CSC411H1 L0101, Fall 2018
Assignment 4

Name: JingXi, Hao
Student Number: 1000654188

Due Date: 02 November, 2018 11:59pm

1. **AlexNext.** For this question, you will first read the following paper:

   A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classi

   cation with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), 2012.

   http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

   This is a highly influential paper (over 45,000 citations on Google Scholar!) because it was one of the first papers to demonstrate impressive performance for a neural network on a modern computer vision benchmark. It generated lots of excitement both in academia and in the tech industry. The architecture presented in this paper widely used today, and is known as 'AlexNet', after the first author. Reading this paper will also help you review a lot of the important concepts from this class.

   (a) They use a conv net architecture which has five convolution layers and three fully connected layers (one of which is the output layer). Your job is to count the number of units, the number of weights, and the number of connections in each layer. i.e., you should complete the following table:

	# Units # Weights # Connections
Convolution Layer 1	
Convolution Layer 2	
Convolution Layer 3	
Convolution Layer 4	
Convolution Layer 5	
Fully Connected Layer 1	
Fully Connected Layer 2	
Output Layer	

   You can ignore the pooling layers when doing these calculations, i.e. you don't need to consider the units in the pooling layers or the connections between convolution and pooling layers. You can also ignore the biases. Note that the paper gives you the answers for the numbers of units in the caption to Figure 2. Therefore, we won't mark the column for units, though you would benefit from trying to work it out yourself.

   When counting the number of connections, we'll adopt the convention that when the input to a convolution layer is zero-padded, the connections to the dummy zero values count towards

the total. (This is the most convenient way to do it, since it means the number of incoming connections is the same for each unit in a given layer.)

Solution:

Based on the graph, showing the architecture of AlexNet, displayed below, we compute and fill out the table.

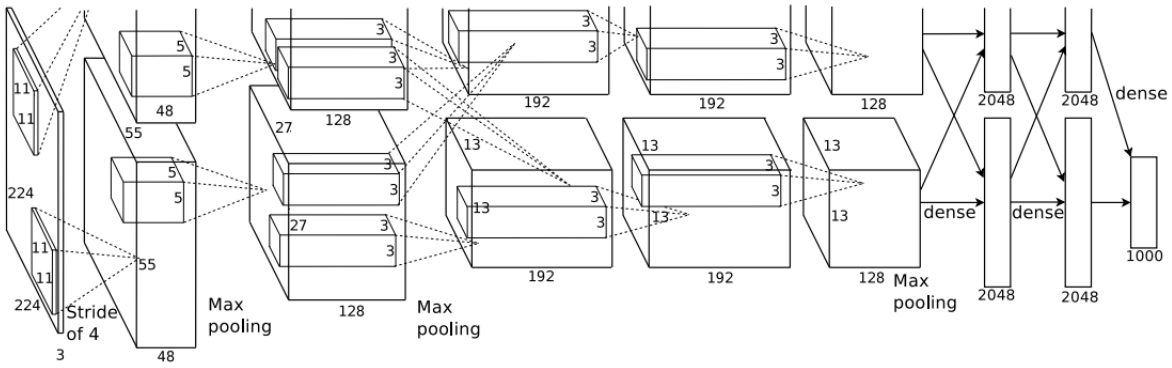

Figure 1: Architecture For AlexNet

Then, we count the number of units, the number of weights, and the number of connections in each layer and complete the following table.

