Statistical Machine Learning
Solutions for Exam 2019-08-21

1. i. **True**
   ii. **False** It is a classification method (despite its name)
   iii. **True**
   iv. **True**
   v. **False**
   vi. **True**
   vii. **False**
   viii. **True**
   ix. **True**
   x. **False**

2. (a) The convolutional layer maps 3×3 pixels to 16 channels, meaning that $\mathbf{W}^{(1)} \in \mathbb{R}^{3\times3\times1\times16}$ and $\mathbf{b}^{(1)} \in \mathbb{R}^{16}$. Because of the stride $[2, 2]$, the hidden layer $\mathbf{H}$ will have dimension $20 \times 20 \times 16$. The number of elements will therefore be:
$\mathbf{W}^{(1)} : 3 \times 3 \times 16 = 144$
$\mathbf{b}^{(1)} : 16$
$\mathbf{H} : 20 \times 20 \times 16 = 6400$.

   (b) The logits have dimension 4 (since there are 4 class probabilities to be predicted). The dense layer maps a vectorized version of $\mathbf{H}$ (dimension $20 \times 20 \times 16$) onto 4 logits $\mathbf{z}$, meaning $\mathbf{W}^{(2)} \in \mathbb{R}^{6400\times4}$ and $\mathbf{b}^{(2)} \in \mathbb{R}^4$. The number of elements will therefore be:
$\mathbf{W}^{(2)} : 6400 \times 4 = 25600$
$\mathbf{b}^{(2)} : 4$
$\mathbf{z} : 4$

   (c) The total number of parameters is the total number of elements in the two weight matrices and the two offset vectors. Consequently, in total there are $144 + 16 + 25600 + 4 = 25764$ parameters.

   (d) In a dense layer each input variable is connected to each hidden unit in the following layer, and each connection has a unique parameter associated with it. In a convolutional layer, however, a hidden unit only depends on a small subset of input variables, which corresponds to forcing most of the parameters in a dense layer to be equal to zero ("sparse interactions"). Moreover, in a convolutional layer the same set of parameters (a so-called kernel) is used for different hidden units ("parameter sharing").

   (e) In gradient descent we use all training data at each iteration to compute the gradient whereas in mini-batch gradient descent we only use a small randomly selected subset of the training data to compute the gradient.

   (f) An epoch is a set of iterations where all the training data has been used once.

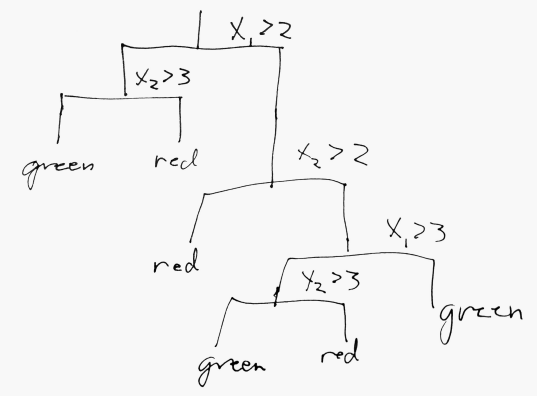
   (g) The main advantage of mini-batch gradient descent in comparison to gradient descent is that it is computationally cheaper to approximate the gradient based on a small subset of the training data, compared to compute it exactly using all training data.

3. (a) See lecture notes.

   (b) See lecture notes.

   (c) See lecture notes.

   (d)



4. (a) The misclassification loss is given by $I(y \neq \widehat{y}(x))$ and the exponential loss is given by $\exp(-yC(x))$ where $C(x) = x - 2.625$. Hence we have:

x_i	1.75	2	2.5	2.75	3
y_i	1	-1	-1	1	1
$C(x_i)$	-0.875	-0.625	-0.125	0.125	0.375
$\widehat{y}(x_i)$	-1	-1	-1	1	1
$y_i C(x_i)$	-0.875	0.625	0.125	0.125	0.375
$I(y_i \neq \widehat{y}(x_i))$	1	0	0	0	0
$\exp(-y_i C(x_i))$	2.40	0.54	0.88	0.88	0.69

The misclassification rate (average misclassification loss) is thus $\frac{1+0+0+0+0}{5} = 0.2$ and the average exponential loss is $\frac{2.40+0.54+0.88+0.88+0.69}{5} = 1.08$

(b) To achieve zero misclassification, the classifier needs to flip sign at least two times;

- from 1 to -1 between 1.75 and 2
- from -1 to 1 between 2.5 and 2.75.

For a LDA classifier the decision boundary will be linear on the form $bx + c = 0$ which is only fulfilled at maximum one point on the real line. Hence, it is not possible to achive zero misclassification for a LDA classifier.

For a QDA classifier the decision boundary will be quadratic on the form $ax^2+bx+c = 0$ which is fulfilled at maximum two points on the real line. Hence, a QDA classifier could obtain zero misclasification on this data.

(c) For the LDA classifier, we obtain $\widehat{\pi}_1 = 0.6$, $\widehat{\pi}_{-1} = 0.4$, $\widehat{\mu}_1 = 2.5$, $\widehat{\mu}_{-1} = 2.25$ and $\widehat{\sigma}^2 = 0.33$.

