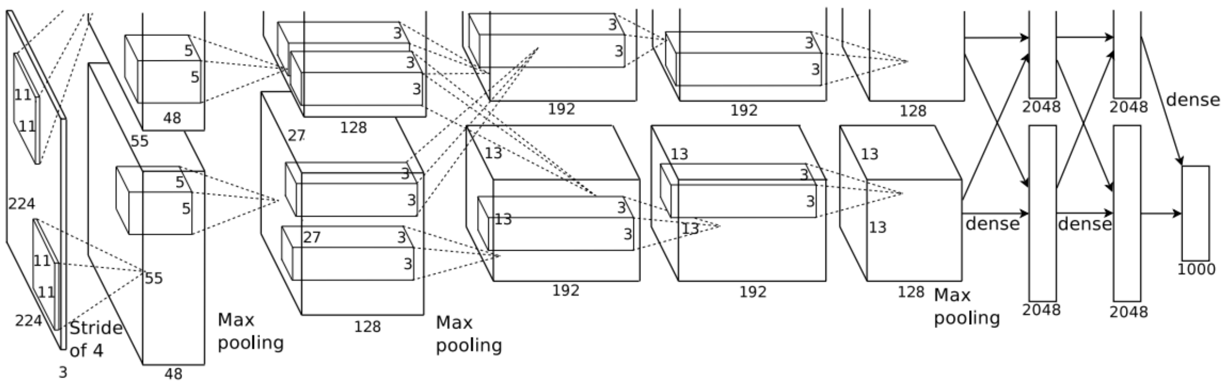
CSC411 A4

Zhihong Wang 1002095207

2018-11

Q1:

(a).



Definition:
Input size $= W \times H \times C$, W is the width, H is the height, C is the channel.
Square kernel size $= K$, Output maps $= M$, Layer $= L$

For convolutionnal L,
Number of neurons (output units) $= WHM = N$
Number of weights (parameters) $= K^2CM = P$
Number of connections $= WHK^2CM = U$

For fully connected L,
Number of neurons (output units) $= WHM = N$
Number of weights (parameters) $= W^2H^2CM = P$
Number of connections $= W^2H^2CM = U$

Input image: $W \times H \times C = 224 \times 224 \times 3 = 150528$
Convolutional layers: $L_1 - L_5$, Fully connected layers: $L_6 - L_7$, Output layer: L_8

L_1 :
96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels. W and H shrink by 4.
So we got $M = 96, K = 11, C = 3, W = H = 55$.
$N_1 = WHM = 55^2 \times 96 = 290400$
$P_1 = K^2CM = 11^2 \times 3 \times 96 = 34848$
$U_1 = WHK^2CM = 55^2 \times 11^2 \times 3 \times 96 = 105415200$

Note: The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer which reside on the same GPU. The kernels of the third convolutional layer are connected to all kernel maps in the second layer.

L_2:
256 kernels of size $5 \times 5 \times 48$.
So we got $M = 256, K = 5, C = 48, W = H = \frac{55}{2} = 27$.
$N_2 = WHM = 27^2 \times 256 = 186624$
$P_2 = K^2CM = 5^2 \times 48 \times 256 = 307200$
$U_2 = WHK^2CM = 27^2 \times 5^2 \times 48 \times 256 \div 2 = 111974400$

L_3:
384 kernels of size $3 \times 3 \times 256$.
So we got $M = 384, K = 3, C = 256, W = H = \frac{27}{2} = 13$.
$N_3 = WHM = 13^2 \times 384 = 64896$
$P_3 = K^2CM = 3^2 \times 256 \times 384 = 884736$
$U_3 = WHK^2CM = 13^2 \times 3^2 \times 256 \times 384 = 149520384$

L_4:
384 kernels of size $3 \times 3 \times 192$.
So we got $M = 384, K = 3, C = 192, W = H = 13$.
$N_4 = WHM = 13^2 \times 384 = 64896$
$P_4 = K^2CM = 3^2 \times 192 \times 384 = 663552$
$U_4 = WHK^2CM = 13^2 \times 3^2 \times 192 \times 384 = 112140288$

L_5:
256 kernels of size $3 \times 3 \times 192$.
So we got $M = 256, K = 3, C = 192, W = H = 13$.
$N_5 = WHM = 13^2 \times 256 = 43264$
$P_5 = K^2CM = 3^2 \times 192 \times 256 = 442368$
$U_5 = WHK^2CM = 13^2 \times 3^2 \times 192 \times 256 = 74760192$

L_6:
Has 4096 units.
So we got $C = 256, W = H = \frac{13}{2} = 6$.
$N_6 = WHM = 4096$
$P_6 = U_6 = W^2H^2CM = N \times WHC = 4096 \times 6 \times 6 \times 256 = 37748736$

L_7:
Has 4096 units.
So we got $C = M$.
$N_7 = WHM = 4096$
$P_7 = U_7 = W^2H^2CM = N \times N = 4096 \times 4096 = 16777216$

L_8:
Has 1000 units.
So we got $C = M$.
$N_8 = 1000$
$P_8 = U_8 = 4096 \times 1000 = 40960000$

