CSC411 Fall 2018: Homework 4

Niloufar Afsariardchi

October 31, 2018

1 Problem 1

a) this is my table results:

	#units	#weights	#connections
1st conv	55*55*96=290400	11*11*3*96=34848	290400*11*11*3=105415200
2nd conv	27*27*256=186624	5*5*48*256=307200	186624*5*5*48=223948800
3rd conv	13*13*384=64896	3*3*256*384=884736	64896*3*3*256=149520384
4th conv	13*13*384=64896	3*3*192*384=663552	64896*3*3*192=112140288
5th conv	13*13*256=43264	3*3*192*256=442368	43264*3*3*192=74760192
1st ful. con.	4096	6*6*256*4096=37748736	37748736
2nd ful. con.	4096	4096*4096=16777216	16777216
Output	1000	4096*1000=4096000	4096000

Several notes regarding my results above: 1) I realized that the paper has an error and the input layer has the size of 227*227*3 not 224*224*3. This error results in a wrong number of units for first later as can be seen in the caption of figure 2. 2) for the first fully connected layer, the max pooling layer from previous layer cannot be ignored, therefore I computed the output size of the pooling layer with stride S=2 and width W=3 giving ouput size of (N-W/S)+1=6. The feature map has the size of 6*6 which is a shrinkage from original 13*13. 3) Note that apart from output and fully connected layers, only third convolutional layer is connected to the output of the other GPU from previous layer. Therefore only this layer has full weights and number of connections.

b) i. Smarphones have a typical memory of a few GBs depending on the model. This memory is not enough for storing 60 millions double precision float parameters. Therefore we should reduce the number of parameters so that they take maximum few hundred of MBs. Towards this end, we could reduce the number of parameters by a factor 2, so that we will have 30 million parameters instead. 30 million parameters takes roughly ∼240 MB of the memory which seems reasonable. Since most of the parameters are defined in the fully connected layers, I would increase the max pooling stride in fifth convolutional layer to 5 so that the number of parameters of 1st fully connected layer reduce by a factor of 4 (because the input size to that layer would shrink to 3*3*256). I would also reduce the number of hidden units in 1st and 2nd fully connected layers to 2048 instead of 4096. So in total, the number of parameters in 1st fully connected layer, 2nd fully connected layer, and output layer would shrink by a factor of 8, 4, and 2 respectively and we will achieve the desired 30 million parameters goal.

b) ii. the number of connections are highest in the convolution layers. So I would either reduce the depth of each layer (i.e., the number of filters) by a factor of e.g., 2 or increase the stride in the pooling layers so that the width and the heights of each layer would decrease. We could also decrease the width and height of filters. Among these solutions of course we need to test to see which one works better, but I think it's better not to reduce the depth of each much, because the depth is very important in finding the hidden patterns of data. Additionally, the widths and heights of the filters are already small in AlexNet (e.g., 3*3 and 5*5), so I would say increasing the stride of the pooling layers would possibly work better in comparison to other solutions.

2 Problem 2

Continue to next page

Question 2

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

$$= \frac{P(x,\mu,\sigma\,|\,y=k)\,p(y=k)}{\sum_{j=1}^{K} p(x,\mu,\sigma\,|\,y=j)\,p(y=j)}$$

$$= \frac{\alpha_K \times \left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-1/2} \exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{Ki})^2\right\}}{\sum_{j=1}^{K} \alpha_j\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-1/2} \exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ji})^2\right\}}$$

$$= \frac{\alpha_K \exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{Ki})^2\right\}}{\sum_{j=1}^{K} \alpha_i \exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ji})^2\right\}}$$

$$= \frac{1}{\sum_{j=1}^{K}\left(\frac{\alpha_i}{\alpha_K}\right)\exp\left\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{j})^2 + \sum_{i=1}^{D}\frac{1}{2\sigma^2}(x_i-\mu_{Kj})^2\right\}}$$

$$= \frac{1}{\sum_{j=1}^{K}\left(\frac{\alpha_i}{\alpha_K}\right)\exp\left\{\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}\left[(x_i-\mu_{Kj})^2 - (x_i-\mu_{ij})^2\right]\right\}}$$

$$= \frac{1}{\sum_{j=1}^{K}\left(\frac{\alpha_i}{\alpha_K}\right)\exp\left\{\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}\left(\mu_{Kj}^2 - 2x_i(\mu_{Kj}-\mu_{ij}) - \mu_{ij}^2\right)\right\}}$$

