EQ2415 – Machine Learning and Data Science
HT22

Tutorial 4 A. Honoré, A. Ghosh

1 Neural networks

1.1 Loss function for regression and classification

Question 1. Suppose that $(\mathbf{t}, \mathbf{x}) \in \mathbb{R}^q \times \mathbb{R}^d$ is a target/input pair for a regression problem, $\mathbf{f}$ is a neural network (NN) model parameterized with a vector $\mathbf{w}$.

Question 1a. Write the likelihood function for the distribution of a target conditional on $\mathbf{x}$ and $\mathbf{w}$. Assume that the noise of the model is Gaussian, that the targets component are independent and that all components share the same noise precision β.
Solution:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{t}|\mathbf{f}(\mathbf{x}, \mathbf{w}), \beta^{-1} I), \tag{1}$$

■

Question 1b. Suppose that you are given a dataset consisting of n independent target/input pairs. Show that maximizing the likelihood of the dataset wrt $\mathbf{w}$ is equivalent to minimizing the MSE wrt $\mathbf{w}$.

Solution: We write the log-likelihood:

$$\begin{aligned} L(\mathbf{w}, \beta) &= \ln \prod_{i=1}^{n} \mathcal{N}(\mathbf{t}_i|\mathbf{f}(\mathbf{x}_i, \mathbf{w}), \beta^{-1} I) \\ &= \sum_{i=1}^{n} \ln \mathcal{N}(\mathbf{t}_i|\mathbf{f}(\mathbf{x}_i, \mathbf{w}), \beta^{-1} I) \\ &= -\frac{1}{2} \sum_{i=1}^{n} (\mathbf{t}_i - \mathbf{f}(\mathbf{x}_i, \mathbf{w}))^T (\beta I)(\mathbf{t}_i - \mathbf{f}(\mathbf{x}_i, \mathbf{w})) + cst \\ &= -\frac{\beta}{2} \sum_{i=1}^{n} ||\mathbf{t}_i - \mathbf{f}(\mathbf{x}_i, \mathbf{w})||^2 \end{aligned} \tag{2}$$

Thus, minimizing the MSE is equivalent to maximizing the likelihood. Also, the noise does not matter for the value of $\mathbf{w}$ that minimizes the MSE. ■

Question 2. (Bishop 5.4) Suppose you are given a binary classification task. You are given a set of n independent training data points $\mathcal{D} = \{(\mathbf{x}_{(i)}, y_{(i)})\}_{i=1}^n$, where $\mathbf{x}_{(i)} \in \mathbb{R}^d$ and $y_{(i)} \in \{0, 1\}$. In general, for a binary classifier, the probability that an input $\mathbf{x}$ is classified with label $y = 1$ is expressed

$$p(y = 1|\mathbf{x}) = y_W(\mathbf{x}), \tag{3}$$

where $y_W : \mathbb{R}^d \to [0, 1]$ is a function of the input $\mathbf{x}$ parameterized with W. Importantly, you are told that the data is mislabeled with probability ϵ. To model this situation, we can introduce a binary random variable modeling the true and unobserved label y_r of an input $\mathbf{x}$. We can also introduce an unobserved binary random variable m associated with each label, indicating whether the label is true or false.

Question 2a. How would you introduce the probability of mislabeling in the output of your classifier ?

Solution: We assume that our classifier models the probability on y_r instead of y. This gives for the output of our model:

$$\begin{aligned} p(y = 1|\mathbf{x}) &= p(y = 1, m = 0|\mathbf{x}) + p(y = 1, m = 1|\mathbf{x}) \\ &= p(m = 0)p(y = 1|m = 0, \mathbf{x}) + p(m = 1)p(y = 1|m = 1, \mathbf{x}). \end{aligned} \tag{4}$$

Now in case there was no mislabeling ($m = 0$) we want the output of our model to use the output of the classifier, i.e. $p(y|m = 0, \mathbf{x}) = p(y_r|\mathbf{x})$ and otherwise ($m = 1$), $p(y|m = 1, \mathbf{x}) = 1 - p(y_r|\mathbf{x})$. Therefore:

$$\begin{aligned} p(y = 1|\mathbf{x}) &= (1 - \epsilon) \cdot p(y_r = 1|\mathbf{x}) + \epsilon \cdot (1 - p(y_r = 1|\mathbf{x})) \\ &= (1 - \epsilon) \cdot \hat{y} + \epsilon \cdot (1 - \hat{y}) \end{aligned} \tag{5}$$

Similarly:

$$\begin{aligned} p(y = 0|\mathbf{x}) &= (1 - \epsilon) \cdot (1 - p(y_r = 1|\mathbf{x})) + \epsilon \cdot p(y_r = 1|\mathbf{x}) \\ &= (1 - \epsilon) \cdot (1 - \hat{y}) + \epsilon \cdot \hat{y} \end{aligned} \tag{6}$$

where we used $\hat{y} = p(y_r = 1|\mathbf{x})$ for short. ■

Question 2b. Write the distribution of the Bernoulli variable $y|\mathbf{x}$.

