CSC 411/2515 Fall 2018

Homework 4

Soon Chee Loong
999295793
cheeloong.soon@mail.utoronto.ca

Last Name: Soon
FirstName: Chee Loong
cdf markus: soonchee

1. AlexNet

a) Convolutional Net Architecture

5 Convolutional Layer

3 Fully Connected Layer

↳ includes 1 output layer

Count # of units, # of weights, # of connections in each layer.

Can ignore
- → units in pooling layers
- → connections between convolution & pooling layers.
- → biases.

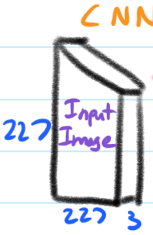
When counting the number of connections, adopt convention that when the input to a convolution layer is zero-padded, the connections to the dummy zero values count towards the total

⇒ number of incoming connections is the same for each unit in a given layer.

AlexNet Architecture

5 Convolutional Layers → 3 Fully Connected Layers ⇒ 5+3 = 8 layers

CNN Layers



GPU#1 → (55, 55, 48) → (27, 27, 128) → (13, 13, 192) → (13, 13, 192) → (13, 13, 128)

(55, 55, 48) → (27, 27, 128) → (", ", ") → (", ", ") → (", ", ")

Kernel (11x11x3)
stride=4
depth=96
GPU#2

(227, 227, 3) → note: 227 NOT 224

CNN Layer 1: $\frac{227-11}{4} + 1 = 55$

" " 2: $\frac{55+2(1)-5}{2} + 1 = 27 \Leftarrow$ (5x5x96), s. = 2, d. = 256, pad = 1

" " 3: $\frac{27-3}{2} + 1 = 13 \Leftarrow$ (3x3x256), s. = 2, d. = 384

" " 4: $\frac{13+2(7)-3}{2} + 1 = 13 \Leftarrow$ (3x3x384), s. = 2, d. = 384, p. = 1

" " 5: $\frac{13+2(7)-3}{2} + 1 = 13 \Leftarrow$ (3x3x384), s. = 2, d. = 256, p. = 7

Fully Connected Layers

(13, 13, 128) → (2048) → " → 1000 neurons ⇒ 1000 way softmax

(", ", ") → (2048) → "

FCNN 5: 4096 neurons

" 6: " "

" 7: 1000 "

weights for each convolution operation replicated 96 times

p. =>

Fully Connected Layers

(13, 13, 128) → (2048) → ... → 1000 neurons => 1000 way softmax
(", ", ") → (2048)

FCNN 5: 4096 neurons
" 6: " "
" 7: 1000 "

weights for each convolution operation replicated 96 times

Definitions

Input Related

I_h = input height

I_w = input width

I_d = input depth = # of input channels = f_d = filter depth

f_h = filter height

f_w = filter width

N = # of filters = O_d = output depth

O_h = output height

O_w = output width

For AlexNet, size = height = width => $I_h = I_w$, $f_h = f_w$, $O_h = O_w$

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} + 2(\text{Padding Size}) - \text{Filter Size}}{\text{Stride}} \right\rfloor + 1$$

Number of Neurons => number of non-linear activation => number of output units

Fully Connected

$\# \text{ of output} = O_h \times O_w \times O_d$

Convolutional

$O_h \times O_w \times O_d$

Number of Weights => number of parameters excluding biases
=> space complexity of model

Fully Connected

$(\# \text{ input}) \times (\# \text{ output})$
$= (I_h \times I_w \times I_d) \times (O_h \times O_w \times O_d)$

Convolutional

$(f_h \times f_w \times I_d) \times N$

Number of Connections => number of computations (multiply operations)
=> forward pass

Fully Connected => ∀ parameter multiplies once

$(I_h \times I_w \times I_d) \times (O_h \times O_w \times O_d)$

Fully Connected ⇒ ∀ parameter multiplies once

$(I_h \times I_w \times I_d) \times (O_h \times O_w \times O_d)$

Convolutional ⇒ Parameters are shared across slides

(# of parameters) × (# of slides of convolution)