Therefore, we got

	# Units	# Weights	# Connections
Conv Layer 1	290,400	34,848	105,415,200
Conv Layer 2	186,624	307,200	111,974,400
Conv Layer 3	64,896	884,736	149,520,384
Conv Layer 4	64,896	663,552	112,140,288
Conv Layer 5	43,264	442,368	74,760,192
Fully Connected Layer 1	4096	37,748,736	37,748,736
Fully Connected Layer 2	4096	16,777,216	16,777,216
Output Layer	1000	4,096,000	4,096,000

(b).

i. To reduce the memory usage, we can reduce the number of filters, or the number of feature maps, rather than directly reduce the number of parameter. By doing this, the depth of next layer will decrease, so the weights will decrease.

ii. To reduce the running time and the connections, we have many options. For example, we can reduce the size of convolutional layers, which affects the performance little, also make the time shorter. We can also reduce or replace the some fully connected layer with the conv layer. Reducing the parameters can also achieve the similar result.

Q2:

model for a discrete class label $y \in (1, 2, ..., k)$ and a real valued vector of d features $\mathbf{x} = (x_1, x_2, ..., x_d)$:

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

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

where α_k is the prior on class k, σ_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 kth row of $\boldsymbol{\mu}$ is the mean for class k.

(a). Use Bayes' rule to derive an expression for $p(y = k|x, \mu, \sigma)$

$$p(y = k|x, \mu, \sigma) = \frac{p(x|y=k,\mu,\sigma)p(y=k)}{\sum_k p(x|y=k,\mu,\sigma)p(y=k)}$$

$$= \frac{(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\}a_k}{\sum_k(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-1/2} exp\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\}a_k}$$

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

$$\ell(\theta; D) = -logp(y^{(1)}, x^{(1)}, y^{(2)}, x^{(2)}, ..., y^{(N)}, x^{(N)}|\theta)$$

of a particular dataset $D = \{(y^{(1)}, x^{(1)}), (y^{(2)}, x^{(2)}), ..., (y^{(N)}, x^{(N)})\}$ with parameters $\theta = \{\alpha, \mu, \sigma\}$. (Assume the data are i.i.d.)

$$-logp(y^{(1)}, x^{(1)}, ..., y^{(N)}, x^{(N)}|\theta) = -\sum_{i=1}^{N} logp(x^{(i)}|y^{(i)}, \theta) + logp(y^{(i)}|\theta)$$

$$= -\sum_{i=1}^{N} log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-1/2} exp\{-\sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j - \mu_{ij})^2\}] - \sum_{i=1}^{N} log\alpha_i$$

$$= -\sum_{i=1}^{N}[-\frac{1}{2}log(\prod_{j=1}^{D} 2\pi\sigma_j^2) - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j - \mu_{ij})^2] - \sum_{i=1}^{N} log\alpha_i$$

$$= -\sum_{i=1}^{N}[-\frac{1}{2}\sum_{j=1}^{D} log(2\pi\sigma_j^2) - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j - \mu_{ij})^2] - \sum_{i=1}^{N} log\alpha_i$$

$$= -\sum_{i=1}^{N}[-\frac{1}{2}\sum_{j=1}^{D}(log2\pi + log\sigma_j^2) - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j - \mu_{ij})^2] - \sum_{i=1}^{N} log\alpha_i$$

$$= -\sum_{i=1}^{N}[-\frac{D}{2}log2\pi - \frac{1}{2}\sum_{j=1}^{D} log\sigma_j^2 - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j - \mu_{ij})^2] - \sum_{i=1}^{N} log\alpha_i$$

$$= \frac{ND}{2}log2\pi + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D} log\sigma_j^2 + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D} \frac{1}{\sigma_j^2}(x_j^{(i)} - \mu_{ij})^2 - \sum_{i=1}^{N} log\alpha_i$$

$$= \frac{ND}{2}log2\pi + \frac{N}{2}\sum_{j=1}^{D} log\sigma_j^2 + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D} \frac{1}{\sigma_j^2}(x_j^{(i)} - \mu_{ij})^2 - \sum_{i=1}^{N} log\alpha_i$$

Note: α_i is the simplified version of $\alpha_{y^{(i)}}$. We can discuss $\alpha_{y^{(i)}}$ later at Q2.d. Similarly, μ_{ij} is the simplified version of $\mu_{y^{(i)}j}$. We will discuss $\mu_{y^{(i)}j}$ at Q2.c.

(c). Take partial derivatives of the likelihood with respect to each of the parameters μ_{ki} and with respect to the shared variances σ_i^2. Based on this, find the maximum likelihood estimates for μ and σ.

Assume each class appears at least once in the dataset.

Note: μ_{ij} is the simplified version of $\mu_{y^{(i)}j}$. x_{ij} is the simplified version of $x_j^{(i)}$.