	# Units	# Weights	# Connections
Convolution Layer 1	(55 * 55 * 48) * 2 = 290400	(11 * 11 * 3 * 48) * 2 = 34848	(11 * 11 * 3 * 55 * 55 * 48) * 2 = 105415200
Convolution Layer 2	(27 * 27 * 128) * 2 = 186624	(5 * 5 * 48 * 128) * 2 = 307200	(5 * 5 * 48 * 27 * 27 * 128) * 2 = 223948800
Convolution Layer 3	(13 * 13 * 192) * 2 = 64896	(3 * 3 * 128 * 2 * 192) * 2 = 884736	(3 * 3 * 128 * 2 * 13 * 13 * 192) * 2 = 149520384
Convolution Layer 4	(13 * 13 * 192) * 2 = 64896	(3 * 3 * 192 * 192) * 2 = 663552	(3 * 3 * 192 * 13 * 13 * 192) * 2 = 112140288
Convolution Layer 5	(13 * 13 * 128) * 2 = 43264	(3 * 3 * 192 * 128) * 2 = 442368	(3 * 3 * 192 * 13 * 13 * 128) * 2 = 74760192
Fully Connected Layer 1	2048 * 2 = 4096	(6 * 6 * 128 * 2 * 2048) * 2 = 37748736	(6 * 6 * 128 * 2 * 2048) * 2 = 37748736
Fully Connected Layer 2	2048 * 2 = 4096	(2048 * 2 * 2048) * 2 = 16777216	(2048 * 2 * 2048) * 2 = 16777216
Output Layer	1000	2048 * 2 * 1000 = 4096000	2048 * 2 * 1000 = 4096000

Note that the number '6' shown in the calculation of the number of weights of fully connected layer

1 is the width and height of input size after applying max pooling, where $w' = [(w - z)/s] + 1 = [(13 - 3)/2] + 1 = 6$ ($z = 3$ and $s = 2$ are stated in the paper) and do the same for computing the height.

(b) Now suppose you're working at a software company and want to use an architecture similar to AlexNet in a product. Your project manager gives you some additional instructions; for each of the following scenarios, based on your answers to Part 1, suggest a change to the architecture which will help achieve the desired objective. i.e., modify the sizes of one or more layers. (These scenarios are independent.)

i. You want to reduce the memory usage at test time so that the network can be run on a cell phone; this requires reducing the number of parameters for the network.

Solution: In order to reduce the memory usage at test time, the decrease in the number of parameters for the network is required. This means that we ought to reduce the number of weights in each layers, especially in the fully connected layers, since those layers contain much more number of weights in comparison with convolution layers based on the table shown in (a). Therefore, for reducing the amount of parameters, we can add extra pooling layers, i.e. max pooling layers, between convolution layers or we also can use larger value of stride instead of 4.

ii. Your network will need to make very rapid predictions at test time. You want to reduce the number of connections, since there is approximately one add-multiply operation per connection.

Solution:
Based on the table shown in (a), we are able to see that the convolution layers have more connections than other layers do. Thus, according to the way of computing the number of connections for convolution layers, we ought to reduce the amount of filters or increase the value of stride to decrease the input size in order to highly reduce the number of connections, which consequently are able to let the network make rapid predictions at test time.

2. **Gaussian Naive Bayes.** In this question, you will derive the maximum likelihood estimates for Gaussian Naive Bayes, which is just like the naive Bayes model from lecture, except that the features are continuous, and the conditional distribution of each feature given the class is (univariate) Gaussian rather than Bernoulli. Start with the following generative model for a discrete class label $y \in (1, 2, \ldots, k)$ and a real valued vector of d features $\mathbf{x} = (x_1, x_2, \ldots, x_d)$

$$p(y = k) = \alpha_k \tag{1}$$

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) = (\prod_{i=1}^{D} 2\pi {\sigma_i}^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2{\sigma_i}^2}(x_i - \mu_{ki})^2\} \tag{2}$$

where α_k is the prior on class k, ${\sigma_i}^2$ are the variances for each feature, which are shared between all classes, and μ_{ki} is the mean of the feature i conditioned on class k. We write $\boldsymbol{\alpha}$ to represent the vector with elements α_k and similarly $\boldsymbol{\sigma}$ is the vector of variances. The matrix of class means is written $\boldsymbol{\mu}$ where the k^{th} row of $\boldsymbol{\mu}$ is the mean for class k.