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

$$= -\log \prod_{j=1}^{N} P(y^{(j)}, x^{(j)} | \theta)$$

$$= -\log \prod_{j=1}^{N} P\left(x^{(j)} | \theta, y^{(j)}\right) P(y^{(j)} | \theta)$$

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

$$= -\sum_{j=1}^{N} \log\left(x^{(j)} | \theta, y^{(j)} = k^{(j)}\right) - \sum_{j=1}^{N} \log P(y^{(j)} | \alpha) \longleftarrow \boxed{\begin{array}{l} P(y|\alpha) \\ \text{has a multinomial} \\ \text{distribution} \end{array}}$$

$$= -N \log\left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-\frac{1}{2}} + \sum_{j=1}^{N} \sum_{i=1}^{D} \frac{1}{2\sigma_i^2} (x_i^{(j)} - \mu_{k^{(j)} i})^2 - \sum_{j=1}^{N} P(y^{(j)} | \alpha)$$

$\Big\downarrow$ removing constant terms

$$\simeq \sum_{j=1}^{N} \sum_{i=1}^{D} \frac{1}{2\sigma_i^2} (x_i^{(j)} - \mu_{k^{(j)} i})^2 - \frac{N!}{u_1! u_2! \dots u_k!} \prod_{\nu=1}^{k} \alpha_\nu^{u_\nu} + N \log\left(\prod_{i=1}^{D} \sigma_i\right)$$

Note that I denoted the label of $y^{(j)}$ with $k^{(j)}$
where u_k denotes the # of times that class k-th occur.

$$\begin{cases} \sum_{\nu=1}^{K} u_\nu = N \\ \sum_{k=1}^{K} \alpha_k = 1 \end{cases}$$

c) $$\boxed{\frac{\partial l(\theta;D)}{\partial \mu_{ki}} = \sum_{j=1}^{N} \frac{1}{\sigma_i^2}(x_i^{(j)} - \mu_{ki})\, \mathbb{1}(y^{(j)} = k)}$$

$$\frac{\partial l(\theta;D)}{\partial \sigma_i} = N \frac{1}{\prod_{u=1}^{D} \sigma_u} \prod_{u \neq i} \sigma_u - \frac{1}{\sigma_i^3} \sum_{j=1}^{N} (x_i^{(j)} - \mu_{ki})^2$$

$$= \boxed{\frac{N}{\sigma_i} - \frac{1}{\sigma_i^3} \sum_{j=1}^{N} (x_i^{(j)} - \mu_{ki})^2}$$

For deriving ML estimate:

$$\frac{\partial l(\theta;D)}{\partial \mu_{ki}} = 0 \implies \boxed{\mu_{ki} = \sum_{j=1}^{N} x_i^{(j)} \mathbb{1}(y^{(j)} = k) \Big/ \sum_{j=1}^{N} \mathbb{1}(y^{(j)} = k)}$$

$$\frac{\partial l(\theta,D)}{\partial \sigma_i} = 0 \implies \frac{N}{\cancel{\sigma_i}} = \frac{1}{\sigma_i^{3}} \sum_{j=1}^{N} (x_i^{(j)} - \mu_{k^{(j)}i})^2$$

$$\implies \boxed{\sigma_i^2 = \frac{1}{N} \sum_{j=1}^{N} (x_i^{(j)} - \mu_{k^{(j)}i})^2} \leftarrow \text{where } k^{(j)} \text{ is the label of } y^{(j)}$$

d) Let's assume that the k-th class occurs u times, this means the term in $l(\theta;D)$ that has α_k would be:

$$\mathcal{L} = \log \prod_{j=1}^{N} P(y^{(j)} = k \mid \alpha) = \log \alpha_k^{u} (1-\alpha_k)^{N-u} = u \log \alpha_k + N-u \log(1-\alpha_k)$$

ML estimation (Note other terms don't have α_k in $l(\theta;D)$)

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = \frac{u}{\alpha_k} - \frac{N-u}{1-\alpha_k} \overset{\swarrow}{=} 0 \implies \frac{u}{\alpha_k} = \frac{N-u}{1-\alpha_k}$$

$$\implies u - \cancel{u\alpha_k} = N\alpha_k - \cancel{u\alpha_k} \implies \boxed{\alpha_k = \frac{u}{N} = \frac{\sum_{j=1}^{N} \mathbb{1}(y^{(j)} = k)}{N}}$$