$= ((f_h \times f_w \times I_d) \times N) \times (O_h \times O_w)$

Layer	# Units	# Weights	# Connections
Convolution Layer 1	55×55×96 = 290400	11×11×3×96 = 34848	(11×11×3×96)×(55×55) = 105415200
Convolution Layer 2	27×27×256 = 186624	5×5×48×256 = 307200	(5×5×48×256)×(27×27) = 223948800
Convolution Layer 3	13×13×384 = 64896	3×3×256×384 = 884736	(3×3×256×384)×(13×13) = 149520384
Convolution Layer 4	13×13×384 = 64896	3×3×192×384 = 663552	(3×3×192×384)×(13×13) = 112140288
Convolution Layer 5	13×13×256 = 43264	3×3×192×256 = 442368	(3×3×192×256)×(13×13) = 74760192
Fully Connected Layer 1	4096	9216×4096 = 37748736	9216×4096 = 37748736
Fully Connected Layer 2	4096	4096×4096 = 16777216	4096×4096 = 16777216
Output Layer	1000	4096×1000 = 4096000	4096×1000 = 4096000

b) Suggest a change to the architecture which will help achieve the desired objective.

⇒ Modify one or more layers.

i) Want to reduce memory usage at test time so network can run on a cellphone.

This requires reducing number of parameters for the network.

Trained parameters need to be stored in memory.

Fully Connected layers account for majority of the # of parameters.

∴, reduce the size of fully connected layer to reduce # of parameters

⇒ reduce memory usage.

ii) Network needs to make rapid predictions at test time.

Want to reduce the number of connections, since there is approximately one add-multiply operation per connection.

# of connections ⇒ # of forward pass computation ⇒ time complexity for predictions at test time.

Convolutional Layers account for majority of the # of connections.

∴, reduce number filters for the convolutional layer.

2. **Gaussian Naive Bayes**

Derive the max. likelihood estimates for Gaussian Naive Bayes, where the features are continuous, and the conditional distribution of each feature given the class is univariate Gaussian rather than Bernoulli. Start with a generative model for a discrete class label $y \in \{1, 2, \dots, k\}$ and a real valued vector of d features $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

$$P(y=k) = \alpha_k$$

$$P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) = \underbrace{\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}}}_{\text{each feature is conditionally independent given the discrete class label.}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2}$$

α_k = prior on class k

σ_i^2 = variances for each feature, shared between all classes

u_{ki} = mean of feature i conditioned on class k

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}, \quad \vec{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_d \end{bmatrix}, \quad \vec{u} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1d} \\ u_{21} & \cdots & \cdots & u_{2d} \\ \vdots & & \ddots & \vdots \\ u_{k1} & \cdots & \cdots & u_{kd} \end{bmatrix}$$

$D = d$ in this question

a) Use Bayes rule to derive an expression for $p(y=k \mid \vec{x}, \vec{u}, \vec{\sigma})$.

Hint: Use law of total probability to derive an expression for $p(\vec{x} \mid \vec{u}, \vec{\sigma})$

$$\begin{aligned} P(\vec{x}, y=k \mid \vec{u}, \vec{\sigma}) &= P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) P(\vec{x} \mid \vec{u}, \vec{\sigma}) \\ &= P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k) \end{aligned}$$

$$\Rightarrow P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) = \frac{P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k)}{P(\vec{x} \mid \vec{u}, \vec{\sigma})}$$

$$P(\vec{x} \mid \vec{u}, \vec{\sigma}) = \sum_{j=1}^{k} P(\vec{x}, y=j \mid \vec{u}, \vec{\sigma}) = \sum_{j=1}^{k} P(\vec{x} \mid y=j, \vec{u}, \vec{\sigma}) P(y=j), \text{ marginalize out } y$$

$$\Rightarrow P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) = \frac{P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k)}{\sum_{j=1}^{k} P(\vec{x} \mid y=j, \vec{u}, \vec{\sigma}) P(y=j)}$$

$$= \frac{\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2} (\alpha_k)}{\sum_{j=1}^{k} \left(\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} (\alpha_j) \right)}$$

$$= \frac{\left(\prod_{i=1}^{D} 2\cancel{\pi}\sigma_i^2\right)^{-\frac{1}{2}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2} (\alpha_k)}{\underbrace{\left(\prod_{i=1}^{D} 2\cancel{\pi}\sigma_i^2\right)^{-\frac{1}{2}}}_{\text{independent of } j} \sum_{j=1}^{k} \left(e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} (\alpha_j) \right)}$$

$$-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2 (\alpha$$

a) Use Bayes rule to derive an expression for $p(y=k \mid \vec{x}, \vec{u}, \vec{\sigma})$.