(a) Use Bayes' rule to derive an expression for $p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. Hint: Use the law of total probability to derive an expression for $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})$.

Solution:

For this question, we are going to apply Bayes' rule to derive an expression for $p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. Therefore, we have that

$$
\begin{aligned}
p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \frac{p(\mathbf{x}|y=k,\boldsymbol{\mu},\boldsymbol{\sigma})p(y=k|\boldsymbol{\mu},\boldsymbol{\sigma})}{p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\sigma})} && \text{\# By Bayes' Rule} \\
&= \frac{p(\mathbf{x}|y=k,\boldsymbol{\mu},\boldsymbol{\sigma})p(y=k|\boldsymbol{\mu},\boldsymbol{\sigma})}{\sum_{k'} p(y=k'|\boldsymbol{\mu},\boldsymbol{\sigma})p(\mathbf{x}|y=k',\boldsymbol{\mu},\boldsymbol{\sigma})} && \text{\# By Law of Total Probability} \\
&= \frac{(\prod_{i=1}^{D} 2\pi{\sigma_i}^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2{\sigma_i}^2}(x_i-\mu_{ki})^2\}\alpha_k}{\sum_{k'}(\prod_{i=1}^{D} 2\pi{\sigma_i}^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2{\sigma_i}^2}(x_i-\mu_{k'i})^2\}\alpha_{k'}} && \text{\# By Substitution of Eq. (1) \& (2)}
\end{aligned}
$$

Hence, we have derived an expression for $p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma})$, where

$$p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{(\prod_{i=1}^{D} 2\pi{\sigma_i}^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2{\sigma_i}^2}(x_i-\mu_{ki})^2\}\alpha_k}{\sum_{k'}(\prod_{i=1}^{D} 2\pi{\sigma_i}^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2{\sigma_i}^2}(x_i-\mu_{k'i})^2\}\alpha_{k'}}$$ by applying Bayes' rule.

(b) Write down an expression for the negative likelihood function (NLL).

$$\ell(\boldsymbol{\theta}; D) = -logp(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, \ldots, y^{(N)}, \mathbf{x}^{(N)}|\boldsymbol{\theta}) \tag{3}$$

of a particular dataset $D = \{(y^{(1)}, \mathbf{x}^{(1)}), \ldots, (y^{(N)}, \mathbf{x}^{(N)})\}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$. (Assume the data are i.i.d.)

Solution:

For this question, we are going to find an expression for the negative likelihood function. Thus, we have that

$$
\begin{aligned}
\ell(\boldsymbol{\theta}; D) &= -logp(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, \ldots, y^{(N)}, \mathbf{x}^{(N)}|\boldsymbol{\theta}) \\
&= -log(\textstyle\prod_{i=1}^{N} p(y^{(i)}, \mathbf{x}^{(i)}|\boldsymbol{\theta})) && \text{\# By Data i.i.d} \\
&= \textstyle\sum_{i=1}^{N}(-log(p(y^{(i)}, \mathbf{x}^{(i)}|\boldsymbol{\theta}))) \\
&= -\textstyle\sum_{i=1}^{N}(log(p(y^{(i)}, \mathbf{x}^{(i)}|\boldsymbol{\theta}))) \\
&= -\textstyle\sum_{i=1}^{N} log(p(\mathbf{x}^{(i)}|y^{(i)}, \boldsymbol{\theta})p(y^{(i)}|\boldsymbol{\theta})) && \text{\# By } p(X,Y) = p(X|Y)p(Y) = p(Y|X)p(X) \\
&= -\textstyle\sum_{i=1}^{N} log(p(\mathbf{x}^{(i)}|y^{(i)}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma})p(y^{(i)}|\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma})) && \text{\# By } \boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}\} \\
&= -\textstyle\sum_{i=1}^{N} log((\prod_{j=1}^{D} 2\pi{\sigma_j}^2)^{-1/2} exp\{-\sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j - \mu_{k_i j})^2\}\alpha_{k_i}) && \text{\# Let } p(y^{(i)} = k_i) = \alpha_{k_i} \\
&= -\textstyle\sum_{i=1}^{N}[-\frac{1}{2}\sum_{j=1}^{D} log(2\pi{\sigma_j}^2) + (-\sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j - \mu_{k_i j})^2) + log(\alpha_{k_i})] \\
&= -\textstyle\sum_{i=1}^{N}[-\frac{1}{2}\sum_{j=1}^{D} log(2\pi{\sigma_j}^2) - \sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j - \mu_{k_i j})^2 + log(\alpha_{k_i})] \\
&= \textstyle\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D} log(2\pi{\sigma_j}^2) + \sum_{i=1}^{N}\sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j^{(i)} - \mu_{k_i j})^2 - \sum_{i=1}^{N} log(\alpha_{k_i}) \\
&= \textstyle\frac{N}{2}\sum_{j=1}^{D} log(2\pi{\sigma_j}^2) + \sum_{i=1}^{N}\sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j^{(i)} - \mu_{k_i j})^2 - \sum_{i=1}^{N} log(\alpha_{k_i})
\end{aligned}
$$

