a model on the training set and apply it to the test set.
(b) Calculate the TP rate, FP rate, FN rate, Precision, Recall, F1 score, and AUC on the training set and test set. Compare the AUC on the training set and test set.
(c) Visualize the feature values and labels of the training set using a scatter plot. Also plot the line that is used to separate the two classes.
(d) Visualize the feature values and decisions on another scatter plot. Also plot the line that is used to separate the two classes.
(e) Calculate the PSI between the age distributions in the training set and test set, and the PSI between the annual salary distributions in the training set and test set.



Problem 6. (*50 points*) **(a)** Suppose we have a neural network in the above graph (Figure from [5]).

In this problem, we are going to use the notations from [4], where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are respectively the weight matrix and the bias vector between the $(l-1)^{th}$ layer and l^{th} layer, $l = 1, 2, 3$. Suppose further that we use the sigmoid function as the activation function.

- Write the feedforward equations to calculate the output from the inputs
- Write the backpropgation equations to calculate the gradient of the loss function $J(\mathbf{W}, \mathbf{b}, \mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$.

(b) Using the training set and test set from https://github.com/zalandoresearch/fashion-mnist, build a neural network with 784 inputs and the biases, and 10 outputs. You have to implement the Backpropagation algorithm and Gradient Descent algorithm by yourself.
(c) Calculate the multi-class AUC for the training set and test set using the formulas given in [8].

References

1. Thomas M. Cover and Joy A. Thomas. 2006. Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience, New York, NY, USA.

2. Bilal Yurdakul, Statistical Properties of Population Stability Index (PSI), PhD Dissertation, Western Michigan University, 2018

3. Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

4. Vu Huu Tiep – Machine Learning co ban, https://machinelearningcoban.com/2017/02/24/mlp/

5. Neural Networks – http://ufldl.stanford.edu/wiki/index.php/Neural_Networks

6. M. Pathak, "Introduction to t-SNE", https://www.datacamp.com/community/tutorials/introduction-t-sne

7. Fashion MNIST dataset – https://github.com/zalandoresearch/fashion-mnist

8. Tom Fawcett. 2006. An introduction to ROC analysis. Pattern Recogn. Lett. 27, 8 (June 2006), 861-874. DOI=http://dx.doi.org/10.1016/j.patrec.2005.10.010