$\frac{\partial(\ell(\theta;D))}{\partial\mu_{kj}}$

$= \frac{\partial(\frac{ND}{2}log2\pi+\frac{N}{2}\sum_{j=1}^{D}log\sigma_j^2+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}(x_j^{(i)}-\mu_{ij})^2-\sum_{i=1}^{N}log\alpha_i)}{\partial\mu_{kj}}$

$= \frac{\partial(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}(x_j^{(i)}-\mu_{ij})^2)}{\partial\mu_{kj}}$

$= \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\sigma_j^2}\frac{\partial((x_j^{(i)}-\mu_{ij})^2)}{\partial\mu_{kj}}$

Note: $y^{(i)} = k \longleftrightarrow \frac{\partial((x_j^{(i)}-\mu_{ij})^2)}{\partial\mu_{kj}} \neq 0$.

$= \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\sigma_j^2} \times -2(x_j^{(i_k)} - \mu_{kj})$

$= -\sum_{i=1}^{N}\frac{1}{\sigma_j^2}(x_j^{(i_k)} - \mu_{kj})$

$= -\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k](x_{ij} - \mu_{kj})\frac{1}{\sigma_j^2}$

Let $\frac{\partial(\ell(\theta;D))}{\partial\mu_{kj}} = 0$, $N_k = \sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]$

which is equal to let $\sum_{i_k=1}^{N_k}(x_j^{(i_k)} - \mu_{kj}) = 0$

then $\sum_{i_k=1}^{N_k}x_j^{(i_k)} - N_k \times \mu_{kj} = 0$

then $N_k \times \mu_{kj} = \sum_{i_k=1}^{N_k}x_j^{(i_k)}$

then $\mu_{kj} = \frac{1}{N_k}\sum_{i_k=1}^{N_k}x_j^{(i_k)}$

Therefore, the MLE for μ is:

$\hat{\mu}_{kj} = \frac{1}{N_k}\sum_{i_k=1}^{N_k}x_j^{(i_k)}$

same as

$\hat{\mu} = \frac{1}{N_k}\sum_{i=1}^{N}\mathbb{1}[y^{(i)} = k]x^{(i)}$

$\frac{\partial(\ell(\theta;D))}{\partial\sigma_j^2}$

$= \frac{\partial(\frac{ND}{2}log2\pi+\frac{N}{2}\sum_{j=1}^{D}log\sigma_j^2+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}(x_j^{(i)}-\mu_{ij})^2-\sum_{i=1}^{N}log\alpha_i)}{\partial\sigma_j^2}$

$= \frac{\partial(\frac{N}{2}\sum_{j=1}^{D}log\sigma_j^2+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}\frac{1}{\sigma_j^2}(x_j^{(i)}-\mu_{ij})^2}{\partial\sigma_j^2}$

$= \frac{N}{2\sigma_j^2} - \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\sigma_j^4}(x_j^{(i)} - \mu_{ij})^2$

Let $\frac{\partial(\ell(\theta;D))}{\partial\sigma_j^2} = 0$

then $\frac{N}{\sigma_j^2} = \frac{1}{\sigma_j^4}\sum_{i=1}^{N}(x_j^{(i)} - \mu_{ij})^2$

then $\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_j^{(i)} - \mu_{ij})^2$

Therefore, the MLE for σ^2 is:

$\hat{\sigma}_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_j^{(i)} - \mu_{ij})^2$

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

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

Assume each class appears at least once.

Note: α_k is not independent of each other.

Based on the result of Q2.b, to maximize $\ell(\theta; D)$, we need to find an $\alpha_{y^{(i)}}$ that makes $\sum_{i=1}^{N} log\alpha_{y^{(i)}}$ as small as possible.

Let $L(\alpha_{y^{(1)}}, ..., \alpha_{y^{(N)}}, \lambda) = \sum_{i=1}^{N} log\alpha_{y^{(i)}} - \lambda(\sum_k p(y = k) - 1)$

$= \sum_{i=1}^{N} log\alpha_{y^{(i)}} - \lambda(\sum_k \alpha_k - 1)$

Note: α_j is a random α, $\alpha_j \in [\alpha_1, ..., \alpha_k]$

$\frac{\partial L}{\partial \alpha_j} = \frac{\partial(\sum_{i=1}^{N} log\alpha_{y^{(i)}} - \lambda(\sum_k \alpha_k - 1))}{\partial \alpha_j} = \sum_{i=1}^{N} \frac{\partial(log\alpha_{y^{(i)}})}{\alpha_j} - \lambda\frac{\partial(\sum_k \alpha_k)}{\alpha_j}$

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

$\frac{\partial L}{\partial \lambda} = \frac{\partial(\sum_{i=1}^{N} log\alpha_{y^{(i)}} - \lambda(\sum_k \alpha_k - 1))}{\partial \lambda} = \sum_k \alpha_k - 1$

Let $\frac{\partial L}{\partial \alpha_j} = 0$

then $\frac{1}{a_j}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = j] = \lambda$

then $a_j = \frac{1}{\lambda}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = j]$

Note: Since $\alpha_k = p(y = k)$, so the sum of α_k is equal to the sum of $p(y = k)$, which is 1.

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

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

then $a_j = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = j]$

Since $j \in [1, ..., k]$

Therefore, we got $a_k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = k]$