Hint: Use law of total probability to derive an expression for $p(\vec{x} \mid \vec{u}, \vec{\sigma})$

$$P(\vec{x}, y=k \mid \vec{u}, \vec{\sigma}) = P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) P(\vec{x} \mid \vec{u}, \vec{\sigma})$$

$$= P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k)$$

$$\Rightarrow P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) = \frac{P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k)}{P(\vec{x} \mid \vec{u}, \vec{\sigma})}$$

$$P(\vec{x} \mid \vec{u}, \vec{\sigma}) = \sum_{j=1}^{K} P(\vec{x}, y=j \mid \vec{u}, \vec{\sigma}) = \sum_{j=1}^{K} P(\vec{x} \mid y=j, \vec{u}, \vec{\sigma}) P(y=j), \text{ marginalize out } y$$

$$\Rightarrow P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) = \frac{P(\vec{x} \mid y=k, \vec{u}, \vec{\sigma}) P(y=k)}{\sum_{j=1}^{K} P(\vec{x} \mid y=j, \vec{u}, \vec{\sigma}) P(y=j)}$$

$$= \frac{\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2} (\alpha_k)}{\sum_{j=1}^{K} \left(\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} (\alpha_j) \right)}$$

$$= \frac{\cancel{\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}}} e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2} (\alpha_k)}{\underbrace{\cancel{\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}}}}_{\text{independent of } j} \sum_{j=1}^{K} \left(e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} (\alpha_j) \right)}$$

$$= \frac{e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2} (\alpha_k)}{\sum_{j=1}^{K} \left(e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} (\alpha_j) \right)}$$

$\therefore$,

$$P(y=k \mid \vec{x}, \vec{u}, \vec{\sigma}) = \frac{(\alpha_k) e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ki})^2}}{\sum_{j=1}^{K} \left((\alpha_j) e^{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - u_{ji})^2} \right)}$$

a specific label $= k \neq K =$ # of labels

b) Write down an expression for the negative likelihood function

$$\ell(\theta; D) = -\log P(y^{(1)}, \vec{x}^{(1)}, y^{(2)}, \vec{x}^{(2)}, \dots, y^{(N)}, \vec{x}^{(N)} \mid \vec{\theta})$$

Dataset: $D = \{(y^{(1)}, \vec{x}^{(1)}), (y^{(2)}, \vec{x}^{(2)}), \dots, (y^{(N)}, \vec{x}^{(N)})\}$, i.i.d.

Parameters $= \vec{\theta} = \{\vec{\alpha}, \vec{u}, \vec{\sigma}\}$

$$-\log\left(P(y^{(1)}, \vec{x}^{(1)}, y^{(2)}, \vec{x}^{(2)}, \dots, y^{(N)}, \vec{x}^{(N)} \mid \vec{\theta})\right)$$

$$= -\log\left(\prod_{i=1}^{N} P(y^{(i)}, \vec{x}^{(i)} \mid \vec{\theta})\right) \quad \text{i.i.d.}$$

b) Write down an expression for the negative likelihood function

$$\ell(\Theta; D) = -\log P(y^{(1)}, \vec{x}^{(1)}, y^{(2)}, \vec{x}^{(2)}, \dots, y^{(N)}, \vec{x}^{(N)} \mid \vec{\theta})$$

Dataset: $D = \{(y^{(1)}, \vec{x}^{(1)}), (y^{(2)}, \vec{x}^{(2)}), \dots, (y^{(N)}, \vec{x}^{(N)})\}$, i.i.d.

