Spring18 Deep Learning Midterm Exam

Solutions and Grading Policy Part 1 by Keyi Liu

March 7, 2018

Problem 1 [10 points]

Answer the following questions (True or False).

(1) **(False)**Deep learning mainly resulted from the latest theoretical advances in machine learning.

- Deep learning is inspired by the big data and computing power.

(2) **(False)**Deep model learning involves learning all parameters, including both the weights and the hyper-parameters.

- The hyper-parameters are typically specified manually, and tuned by the cross validation.

(3) **(False)**Deep models always outperform shallow models.

- Given a small number of data, deep models may overfit.

(4) **(False)**Overfitting happens when the model is too complex and data is too much.

- Overfitting happens when we do not have enough training data, in other words, the data is two little to explain a complex model.

(5) **(False)**Underfitting occurs when the model is too simple and training data is too little.

- Underfitting happens when the training data is more than necessary to train a simple model.

(6) **(False)**Regularization is used to improve the classifier's training performance.

- Regularization is used to cure overfitting, thus it can improve the testing performance.

(7) **(False)**Shallow models can perform equally well if given a big training data.

- Shallow models generally do not have as much capacity as deep models.

(8) **(False)**The same training data and the same deep model architecture will yield the same deep model.

- Initialization could be different, therefore, results in a different model.

(9) **(False)**Validation data is part of training data and they are used to tune the hyper-parameters.

- Validation data is not part of training data. They are a separate set of data for tuning hyper parameters.

(10) **(False)**Logistic regression is a special kind of regression, where regression value lies between 0 and 1.

- Since logic regression is for binary classification, it is not regression even though it outputs an intermediate probability, based on which class label is decided.

Problem 2 [20 points 6000 level only]

Given N training data points $\{(x_i, y_i)\}$, where $i = 1, 2, \cdots, N$ and both x_i, and y_i are scalars . For each x_i, the corresponding y_i is sampled from $y_i \sim \mathcal{N}(wx_i + b, \sigma^2 x_i^2)$, assuming the value of σ is the same for all points.

(a) Estimate w and b by minimizing the negative log conditional likelihood.

By definition, the negative log conditional likelihood is defined as,

$$\begin{aligned}
-\mathcal{LCL}(X, Y|w, b) &= -\sum_{i=1}^{N} \log P(y_i|x_i, w, b) \\
&= -\sum_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi}\sigma x_i} exp \left(-\frac{(y_i - wx_i - b)^2}{2\sigma^2 x_i^2} \right) \right] \\
&= \sum_{i=1}^{N} \left[\frac{(y_i - wx_i - b)^2}{2\sigma^2 x_i^2} + \frac{1}{2} \log 2\pi + \log \sigma x_i \right]
\end{aligned} \tag{1}$$

It is obvious that the last two terms are constant with respect to w and b, thus our optimization objective is,

$$\min_{w,b} \ F(w, b) = \sum_{i=1}^{N} \frac{(y_i - wx_i - b)^2}{2\sigma^2 x_i^2} \tag{2}$$

Based on the First Order Necessary Condition(FONC), we can find the minimizer w^* and b^* by setting the first order derivatives to zero.

$$\frac{\partial F(w, b)}{\partial w} = -\sum_{i=1}^{N} \frac{y_i - wx_i - b}{\sigma^2 x_i} \to 0 \tag{3}$$

$$\frac{\partial F(w, b)}{\partial b} = -\sum_{i=1}^{N} \frac{y_i - wx_i - b}{\sigma^2 x_i^2} \to 0 \tag{4}$$

Combine Equation (3) and (4), we can set up a linear system to solve for w^* and b^*,

$$\begin{cases}
-\sum_{i=1}^{N} \frac{y_i}{\sigma^2 x_i} + \left(\sum_{i=1}^{N} \frac{1}{\sigma^2} \right) w + \left(\sum_{i=1}^{N} \frac{1}{\sigma^2 x_i} \right) b = 0 \\
-\sum_{i=1}^{N} \frac{y_i}{\sigma^2 x_i^2} + \left(\sum_{i=1}^{N} \frac{1}{\sigma^2 x_i} \right) w + \left(\sum_{i=1}^{N} \frac{1}{\sigma^2 x_i^2} \right) b = 0
\end{cases} \tag{5}$$

Cancel out σ^2, we can obtain the minimizer,

$$\begin{aligned}
w^* &= \frac{\left(\sum_{i=1}^{N} \frac{y_i}{x_i} \right) \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right) \left(\sum_{i=1}^{N} \frac{y_i}{x_i^2} \right)}{N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2} \\
b^* &= \frac{N \left(\sum_{i=1}^{N} \frac{1}{x_i} \right) \left(\sum_{i=1}^{N} \frac{y_i}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{y_i}{x_i} \right) \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2}{N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2}
\end{aligned} \tag{6}$$

Thus we find the optimal w^* and b^* as desired.