Therefore, we have found an expression for the negative likelihood function, which is $\ell(\boldsymbol{\theta}; D) = \frac{N}{2}\sum_{j=1}^{D} log(2\pi{\sigma_j}^2) + \sum_{i=1}^{N}\sum_{j=1}^{D} \frac{1}{2{\sigma_j}^2}(x_j^{(i)} - \mu_{k_i j})^2 - \sum_{i=1}^{N} log(\alpha_{k_i})$.

(c) Take partial derivatives of the likelihood with respect to each of the parameters μ_{ki} and with respect to the shared variances ${\sigma_i}^2$. Based on this, find the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. You may assume that each class appears at least once in the dataset.

Solution:

- First, we take the partial derivative of the log likelihood with respect to μ_{ki}. Hence, we have that

$$\begin{aligned}
\frac{\partial \ell}{\partial \mu_{ki}} &= \frac{\partial(\frac{N}{2}\sum_{i=1}^{D} log(2\pi{\sigma_i}^2)+\sum_{j=1}^{N}\sum_{i=1}^{D}\frac{1}{2{\sigma_i}^2}(x_i^{(j)}-\mu_{k_j i})^2-\sum_{j=1}^{N} log(\alpha_{k_j}))}{\partial \mu_{ki}} \quad \text{\# Where } p(y^{(j)}=k_j)=\alpha_{k_j} \\
&= \sum_{j=1}^{N}\frac{2}{2{\sigma_i}^2}(x_i^{(j)}-\mu_{ki})(-1)\mathbb{1}[y^{(j)}=k] \\
&= \sum_{j=1}^{N}\frac{-1}{{\sigma_i}^2}(x_i^{(j)}-\mu_{ki})\mathbb{1}[y^{(j)}=k] \\
&= -\frac{1}{{\sigma_i}^2}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{ki})\mathbb{1}[y^{(j)}=k]
\end{aligned}$$

  Based on the partial derivative equation, we compute the maximum likelihood estimation for $\boldsymbol{\mu}$. Then, let $\frac{\partial \ell}{\partial \mu_{ki}}=0$ and we have that