Parameters: $\vec{\theta} = \{\vec{\alpha}, \vec{u}, \vec{\sigma}\}$

$$-\log\left(P(y^{(1)}, \vec{x}^{(1)}, y^{(2)}, \vec{x}^{(2)}, \dots, y^{(N)}, \vec{x}^{(N)} \mid \vec{\theta})\right)$$

$$= -\log\left(\prod_{i=1}^{N} P(y^{(i)}, \vec{x}^{(i)} \mid \vec{\theta})\right) \text{, since data are i.i.d.}$$

$$= -\sum_{i=1}^{N} \log\left(P(y^{(i)}, \vec{x}^{(i)} \mid \vec{\theta})\right) \text{, since } \log\left(\prod_i \dots\right) = \sum_i \left(\log(\dots)\right)$$

$$= -\sum_{i=1}^{N} \log\left(P(\vec{x}^{(i)} \mid y^{(i)} \mid \vec{\theta})\, P(y^{(i)} \mid \vec{\theta})\right)$$

$$= -\sum_{i=1}^{N} \left(\log\left(P(\vec{x}^{(i)} \mid y^{(i)} \mid \vec{\theta})\right) + \log\left(P(y^{(i)})\right)\right), \quad P(y^{(i)} \mid \vec{\theta}) = P(y^{(i)}) = \alpha_{y^{(i)}}$$

$$= -\sum_{i=1}^{N} \left(\log\left(\left(\prod_{j=1}^{D} 2\pi\sigma_j^2\right)^{-\frac{1}{2}} e^{-\sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j^{(i)} - u_{y^{(i)}j})^2}\right) + \log(\alpha_{y^{(i)}})\right)$$

$$= -\sum_{i=1}^{N} \left(-\frac{1}{2}\sum_{j=1}^{D} \log(2\pi\sigma_j^2) + \left(-\sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j^{(i)} - u_{y^{(i)}j})^2\right) + \log(\alpha_{y^{(i)}})\right)$$

$\therefore$,

Negative Log Likelihood

$$\ell(\Theta, D) = -\sum_{i=1}^{N} \left(-\frac{1}{2}\sum_{j=1}^{D} \log(2\pi\sigma_j^2) - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j^{(i)} - u_{y^{(i)}j})^2 + \log(\alpha_{y^{(i)}})\right)$$

c) Take partial derivatives of the likelihood w.r.t. each of the parameters u_{ki} and w.r.t. the shared variances σ_i^2. Based on this, find the max. likelihood estimates for $\vec{u}$ and $\vec{\sigma}$.

Assume each class appears at least once in the dataset.

Partial derivatives w.r.t. u_{ki}

$$\frac{\partial \ell(\Theta, D)}{\partial u_{ki}}$$

$$= \frac{\partial}{\partial u_{ki}}\left(-\sum_{m=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D} \cancel{\log(2\pi\sigma_j^2)} - \sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j^{(m)} - u_{y^{(m)}j})^2 + \cancel{\log(\alpha_{y^{(m)}})}\right)\right)$$

0 since constant w.r.t. u_{ki} 0 since const. w.r.t. u_{ki}

$$= \frac{\partial}{\partial u_{ki}}\left(\sum_{m=1}^{N}\sum_{j=1}^{D} \frac{1}{2\sigma_j^2}(x_j^{(m)} - u_{y^{(m)}j})^2\right)$$

$$= \sum_{m=1}^{N} \mathbb{1}[y^{(m)} = k] \frac{2}{2\sigma_i^2}(x_i^{(m)} - u_{ki})(-1)$$

$$\therefore, \frac{\partial(\ell(\Theta, D))}{\partial u_{ki}} = \sum_{m=1}^{N} \mathbb{1}[y^{(m)} = k] \frac{-1}{\sigma_i^2}(x_i^{(m)} - u_{ki})$$

Maximum Likelihood Estimate for $\vec{u}$

$\Rightarrow$ Set $\dfrac{\partial(\ell(\Theta, D))}{\partial u_{ki}} = 0$ and solve for u_{ki}