Write down the negative log conditional likelihood function, and plug in the Guassian distribution correctly get **(3 points)**, compute the gradient correctly and setup the linear system get **(5 points)**, compute the result for w^* and b^* get the remaining **(2 points)**.

(b) Find the variance of the estimated w^*.

Directly compute the variance Use the fact that all data points are i.i.d., and y_i, w, and b are independent of each other, then the variance of the sum equals the sum of their variance, also $Var[y_i] = \sigma^2 x_i^2$, first we further simplify w^* to a desired form,

$$w^* = \frac{\sum_{i=1}^{N} \left[\frac{\sum_{j=1}^{N} \frac{1}{x_j^2}}{x_i} - \frac{\sum_{j=1}^{N} \frac{1}{x_j}}{x_i^2} \right] y_i}{N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2} \tag{7}$$

then, the variance of w^* can be found by,

$$\begin{aligned} Var[w^*] &= \frac{Var\left[\sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{N} \frac{1}{x_j^2}}{x_i} - \frac{\sum_{j=1}^{N} \frac{1}{x_j}}{x_i^2} \right) y_i \right]}{\left[N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2 \right]^2} \\ &= \frac{\sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{N} \frac{1}{x_j^2}}{x_i} - \frac{\sum_{j=1}^{N} \frac{1}{x_j}}{x_i^2} \right)^2 Var\left[y_i \right]}{\left[N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2 \right]^2} \\ &= \frac{\sigma^2 \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{1}{x_j^2} - \frac{1}{x_i} \sum_{j=1}^{N} \frac{1}{x_j} \right)^2}{\left[N \left(\sum_{i=1}^{N} \frac{1}{x_i^2} \right) - \left(\sum_{i=1}^{N} \frac{1}{x_i} \right)^2 \right]^2} \end{aligned} \tag{8}$$

If you can get the correct result for w^*, and you take y_i as the random variable, you get **(5 points)**, if you do not have the close form solution for w^*, and compute the variance by definition, you will get **(4 points)**, if you know how to compute the variance, the sum rule of random variables, how the factor impact the variance, you will get another **(5 points)**.

Midterm Exam Solution

1 Problem 3 (20 points)

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial \mathbf{w}}\ell(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m) + \frac{\lambda}{2}\frac{\partial\|\mathbf{w}\|^2}{\partial \mathbf{w}} \tag{1}$$

$$= \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial \mathbf{w}}\ell(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m) + \lambda\mathbf{w}\ , \tag{2}$$

and by defining $z_m = \mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m$ we have

$$\frac{\partial}{\partial \mathbf{w}}\ell(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m) = \frac{\partial \ell(z_m)}{\partial \mathbf{w}} \tag{3}$$

$$= \frac{\partial \ell(z_m)}{\partial z_m}\cdot\frac{\partial z_m}{\partial \mathbf{w}} \tag{4}$$

$$= \begin{cases} -(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m)\mathbf{x}_m & \text{if } |\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m| < 1 \\ -\mathrm{sign}(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m)\mathbf{x}_m & \text{else} \end{cases}\ . \tag{5}$$

Therefore, we have

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \begin{cases} -\dfrac{1}{M}\displaystyle\sum_{m=1}^{M}(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m)\mathbf{x}_m + \lambda\mathbf{w} & \text{if } |\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m| < 1 \\ \\ -\dfrac{1}{M}\displaystyle\sum_{m=1}^{M}\mathrm{sign}(\mathrm{y}_m - \mathbf{w}^T\mathbf{x}_m)\mathbf{x}_m + \lambda\mathbf{w} & \text{else} \end{cases}\ . \tag{6}$$

Correctly calculating the gradient of the regularization term **3 points**, moving the gradient inside the summation **2 points**, correctly using the chain rule on the loss function **5 points**, correctly calculating the derivative of each term **6 points** in total, and you will get **5 points** for correctness of your final answer.

Problem 4 [30 points]

Forward and Backward propagation of a Neural Network. Given the input layer $\mathbf{x} \in \mathbb{R}^{3\times 1}$, the hidden layer $\mathbf{h} \in \mathbb{R}^{2\times 1}$, the weights between input and hidden layer is a matrix $W_1 \in \mathbb{R}^{3\times 2}$, and the weights between the hidden layer and the output is a vector $W_2 \in \mathbb{R}^{2\times 1}$, the output is a single scalar $y \in (0,1)$, a sample is being used (x, y). The linear rectified unit activation function $Relu(\cdot)$ is used in the hidden layer, and the sigmoid function $\sigma(\cdot)$ is used at the output node.