Solution: Using (5) and (6), the Bernoulli distribution on variable y can then be written:

$$\begin{aligned} p(y|\mathbf{x}) &= [p(y = 1|\mathbf{x})]^y [1 - p(y = 1|\mathbf{x})]^{1-y} \\ &= (1 - \epsilon) \cdot (\hat{y})^y (1 - \hat{y})^{1-y} + \epsilon \cdot (1 - \hat{y})^y (\hat{y})^{1-y}, \end{aligned} \tag{7}$$

here the second line does not follow directly from the first, rather we write the distribution by looking at the factors of $(1 - \epsilon)$ and ϵ that remains for $y = 0$ and $y = 1$. ■

Question 2c. Show that the negative likelihood function on the dataset corresponds to the cross entropy function when $\epsilon = 0$.

Solution: Since the data samples are independent, the likelihood is the product of the marginal likelihood of each sample:

$$\begin{aligned} E &= -\ln \prod_{i=1}^{n} p(y_{(i)}|\mathbf{x}_{(i)}) \\ &= -\sum_{i=1}^{n} \ln[(1 - \epsilon) \cdot (\hat{y}_{(i)})^{y_{(i)}} (1 - \hat{y}_{(i)})^{1-y_{(i)}} + \epsilon \cdot (1 - \hat{y}_{(i)})^{y_{(i)}} (\hat{y}_{(i)})^{1-y_{(i)}}] \end{aligned} \tag{8}$$

When $\epsilon = 0$ we find the cross-entropy function:

$$E = -\sum_{i=1}^{n} y_{(i)} \ln[\hat{y}_{(i)}] + (1 - y_{(i)}) \ln[1 - \hat{y}_{(i)}] \tag{9}$$

■

1.2 Standard results on activation functions

Let us consider a real valued functions $h : \mathbb{R} \to \mathbb{R}$. We calculate the derivative of h wrt to its argument $x \in \mathbb{R}$ for different values of h.

Question 1.

$$h(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

Solution:

$$\begin{aligned}
h'(x) &= -\frac{-e^{-x}}{(1+e^{-x})^2} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} \\
&= h(x)\frac{1+e^{-x}-1}{1+e^{-x}} \\
&= h(x)(1-\frac{1}{1+e^{-x}}) \\
&= h(x)(1-h(x))
\end{aligned} \tag{11}$$

■

Question 2.

$$h(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{12}$$

Solution:

$$\begin{aligned}
h'(x) &= \frac{(e^x+e^{-x})(e^x+e^{-x}) - (e^x-e^{-x})(e^x-e^{-x})}{(e^x+e^{-x})^2} \\
h'(x) &= 1 - \left(\frac{e^x-e^{-x}}{e^x+e^{-x}}\right)^2 \\
h'(x) &= 1 - h(x)^2
\end{aligned} \tag{13}$$

■

Question 3.

$$h(x) = \max(0, x) \tag{14}$$

Solution:

$$h'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

■

1.3 Multi-layer perceptron

Suppose you are given n data points for a regression or classification task in the form of two matrices: $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times q}$. Note that this time the data and target vectors are row vectors. This is the standard notation in ML.

In what follows, we will derive the back-propagation update rules for an artificial neural network composed of 1 hidden layer with m hidden neurons and a component wise sigmoid activation function σ. We denote the input weight matrix by $W^{(1)} \in \mathbb{R}^{d \times m}$, and the output weight matrix by $W^{(2)} \in \mathbb{R}^{m \times q}$.

Question 0. Draw the network. Specify the meaning of the edge and nodes in terms of the parameters, inputs and outputs of the network.