Maximum Likelihood Estimate for $\vec{u}$

$\Rightarrow$ Set $\dfrac{\partial(\ell(\theta,D))}{\partial u_{ki}} = 0$ and solve for u_{ki}

$$\Rightarrow \sum_{m=1}^{N} \mathbb{1}[y^{(m)}=k] \frac{-1}{\sigma_i^2}(x_i^{(m)}-u_{ki}) = 0$$

$$\Rightarrow u_{ki_{M.L.E.}} = \frac{\sum_{m=1}^{N} \mathbb{1}[y^{(m)}=k]\, x_i^{(m)}}{\sum_{m=1}^{N} \mathbb{1}[y^{(m)}=k]}$$

$$\Rightarrow \vec{u} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1d} \\ u_{21} & \cdots & \cdots & u_{2d} \\ \vdots & & \ddots & \vdots \\ u_{k1} & \cdots & \cdots & u_{kd} \end{bmatrix}_{(k\times d)}, \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}_{(d\times 1)}, \quad \text{diag}(\vec{x}) = \begin{bmatrix} x_1 & & 0 \\ & x_2 & \\ 0 & & \ddots \\ & & & x_d \end{bmatrix}_{(d\times d)}$$

$$= \begin{bmatrix} \vec{u}_1^T \\ \vec{u}_2^T \\ \vdots \\ \vec{u}_k^T \end{bmatrix}_{(k\times d)}, \quad \vec{u}_k \in (d\times 1)$$

$\therefore$,

$$\vec{u}_{M.L.E.} = \begin{bmatrix} \vec{u}_{1_{M.L.E.}}^T \\ \vec{u}_{2_{M.L.E.}}^T \\ \vdots \\ \vec{u}_{k_{M.L.E.}}^T \end{bmatrix}$$

where

$$\vec{u}_{k_{M.L.E.}(d\times 1)} = \frac{\sum_{m=1}^{N} \mathbb{1}[y^{(m)}=k]\, \vec{x}^{(m)}}{\sum_{m=1}^{N} \mathbb{1}[y^{(m)}=k]}, \quad k \in \{1,2,\ldots,K\}$$

Partial derivatives w.r.t. σ_i^2

$$\frac{\partial \ell(\theta,D)}{\partial \sigma_i^2}$$

$$= \frac{\partial}{\partial \sigma_i^2}\left(-\sum_{m=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D}\log(2\pi\sigma_j^2) - \sum_{j=1}^{D}\frac{1}{2\sigma_j^2}(x_j^{(m)}-u_{y^{(m)}j})^2 + \underbrace{\log(\alpha_{y^{(m)}})}_{0}\right)\right)$$

0 since constant w.r.t. σ_i^2

$$= -\sum_{m=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{\partial(\log(2\pi\sigma_j^2))}{\partial\sigma_i^2} - \sum_{j=1}^{D}(x_j^{(m)}-u_{y^{(m)}j})^2\frac{\partial\left(\frac{1}{2\sigma_j^2}\right)}{\partial\sigma_i^2}\right)$$

$$= \sum_{m=1}^{N}\left(\left(\frac{1}{2}\right)\frac{1}{2\pi\sigma_i^2}2\pi + (x_i^{(m)}-u_{y^{(m)}i})^2\left(\frac{-1}{2\sigma_i^4}\right)\right)$$

Fall2018 » CSC2515

Partial derivatives w.r.t. σ_i^2

$$\frac{\partial \ell(\Theta,D)}{\partial \sigma_i^2}$$

$$= \frac{\partial}{\partial \sigma_i^2}\left(-\sum_{m=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D}\log(2\pi\sigma_j^2) - \sum_{j=1}^{D}\frac{1}{2\sigma_j^2}(x_j^{(m)}-u_{y^{(m)}j})^2 + \cancel{\log(\alpha_{y^{(m)}})}\right)\right)$$

0 since constant w.r.t. σ_i^2

$$= -\sum_{m=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{\partial(\log(2\pi\sigma_j^2))}{\partial\sigma_i^2} - \sum_{j=1}^{D}(x_j^{(m)}-u_{y^{(m)}j})^2\frac{\partial\left(\frac{1}{2\sigma_j^2}\right)}{\partial\sigma_i^2}\right)$$

$$= \sum_{m=1}^{N}\left(\left(\frac{1}{2}\right)\frac{1}{2\pi\sigma_i^2}2\pi + (x_i^{(m)}-u_{y^{(m)}i})^2\left(\frac{-1}{2\sigma_i^4}\right)\right)$$

$$= \sum_{m=1}^{N}\frac{1}{2\sigma_i^2} - \frac{(x_i^{(m)}-u_{y^{(m)}i})^2}{2\sigma_i^4}$$