The decision boundary is defined by x such that $p(y = 1 \mid x) = p(y = -1 \mid x)$,

$$
\begin{aligned}
p(y = 1 \mid x) &= p(y = -1 \mid x) \Leftrightarrow \\
\frac{p(x \mid y = 1)p(y = 1)}{p(x)} &= \frac{p(x \mid y = -1)p(y = -1)}{p(x)} \Leftrightarrow \\
p(x \mid y = 1)p(y = 1) &= p(x \mid y = -1)p(y = -1) \Leftrightarrow \\
\log p(x \mid y = 1) + \log p(y = 1) &= \log p(x \mid y = -1) + \log p(y = -1) \Leftrightarrow \\
\log \mathcal{N}(x; \widehat{\mu}_1, \widehat{\sigma}^2) + \log \pi_1 &= \log \mathcal{N}(x; \widehat{\mu}_{-1}, \widehat{\sigma}^2) + \log \pi_{-1} \Leftrightarrow \\
-\frac{1}{2}\log 2\pi\widehat{\sigma}^2 - \frac{1}{2\widehat{\sigma}^2}(x - \widehat{\mu}_1)^2 + \log \pi_1 &= -\frac{1}{2}\log 2\pi\widehat{\sigma}^2 - \frac{1}{2\widehat{\sigma}^2}(x - \widehat{\mu}_{-1})^2 + \log \pi_{-1} \Leftrightarrow \\
(x - \widehat{\mu}_{-1})^2 - (x - \widehat{\mu}_1)^2 &= 2\widehat{\sigma}^2(\log \pi_{-1} - \log \pi_1) \Leftrightarrow \\
2x(\widehat{\mu}_1 - \widehat{\mu}_{-1}) - (\widehat{\mu}_1^2 - \widehat{\mu}_{-1}^2) &= 2\widehat{\sigma}^2(\log \pi_{-1} - \log \pi_1) \Leftrightarrow \\
x &= \frac{2\widehat{\sigma}^2(\log \pi_{-1} - \log \pi_1) + (\widehat{\mu}_1^2 - \widehat{\mu}_{-1}^2)}{2(\widehat{\mu}_1 - \widehat{\mu}_{-1})} = 1.83
\end{aligned}
$$

That is,

$$
\widehat{y}(x) = \begin{cases} -1 & \text{if } x \leq 1.83 \\ 1 & \text{otherwise} \end{cases},
$$

which gives a misclassification rate $3/5$ for the training data.

(d) For the QDA classifier, we obtain $\widehat{\pi}_1 = 0.6$, $\widehat{\pi}_{-1} = 0.4$, $\widehat{\mu}_1 = 2.5$, $\widehat{\mu}_{-1} = 2.25$, $\widehat{\sigma}_1^2 = 0.44$ and $\widehat{\sigma}_{-1}^2 = 0.13$. Similarly to LDA, we find the decision boundary as

$$p(y = 1 \mid x) = p(y = -1 \mid x) \Leftrightarrow$$
$$-\frac{1}{2}\log\widehat{\sigma}_1^2 - \frac{1}{2\widehat{\sigma}_1^2}(x - \widehat{\mu}_1)^2 + \log\pi_1 = -\frac{1}{2}\log\widehat{\sigma}_{-1}^2 - \frac{1}{2\widehat{\sigma}_{-1}^2}(x - \widehat{\mu}_{-1})^2 + \log\pi_{-1},$$

which is a quadratic equation with solutions $x = 1.81$ and $x = 2.48$, and hence

$$\widehat{y}(x) = \begin{cases} -1 & \text{if } 1.81 \leq x \leq 2.48 \\ 1 & \text{otherwise} \end{cases},$$

which gives a misclassification rate $1/5$ for the training data.

5. (a) We assume that the data points are independent, meaning that we can write

$$\log p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta}) = \log \prod_{i=1}^{n} p(y_i \mid \mathbf{x}_i; \boldsymbol{\beta}) = \sum_{i=1}^{n} \log p(y_i \mid \mathbf{x}_i; \boldsymbol{\beta}).$$

With the logistic regression model, we have that

$$p(y_i = 1 \mid \mathbf{x}_i; \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}} = \frac{e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}},$$

and consequently

$$\begin{aligned} p(y_i = -1 \mid \mathbf{x}_i; \boldsymbol{\beta}) = 1 - p(y_i = 1 \mid \mathbf{x}_i; \boldsymbol{\beta}) &= 1 - \frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}} \\ &= \frac{1}{1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}} = \frac{e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}} = \frac{e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}, \end{aligned}$$

Combining this gives

$$\log p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\beta}) = \sum_{i=1}^{n} \log \frac{e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}}{1 + e^{y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i}} = \sum_{i=1}^{n} -\log(1 + \exp(-y_i\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i)).$$

(b) One option is to use the dummy variables x_1, x_2 and x_3 as

$$x_1 = \begin{cases} 1 & \text{if } \texttt{green} \\ 0 & \text{else} \end{cases}, \quad x_2 = \begin{cases} 1 & \text{if } \texttt{blue} \\ 0 & \text{else} \end{cases}, \quad x_3 = \begin{cases} 1 & \text{if } \texttt{pink} \\ 0 & \text{else} \end{cases}.$$