$$\begin{aligned}
&\frac{\partial \ell}{\partial \mu_{ki}} = 0 \\
&-\frac{1}{{\sigma_i}^2}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{ki})\mathbb{1}[y^{(j)}=k]=0 \\
&\sum_{j=1}^{N}(x_i^{(j)}-\mu_{ki})\mathbb{1}[y^{(j)}=k]=0 \\
&\sum_{j=1}^{N}(x_i^{(j)}\mathbb{1}[y^{(j)}=k]-\mu_{ki}\mathbb{1}[y^{(j)}=k])=0 \\
&\sum_{j=1}^{N}x_i^{(j)}\mathbb{1}[y^{(j)}=k]-\sum_{j=1}^{N}\mu_{ki}\mathbb{1}[y^{(j)}=k]=0 \\
&\sum_{j=1}^{N}x_i^{(j)}\mathbb{1}[y^{(j)}=k]=\sum_{j=1}^{N}\mu_{ki}\mathbb{1}[y^{(j)}=k] \\
&\sum_{j=1}^{N}x_i^{(j)}\mathbb{1}[y^{(j)}=k]=\mu_{ki}\sum_{j=1}^{N}\mathbb{1}[y^{(j)}=k] \\
&\mu_{ki}=\frac{\sum_{j=1}^{N}x_i^{(j)}\mathbb{1}[y^{(j)}=k]}{\sum_{j=1}^{N}\mathbb{1}[y^{(j)}=k]}
\end{aligned}$$

  Hence, we have that $\widehat{\mu_{ki}}=\frac{\sum_{j=1}^{N}x_i^{(j)}\mathbb{1}[y^{(j)}=k]}{\sum_{j=1}^{N}\mathbb{1}[y^{(j)}=k]}$ by using MLE. Thus, we are able to estimate every entry in $\boldsymbol{\mu}$.

- Then, we take the partial derivative of the log likelihood with respect to ${\sigma_i}^2$. Therefore, we have that

$$\begin{aligned}
\frac{\partial \ell}{\partial {\sigma_i}^2} &= \frac{\partial(\frac{N}{2}\sum_{i=1}^{D} log(2\pi{\sigma_i}^2)+\sum_{j=1}^{N}\sum_{i=1}^{D}\frac{1}{2{\sigma_i}^2}(x_i^{(j)}-\mu_{k_j i})^2-\sum_{j=1}^{N} log(\alpha_{k_j}))}{\partial {\sigma_i}^2} \quad \text{\# Where } p(y^{(j)}=k_j)=\alpha_{k_j} \\
&= \frac{N}{2}\frac{2\pi}{2\pi{\sigma_i}^2}+\sum_{j=1}^{N}(-1)(2{\sigma_i}^2)^{-2}(2)(x_i^{(j)}-\mu_{k_j i})^2 \\
&= \frac{N}{2}\frac{2\pi}{2\pi{\sigma_i}^2}+\sum_{j=1}^{N}(-1)\frac{2}{(2{\sigma_i}^2)^2}(x_i^{(j)}-\mu_{k_j i})^2 \\
&= \frac{N}{2{\sigma_i}^2}-\sum_{j=1}^{N}\frac{1}{2{\sigma_i}^4}(x_i^{(j)}-\mu_{k_j i})^2 \\
&= \frac{N}{2{\sigma_i}^2}-\frac{1}{2{\sigma_i}^4}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2
\end{aligned}$$

  Based on the partial derivative equation, we compute the maximum likelihood estimation for $\boldsymbol{\sigma}$. Then, let $\frac{\partial \ell}{\partial {\sigma_i}^2}=0$ and we have that

$$\begin{aligned}
&\frac{\partial \ell}{\partial {\sigma_i}^2} = 0 \\
&\frac{N}{2{\sigma_i}^2}-\frac{1}{2{\sigma_i}^4}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2=0 \\
&\frac{1}{2{\sigma_i}^4}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2=\frac{N}{2{\sigma_i}^2} \\
&\frac{1}{{\sigma_i}^2}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2=N \\
&{\sigma_i}^2=\frac{\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2}{N} \\
&{\sigma_i}^2=\frac{1}{N}\sum_{j=1}^{N}(x_i^{(j)}-\mu_{k_j i})^2
\end{aligned}$$

Hence, we have that $\widehat{\sigma_i}^2 = \frac{1}{N}\sum_{j=1}^{N}(x_i^{(j)} - \mu_{k_j i})^2$ by using MLE. Thus, we are able to estimate every entry in $\boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is a vector of ${\sigma_i}^2$.