Question 1. Express the output of the network $\hat{Y} \in \mathbb{R}^{n \times q}$ in terms of the network parameters and activation function.
Solution:

$$\hat{Y} = \sigma(XW^{(1)})W^{(2)} \tag{16}$$

Let us denote $A = \sigma(XW^{(1)})$. ■

We train the NN to minimize the MSE loss wrt both $W^{(1)}$ and $W^{(2)}$. The loss can be written:

$$\begin{aligned} E(\hat{Y}) &= \frac{1}{2n}||\hat{Y} - Y||_F^2 \\ &= \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{q}\frac{1}{2}(\hat{y}_{k,j} - y_{k,j})^2. \end{aligned} \tag{17}$$

Question 2. (Back-propagation update rules.)

Question 2a. Calculate the Jacobian matrix of $E(\hat{Y})$ wrt $W^{(2)} \in \mathbb{R}^{m\times q}$:

$$\frac{\partial E(\hat{Y})}{\partial W^{(2)}}. \tag{18}$$

Hint: You can first derive the value of every index, and then find the Jacobian in matrix form.

Solution:

$$\begin{aligned} \frac{\partial E(\hat{Y})}{\partial w_{i,j}^{(2)}} &= \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{q}\frac{1}{2}\frac{\partial(\hat{y}_{k,j} - y_{k,j})^2}{\partial w_{i,j}^{(2)}} \\ &= \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{q}\frac{1}{2}\frac{\partial(\hat{y}_{k,j} - y_{k,j})^2}{\partial \hat{y}_{k,j}}\frac{\partial \hat{y}_{k,j}}{w_{i,j}^{(2)}} \end{aligned} \tag{19}$$

We have

$$\hat{y}_{k,j} = \sum_{l=1}^{m} a_{k,l} w_{l,j}^{(2)}. \tag{20}$$

Thus

$$\frac{\partial E(\hat{Y})}{\partial w_{i,j}^{(2)}} = \frac{1}{n}\sum_{k=1}^{n}(\hat{y}_{k,j} - y_{k,j})a_{k,i} \tag{21}$$

And in matrix form:

$$\frac{\partial E(\hat{Y})}{\partial W^{(2)}} = \frac{1}{n}A^T(\hat{Y} - Y) \tag{22}$$

■

Question 2b. What is the derivative of the composition of scalar functions: $l \circ f \circ h \circ g$?

Solution:

$$(l \circ f \circ h \circ g)'(x) = g'(x) \cdot h' \circ g(x) \cdot f' \circ h \circ g(x) \cdot l' \circ f \circ h \circ g(x) \tag{23}$$

■

Question 2c. Calculate the Jacobian matrix of $E(\hat{Y})$ wrt $W^{(1)} \in \mathbb{R}^{d\times m}$:

$$\frac{\partial E(\hat{Y})}{\partial W^{(1)}}. \tag{24}$$

Solution:

$$\begin{aligned} \frac{\partial E(\hat{Y})}{\partial W^{(1)}} &= \frac{1}{n}\frac{\partial[XW^{(1)}]}{\partial W^{(1)}} \cdot \left[\left.\frac{\partial \sigma(x)}{\partial x}\right|_{XW^{(1)}} \odot \left(\frac{\partial E}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial A}\right)\right] \\ &= \frac{1}{n}X^T\left[\sigma(XW^{(1)})(1 - \sigma(XW^{(1)}) \odot \left((\hat{Y} - Y){W^{(2)}}^T\right)\right], \end{aligned} \tag{25}$$

where $\odot$ denotes the Hadamart product of matrices (element wise product). ■

Question 2d. Write the update rule for $W^{(1)}$ and $W^{(2)}$, assuming that the network is trained using batch gradient descent and with learning rate $\eta > 0$. Suppose that we are computing the update rule for step $k+1$, i.e. we can denote $\hat{Y}_k$, the output of the network when the weights are $W_k^{(1)}$ and $W_k^{(2)}$.

Solution:

$$\begin{aligned} W_{k+1}^{(1)} &= W_k^{(1)} - \eta \left. \frac{\partial E(\hat{Y})}{\partial W^{(1)}} \right|_{\hat{Y}_k, W_k^{(1)}, W_k^{(2)}} \\ W_{k+1}^{(2)} &= W_k^{(2)} - \eta \left. \frac{\partial E(\hat{Y})}{\partial W^{(2)}} \right|_{\hat{Y}_k, W_k^{(1)}, W_k^{(2)}} \end{aligned} \tag{26}$$

■

Question 3. Implement backprop for this 1 hidden layer neural network example !