THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY

Machine Learning

Homework 2 Solutions

Due Date: See course website

Your answers should be typed, not handwritten. You can submit a Word file or a pdf file. Submissions are to be made via Canvas. Note that penalty applies if your similarity score exceeds 40. To minimize your similarity score, don't copy the questions.

Question 1 Consider the following dataset:

Instance	y	x_1	x_2
1	1	0	0
2	1	0	0
3	1	0	1
4	1	0	1
5	0	1	0
6	0	1	0
7	1	1	1
8	0	1	1

(a) Give the Naïve Bayes model for the data. There is no need to use Laplace smoothing, and there is no need to show the process of calculation.

(b) Calculate the posterior probabilities of the Instances 1 and 7 belonging to the two classes according to the model of the previous sub-question. Show the process of calculation.

Solution: (a) $p(y=0)=3/8$, $p(x_1=0|y=0)=0$, $p(x_2=0|y=0)=2/3$, $p(x_1=0|y=1)=4/5$, $p(x_2=0|y=1)=2/5$

(b)

$$\begin{aligned}
p(\mathbf{x}_1|y=0) &= p(x_1=0|y=0)p(x_2=0|y=0)=0\\
p(\mathbf{x}_1|y=1) &= p(x_1=0|y=1)p(x_2=0|y=1)=8/25\\
p(y=0|\mathbf{x}_1) &= \frac{p(y=0)p(\mathbf{x}_1|y=0)}{p(y=0)p(\mathbf{x}_1|y=0)+p(y=1)p(\mathbf{x}_1|y=1)}=0\\
p(y=1|\mathbf{x}_1) &= 1-p(y=0|\mathbf{x}_1)=1.
\end{aligned}$$

$$\begin{aligned}
p(\mathbf{x}_7|y=0) &= p(x_1=1|y=0)p(x_2=1|y=0)=1/3\\
p(\mathbf{x}_7|y=1) &= p(x_1=1|y=1)p(x_2=1|y=1)=3/25\\
p(y=0|\mathbf{x}_7) &= \frac{p(y=0)p(\mathbf{x}_7|y=0)}{p(y=0)p(\mathbf{x}_7|y=0)+p(y=1)p(\mathbf{x}_7|y=1)}=\frac{\frac{3}{8}\frac{1}{3}}{\frac{3}{8}\frac{1}{3}+\frac{5}{8}\frac{3}{25}}=\frac{5}{8}\\
p(y=1|\mathbf{x}_7) &= 1-p(y=0|\mathbf{x}_7)=\frac{3}{8}.
\end{aligned}$$

Question 2 [Optional]

Suppose there are K i.i.d training sets $S_k=\{\mathbf{x}_{ki},y_{ki}\}_{i=1}^m$ $(k=1,\ldots,K)$ for a regression problem with

a hypothesis class $\mathcal{H}$. For each k, let

$$h_k = \arg\min_{h\in\mathcal{H}} \hat{\epsilon}(h), \text{ where } \hat{\epsilon}(h) = \frac{1}{m}\sum_{i=1}^{m}(y_{ki} - h(\mathbf{x}_{ki}))^2$$

The variance component of the expected error of h_k is:

$$\text{Var}(h_k) = E_{\mathbf{x}}E_{S_k}[E_{S_k}(h_k(\mathbf{x})) - h_k(\mathbf{x})]^2.$$

Because the training sets are i.i.d, $\text{Var}(h_k)$ is the same for different k. Let $\text{Var}(h_k) = \sigma^2$.

(a) Let $\bar{h} = \frac{1}{K}\sum_{k=1}^{K} h_k$. Show that the variance component of the expected error of $\bar{h}$ is:

$$\text{Var}(\bar{h}) = \frac{1}{K}\sigma^2.$$

(b) Based on part (a), a variance reduction technique called **bagging** is proposed. Find out how bagging works, and explain why it reduces variance.