(d) Show that the MLE for α_k is given by the following equation:

$$\alpha_k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = k] \tag{4}$$

You may assume that each class appears at least once. You will find it helpful to read about Lagrange multipliers.

Solution:

Let $\mathcal{L} = f + \lambda g$, where $f = \ell(\boldsymbol{\theta}; D)$ and $g = 1 - \sum_k \alpha_k$ since we want to maximize the $\ell(\boldsymbol{\theta}; D)$ under the constraint $\sum_k \alpha_k = 1$. We have that the partial derivative of $\mathcal{L}$ over α_k is $\frac{\partial \mathcal{L}}{\partial \alpha_k} = \frac{\partial f}{\partial \alpha_k} - \lambda \frac{\partial g}{\partial \alpha_k}$. Then, there exists a Lagrange multiplier λ such that $\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0$. Therefore, we have that

$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0$

$\frac{\partial f}{\partial \alpha_k} - \lambda \frac{\partial g}{\partial \alpha_k} = 0$

$\frac{\partial f}{\partial \alpha_k} = \lambda \frac{\partial g}{\partial \alpha_k}$

$\frac{\partial \ell}{\partial \alpha_k} = \lambda \frac{\partial g}{\partial \alpha_k}$

$\frac{\partial(\frac{N}{2}\sum_{j=1}^{D} log(2\pi {\sigma_j}^2) + \sum_{i=1}^{N}\sum_{j=1}^{D}\frac{1}{2{\sigma_j}^2}(x_j^{(i)} - \mu_{k_i j})^2 - \sum_{i=1}^{N} log(\alpha_{k_i}))}{\partial \alpha_k} = \lambda \frac{\partial g}{\partial \alpha_k}$ # Where $p(y^{(i)} = k_i) = \alpha_{k_i}$

$-\sum_{i=1}^{N}\frac{1}{\alpha_k}\mathbb{1}[y^{(i)} = k] = \lambda \frac{\partial(1 - \sum_{k'}\alpha_{k'})}{\partial \alpha_k}$ # Write k' in the numerator to distinguish between k in the denominator

$-\sum_{i=1}^{N}\frac{1}{\alpha_k}\mathbb{1}[y^{(i)} = k] = (-1)\lambda$

$\sum_{i=1}^{N}\frac{1}{\alpha_k}\mathbb{1}[y^{(i)} = k] = \lambda$

$\lambda = \sum_{i=1}^{N}\frac{1}{\alpha_k}\mathbb{1}[y^{(i)} = k]$

$\alpha_k = \frac{1}{\lambda}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$

Therefore, we have that $\alpha_k = \frac{1}{\lambda}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$. Then, we find that the partial derivative of $\mathcal{L}$ over λ is $\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{\partial f}{\partial \lambda} - g$. Then, there exists a Lagrange multiplier λ such that $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$. Therefore, we have that

$\frac{\partial \mathcal{L}}{\partial \lambda} = 0$

$\frac{\partial f}{\partial \lambda} - g = 0$

$g = 0$

$1 - \sum_k \alpha_k = 0$

$1 - \sum_k \frac{1}{\lambda}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k] = 0$

$1 - \frac{1}{\lambda}\sum_{i=1}^{N}\sum_k \mathbb{1}[y^{(i)} = k] = 0$

$1 - \frac{1}{\lambda}\sum_{i=1}^{N} 1 = 0$ # $\sum_k \mathbb{1}[y^{(i)} = k] = 1$

$1 - \frac{1}{\lambda}N = 0$

$\lambda = N$

Hence, we substitute $\lambda = N$ back into the equation of α_k obtained above. Then, we have that

$\alpha_k = \frac{1}{\lambda}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$

$= \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$.

Consequently, we have proven that $\alpha_k = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$.