(1) Write out symbolically the value of $\mathbf{h}$ and output $\hat{y}$ as a function input $\mathbf{x}$ and weight matrices through forward propagation.

The value of the hidden layer is,

$$\mathbf{h} = Relu(W_1^T \mathbf{x}) \tag{9}$$

The value of the output node is,

$$\hat{y} = \sigma(W_2^T \mathbf{h}) = \frac{1}{1 + e^{-W_2^T Relu(W_1^T \mathbf{x})}} \tag{10}$$

Correctly espress **h (3 points)**, correctly express $\hat{y}$ **(3 points)**, give the final output with respect to input and weights get the rest **(4 points)**.

(2) Assume the current input value is $\mathbf{x} = (1, 2, 1)^T$, compute numerically the output $\hat{y}$.

Plug in the values into Equation (10) to get the value for $\mathbf{h} = (2, 0)^T$,

$$\mathbf{h} = Relu\left(\begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \tag{11}$$

Plug in the values into Equation (11), we get the value for the predicted output $\hat{y} = 0.5$.

$$\hat{y} = \sigma\left(\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix}\right) = 0.5 \tag{12}$$

Correctly compute **h (2 points)**, correctly get the output value **(3 points)**.

(3) Write out symbolically the gradient of the loss function $\mathcal{L}(y, \hat{y})$ with respect to W_1 and W_2.

The partial derivative with respect to W_2,

$$\begin{aligned} \frac{\partial \mathcal{L}(y, \hat{y})}{\partial W_2} &= \frac{\partial W_2^T \mathbf{h}}{\partial W_2} \frac{\partial \hat{y}}{\partial W_2^T \mathbf{h}} \frac{\partial \mathcal{L}(y, \hat{y})}{\partial \hat{y}} \\ &= \mathbf{h}\sigma(W_2^T \mathbf{h})(1 - \sigma(W_2^T \mathbf{h}))(\hat{y} - y) \end{aligned} \tag{13}$$

Suppose we denote $W_1[k]$ as the k^{th} column of matrix W_1, and $\mathbf{h}[i]$ as the i^{th} element of $\mathbf{h}$, and $W_2[i]$ as the i^{th} element of W_2, thus we have,

$$\begin{aligned} \frac{\partial \mathcal{L}(y, \hat{y})}{\partial W_1[k]} &= \frac{\partial \mathbf{h}}{\partial W_1[k]} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathcal{L}(y, \hat{y})}{\partial \hat{y}} \\ &= \sum_{i=1}^{2} \left(\frac{\partial \mathbf{h}[i]}{\partial W_1[k]} \frac{\partial \hat{y}}{\partial \mathbf{h}[i]} \right) (\hat{y} - y) \end{aligned} \tag{14}$$

where,

$$\frac{\partial \hat{y}}{\partial \mathbf{h}[i]} = \begin{bmatrix} \sigma(W_2^T \mathbf{h})(1 - \sigma(W_2^T \mathbf{h}))W_2[1] \\ \sigma(W_2^T \mathbf{h})(1 - \sigma(W_2^T \mathbf{h}))W_2[2] \end{bmatrix} \tag{15}$$

and where,

$$\frac{\partial \mathbf{h}[i]}{\partial W_1[k]} = \begin{cases} \mathbf{x} & \text{if } W_1[k]^T\mathbf{x} > 0, \text{ and } i = k \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Get the correct result for W_1, get **(10 points)** (get the gradient with respect to the output can get partial 4 points), get the gradient with respect to the $\mathbf{h}$ correctly get **(3 points)**, get the final result correctly for W_2 get **(6 points)**(get gradient for Relu correctly get partial 4 points, write out the symbolic format for the chain rule correctly get 3 partial points), total is 10 points.