$\therefore$

$$\frac{\partial \ell(\Theta,D)}{\partial \sigma_i^2} = \sum_{m=1}^{N}\frac{1}{2\sigma_i^2} - \frac{(x_i^{(m)}-u_{y^{(m)}i})^2}{2\sigma_i^4}$$

4 try

Maximum Likelihood Estimate for $\vec{\sigma}$

$\Rightarrow$ set $\frac{\partial \ell(\Theta,D)}{\partial\sigma_i^2} = 0$ and solve for σ_i

$$\Rightarrow \sum_{m=1}^{N}\frac{1}{2\sigma_i^2} - \frac{(x_i^{(m)}-u_{y^{(m)}i})^2}{2\sigma_i^4} = 0$$

Multiply both sides by $2\sigma_i^4$

$\Rightarrow$

$$\sum_{m=1}^{N}\sigma_i^2 - (x_i^{(m)}-u_{y^{(m)}i})^2 = 0$$

$$\Rightarrow \sigma_i^2 = \frac{\sum_{m=1}^{N}(x_i^{(m)}-u_{y^{(m)}i})^2}{N}$$

$$\Rightarrow \sigma_{i_{M.L.E.}} = \sqrt{\frac{\sum_{m=1}^{N}(x_i^{(m)}-u_{y^{(m)}i})^2}{N}}$$

$$\vec{\sigma}_{M.L.E.} = \begin{bmatrix}\sigma_{1_{M.L.E.}}\\ \sigma_{2_{M.L.E.}}\\ \vdots \\ \sigma_{d_{M.L.E.}}\end{bmatrix}$$

$\therefore$

$$\vec{\sigma}_{M.L.E.} = \sqrt{\frac{\sum_{m=1}^{N}(\vec{x}^{(m)}-\vec{u}_{y^{(m)}})^2}{N}}$$

d) Show that the M.L.E. for α_k is given by

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

Assume each class appears at least once.

Hint: Use Lagrange Multipliers.

Since α_k is the prior probability, $P(y=k)$, will need to satisfy constraint $\sum_{j=1}^{K}\alpha_j = 1$ to be a valid probability distribution.

Lagrange Multiplier

Use Lagrange Multiplier to turn this constrained optimization probl into an unconstrained optimization problem.

Minimize $f(z)$

s.t. $g(z)=0$

$$L(z,\lambda) = f(z) + \lambda g(z)$$

$$\Rightarrow f(z) = \ell(\theta, D)$$

$$g(z) = \sum_{j=1}^{K}\alpha_j - 1 = 0$$

$$\Rightarrow L(z,\lambda) = \ell(\theta,D) + \lambda\left(\sum_{j=1}^{K}\alpha_j - 1\right) \text{, set } \lambda = 1$$

$$\frac{\partial L(z,\lambda)}{\partial \alpha_k}$$

$$= \frac{\partial}{\partial\alpha_k}\left(-\sum_{i=1}^{N}\left(-\frac{1}{2}\sum_{j=1}^{D}\log(2\pi\sigma_j^2) - \sum_{j=1}^{D}\frac{1}{2\sigma_j^2}(x_j^{(i)} - \mu_{y^{(i)}j})^2 + \log(\alpha_{y^{(i)}})\right) + \sum_{i=1}^{N}\left(\sum_{q=1}^{K}\alpha_q - 1\right)\right)$$

0 since constant w.r.t. α_k 0 since constant w.r.t. α_k

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

$$\frac{\partial L(z,\lambda)}{\partial \alpha_k} = \sum_{i=1}^{N}\left(\mathbb{1}[y^{(i)}=k]\left(-\frac{1}{\alpha_k}\right) + 1\right)$$

Set $\frac{\partial L(z,\lambda)}{\partial \alpha_k} = 0$ and solve for α

$$\Rightarrow \sum_{i=1}^{N}\left(\mathbb{1}[y^{(i)}=k]\left(-\frac{1}{\alpha_k}\right) + 1\right) = 0$$

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

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

$$\Rightarrow \alpha_{k_{M.L.E}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}[y^{(i)}=k]$$ $\therefore$, proven