Solution: (a) Let $e_k(\mathbf{x}) = E_{S_k}[h_k(\mathbf{x})] - h_k(\mathbf{x})$. Then, different e_k's are independent of each other, and $E_{S_k}[e_k(\mathbf{x})] = 0$. Hence, we have:

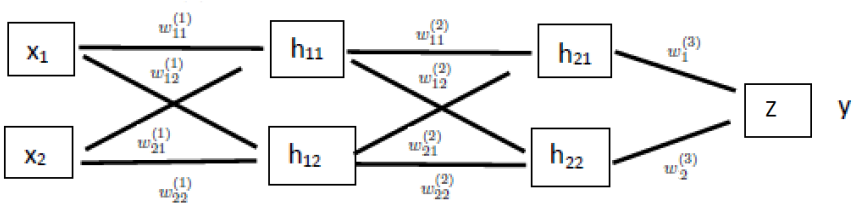
$$\begin{aligned}
\text{Var}(\bar{h}) &= E_{\mathbf{x}}E_{S_1,\ldots,S_k}[E_{S_1,\ldots S_k}(\bar{h}(\mathbf{x})) - \bar{h}(\mathbf{x})]^2 \\
&= E_{\mathbf{x}}E_{S_1,\ldots,S_k}[E_{S_1,\ldots S_k}([\frac{1}{K}\sum_{k=1}^{K} h_k(\mathbf{x})) - \frac{1}{K}\sum_{k=1}^{K} h_k(\mathbf{x})]^2 \\
&= E_{\mathbf{x}}E_{S_1,\ldots,S_k}[\frac{1}{K}\sum_{k=1}^{K} e_k]^2 \\
&= \frac{1}{K^2}E_{\mathbf{x}}[\sum_{k=1}^{K} E_{s_k}[e_k]^2 + \sum_{j\neq k} E_{s_j}[e_j]E_{s_k}[e_k]] \qquad \text{(independence)} \\
&= \frac{1}{K^2}\sum_{k=1}^{K} E_{\mathbf{x}}E_{s_k}[e_k]^2 \qquad (E_{S_k}[e_k(\mathbf{x})] = 0) \\
&= \frac{1}{K^2}\sum_{k=1}^{K} \sigma^2 \\
&= \frac{1}{K}\sigma^2.
\end{aligned}$$

(b) In bagging, we start with one training set $S = \{\mathbf{x}_i, y_i\}_{i=1}^m$. K training sets $S_1, \ldots, S_K$ of the same sample size are created by randomly sampling from S with replacement. Those training sets are called *bootstrap samples*. A regression function h_k is learned from each bootstrap sample, and their average $\bar{h} = \frac{1}{K}\sum_{k=1}^{K} h_k$ is the final output regressor.

For simplicity, assume that each bootstrap sample consists of m data points. Let h be the regressor learned from S. Then, $\text{Var}(h) \approx \text{Var}(h_k) = \sigma^2$. However, $\text{Var}(\bar{h}) \approx \frac{1}{K}\sigma^2$. Hence the variance is reduced by a factor of K.

Note that the conditions of Part (a) are not strictly satisfied, and $\text{Var}(h) > \text{Var}(h_k)$. Hence, the actual reduction is less than the factor of K.

Question 3 Consider the following feedforward neural network with one input layer, two hidden layers, and one output layer. The hidden neurons are **tanh** units, while the output neuron is a sigmoid unit.



The weights of the network and their initial values are as follows:

$$\text{Between input and first hidden:} \quad \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\text{Between two hidden layers:} \quad \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\text{Between second hidden and output:} \quad \begin{bmatrix} w_1^{(3)} \\ w_1^{(3)} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For simplicity, assume the units do not have bias parameters. Let there be only one training example $(x_1, x_2, y) = (1, 2, 0)$.

(a) Consider feeding $(x_1, x_2) = (1, 2)$ to the network. What are the outputs of the hidden units? What is the logit $z = u_{21}w_1^{(3)} + u_{22}w_2^{(3)}$ calculated at the output unit? The output of the output unit is a probability distribution $p(y|x_1 = 1, x_2 = 2, \theta)$. What is the distribution?

(b) Next consider backpropagation. The loss function for the training example is $L = -\log p(y = 0|x_1 = 1, x_2 = 2, \theta)$. What is the error $\frac{\partial L}{\partial z}$ for the output unit? What are the errors for the hidden units? What are $\frac{\partial L}{\partial w_{22}^{(2)}}$ and $\frac{\partial L}{\partial w_{22}^{(1)}}$? If we want to reduce the loss on the example, should we increase or decrease the two parameters?