(4) Given the initial value of $\mathbf{x}$, and y, we compute,

$$\begin{aligned} \frac{\partial \mathcal{L}(y,\hat{y})}{\partial W_2} &= (\hat{y} - y)\sigma(W_2^T\mathbf{h})(1 - \sigma(W_2^T\mathbf{h}))\mathbf{h} \\ &= \begin{bmatrix} -0.25 \\ 0 \end{bmatrix} \end{aligned} \tag{17}$$

For weight W_1, first the gradient with respect to the hidden layer is,

$$\frac{\partial \hat{y}}{\partial \mathbf{h}[i]} = \begin{bmatrix} \sigma(W_2^T\mathbf{h})(1 - \sigma(W_2^T\mathbf{h}))W_2[1] \\ \sigma(W_2^T\mathbf{h})(1 - \sigma(W_2^T\mathbf{h}))W_2[2] \end{bmatrix} = \begin{bmatrix} 0 \\ 0.25 \end{bmatrix} \tag{18}$$

the other term

$$\begin{aligned} \frac{\partial \mathbf{h}[1]}{\partial W_1[1]} &= \mathbf{x}, \text{ since } W_1[1]^T\mathbf{x} = 2 > 0 \\ \frac{\partial \mathbf{h}[1]}{\partial W_1[2]} &= 0 \\ \frac{\partial \mathbf{h}[2]}{\partial W_1[1]} &= 0 \\ \frac{\partial \mathbf{h}[2]}{\partial W_1[2]} &= 0, \text{ since } W_1[2]^T\mathbf{x} = -1 < 0 \end{aligned} \tag{19}$$

Plug in Equation (14),

$$\begin{aligned} \frac{\partial \mathcal{L}(y,\hat{y})}{\partial W_1[1]} &= \frac{\partial \mathcal{L}(y,\hat{y})}{\partial \hat{y}} \sum_{i=1}^{2} \frac{\partial \mathbf{h}[i]}{\partial W_1[1]} \frac{\partial \hat{y}}{\partial \mathbf{h}[i]} \\ &= (0.5 - 1)\left[\frac{\partial \mathbf{h}[1]}{\partial W_1[1]} \frac{\partial \hat{y}}{\partial \mathbf{h}[1]} + \frac{\partial \mathbf{h}[2]}{\partial W_1[1]} \frac{\partial \hat{y}}{\partial \mathbf{h}[2]}\right] \\ &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{20}$$

similarly,

$$\begin{aligned} \frac{\partial \mathcal{L}(y,\hat{y})}{\partial W_1[2]} &= \frac{\partial \mathcal{L}(y,\hat{y})}{\partial \hat{y}} \sum_{i=1}^{2} \frac{\partial \mathbf{h}[i]}{\partial W_1[2]} \frac{\partial \hat{y}}{\partial \mathbf{h}[i]} \\ &= (0.5 - 1)\left[\frac{\partial \mathbf{h}[1]}{\partial W_1[2]} \frac{\partial \hat{y}}{\partial \mathbf{h}[1]} + \frac{\partial \mathbf{h}[2]}{\partial W_1[2]} \frac{\partial \hat{y}}{\partial \mathbf{h}[2]}\right] \\ &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{21}$$

Therefore the gradient of W_1 is

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial W_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{22}$$

Get the gradient of the hidden layer **(2 points)**, get the final results correctly **(3 points)**.

2 Problem 5 (20 points)

2.1 Convolution (7 points)

The convolution formula for stride of 1 is as follows:

$$C[i,j] = \sum_{m=1}^{3}\sum_{n=1}^{3} F[m,n]X[m+i-1, n+j-1] \; . \tag{7}$$

Applying this on the given image and filter, we get

$$C = \begin{bmatrix} -8 & 12 & 14 \\ -6 & 21 & 7 \\ -3 & 16 & 2 \end{bmatrix} \; . \tag{8}$$

Understanding how to slide the filter over the image **3 points**, applying the stride of 1 **2 points**, and **2 points** for the final answer.

2.2 Activation (4 points)

For the activation layer we only need to apply the ReLU function on each element of the output matrix. Hence, we have

$$R = \begin{bmatrix} 0 & 12 & 14 \\ 0 & 21 & 7 \\ 0 & 16 & 2 \end{bmatrix} \; . \tag{9}$$

2 points for setting the negative entries to zero, and **2 points** for keeping the positive entries.

2.3 Pooling (4 points)

For max pooling, with stride of 1 we have

$$P[i,j] = \max_{m=1,2} \; \max_{n=1,2} \; R[m+i-1, n+j-1] \; , \tag{10}$$

and for matrix R we have

$$P = \begin{bmatrix} 21 & 21 \\ 21 & 21 \end{bmatrix} \; . \tag{11}$$

Each element of the matrix has **1 point**.

2.4 Fully Connected (5 points)

First, we vectroize the output of the pooling layer and then calculate its inner product with the weight vector and find its sigmoid value:

$$\hat{y} = \sigma(W^T P_v) = \sigma(14.7) \approx 1 \; . \tag{12}$$

Vectorizing the output of pooling layer **1 points**, calculate its inner product with the weight vector **2 points**, and applying the sigmoid function **2 points**.