Solution: (a) Here are the outputs of the hidden units in forward propagation

$$\begin{bmatrix} x_1 = 1 \\ x_2 = 2 \end{bmatrix} \Rightarrow \begin{bmatrix} u_{11} = \tanh(z_{11}) = \tanh(x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)}) = \tanh(1 - 2) = -0.7615 \\ u_{12} = \tanh(z_{12}) = \tanh(x_1 w_{12}^{(1)} + x_2 w_{22}^{(1)}) = \tanh(-1 + 2) = 0.7615 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u_{21} = \tanh(z_{21}) = \tanh(u_{11} w_{11}^{(2)} + u_{12} w_{21}^{(2)}) = \tanh(0.7615 + 0.7615) = 0.9092 \\ u_{22} = \tanh(z_{22}) = \tanh(u_{11} w_{12}^{(2)} + u_{12} w_{22}^{(2)}) = \tanh(0.7615 + 0.7615) = 0.9092 \end{bmatrix}$$

Note that z_{11} is the net input of unit h_{11} and its values is -1 in this case. At the output unit y, we first compute the logit:

$$z_y = u_{21}w_1^{(3)} + u_{22}w_2^{(3)} = 1.8184$$

Hence, the output of the output unit is a the following probability distribution

$$p(y|x_1, x_2) = \sigma((2y - 1)z_y) = \sigma(1.8184(2y - 1))$$

(b) According to page 20, L04, we have

$$\delta_y = \frac{\partial L}{\partial z_y} = -(y - \sigma(z_y)) = \sigma(1.8184) \approx 0.86$$

Through backprop, we get errors for the hidden units:

$$\begin{bmatrix} \delta_{11} = \frac{\partial u_{11}}{\partial z_{11}}[\delta_{21} w_{11}^{(2)} + \delta_{22} w_{12}^{(2)}] = (1 - \tanh(z_{11})^2) \times (-0.3) = 0.42 \times (-0.3) = -0.126 \\ \delta_{12} = \frac{\partial u_{12}}{\partial z_{12}}[\delta_{21} w_{21}^{(2)} + \delta_{22} w_{22}^{(2)}] = (1 - \tanh(z_{12})^2) \times (0.3) = 0.42 \times (0.3) = 0.126 \end{bmatrix} \Leftarrow$$

$$\begin{bmatrix} \delta_{21} = \frac{\partial u_{21}}{\partial z_{21}}[\delta_y w_1^{(3)}] = (1 - \tanh(z_{21})^2) \times 0.86 = 0.17 \times 0.86 = 0.15 \\ \delta_{22} = \frac{\partial u_{22}}{\partial z_{22}}[\delta_y w_2^{(3)}] = (1 - \tanh(z_{22})^2) \times 0.86 = 0.17 \times 0.86 = 0.15 \end{bmatrix} \Leftarrow \quad \delta_y = 0.86$$

Consequently,

$$\frac{\partial L}{\partial w_{22}^{(2)}} = u_{12}\delta_{22} = 0.7615 \times 0.15 = 0.1142$$

$$\frac{\partial L}{\partial w_{22}^{(1)}} = x_2\delta_{12} = 2 \times 0.126 = 0.252.$$

Because the gradients are positive, we should decrease the two parameters if we want to reduce the loss on the example.

Question 4: Why is the sigmoid activation function not recommended for hidden units, but it is fine for an output unit.

Solution: The sigmoid activation function $\sigma(z) = \frac{1}{1+exp(-z)}$ is not recommended for hidden units because it saturates across most of its domain. In other words, its derivative is usually small, preventing errors to back-propagate to units at previous layers.

It is fine for an output unit because of the use of negative log-likelihood (i.e., cross entropy) as the loss function. As such, we back-propagate the gradient of the **logarithm of a sigmoid function**, i.e., $-\log \sigma((2y-1)z)$, instead of the gradient of a sigmoid function itself. The function $-\log \sigma((2y-1)z)$ saturates only when the training example is classified correctly by the model.

Question 5: What is dropout used for in deep learning? Why does it work? Answer briefly.

Solution: Dropout is regularization technique used in deep learning to avoid overfitting. It associates a binary mask variable with some of the units. During training, values for the mask variables are randomly sampled for each minibatch of data, and only parameters for the units with mask variable taking value 1 are oupdated on the minibatch. It reduces overfitting by preventing complex co-adaptation of parameters.

Question 6: What are the key ideas behind the Adam algorithm for training deep neural networks? Answer briefly.

Solution: There are three key ideas: The use of momentum to accelerate learning; Adaption of learning rate to slow down changes on parameters that have changed a lot before; The correction of bias in moment estimates.