Theoretical part
a) 5 / 7
b) 8 / 8
b) 2 / 6
Total 15 / 21
Practical part
a) 3 / 5
b) 5 / 5
c) 4 / 5
Total 12 / 15

Deep Learning for Visual Recognition - Assignment 5

Mayara E. Bonani, Guillaume Rouvarel, Arash Safavi, Vardeep Singh, Cüneyt Erem

January 1, 2021

Theoretical Part

5 / 7 pts

a) Mean square error and estimators

1) From the definition, we know that;

$$\Pr(y|x,\theta) = N(y|f(x;\theta),\sigma^2) \tag{1}$$

and the model can be defined as;

$$\log \Pr(y|x,\theta) = \sum_{i=1}^{m} \log \Pr(y_i|x_i,\theta) \tag{2}$$

$$\log \Pr(y|x,\theta) = \sum_{i=1}^{m} N(y_i|f(x_i;\theta),\sigma^2) \tag{3}$$

$$\log \Pr(y|x,\theta) = -\frac{m}{2}\log 2\pi\sigma_j^2 - \sum_{i=1}^{m}\frac{(y_i - f(x_i;\theta))^2}{2\sigma_j^2} \tag{4}$$

then theta becomes;

$$\theta_{MSE} = argmax_\theta - \sum_{i=1}^{m}(y_i - f(x_i;\theta))^2 \tag{5}$$

$$\theta_{MSE} = argmin_\theta \sum_{i=1}^{m}(y_i - f(x_i;\theta))^2 \tag{6}$$

So, the maximizing the log-likelihood depends on the minimizing the mean-squared error ✓

(7)

2) The Gaussian likelihood is defined by; $N(y|f(x;\theta),\sigma^2)$ and if we regulate the theta with Gaussian Prior, $N(\theta|0,1/\lambda)$. Then we have;

$$N(y|f(x;\theta),\sigma^2)N(\theta|0,1/\lambda) \tag{8}$$

$$\prod_{i=1}^{m} N(y_i|f(x_i;\theta),\sigma^2)N(\theta|0,1/\lambda) \tag{9}$$

Which is equivalent to this expression for the l2 regularization;

$$argmax_\theta \sum_{i=1}^{m} -(\frac{(y_i - f(x_i;\theta))^2}{\sigma_j^2} + \lambda\theta^2) \tag{10}$$

If you look at this in more detail, you have to be careful how to choose the variance of the prior for the factors to cancel out correctly.

$$argmax_\theta \sum_{i=1}^{m} -((y_i - f(x_i;\theta))^2 + \lambda\theta^2) \tag{11}$$

We know that Bayesian principle is posterior = likelihood $\times$ prior or

$$log(posterior) = log(likelihood) + log(l2regularization) \tag{12}$$

Then we can say that minimizing MSE with l2 regularization depends on maximizing the posterior for the posteriori of the theta by looking at Bayesian, ✓

(13)

3)From the previous expressions, we found that maximum log likelihood;

$$\theta_{MLE} = argmax_\theta \Pr(y|x,\theta) \tag{14}$$

$$\theta_{MLE} = argmax_\theta - \sum_{i=1}^{m}(y_i - f(x_i;\theta))^2 \tag{15}$$

and maximum posteriori;

$$\theta_{MAP} = argmax_\theta \sum_{i=1}^{m} -((y_i - f(x_i;\theta))^2 + \lambda\theta^2) \tag{16}$$

Why/When is this term constant?

$$\theta_{MAP} = argmax_\theta - \sum_{i=1}^{m}(y_i - f(x_i;\theta))^2 + constant \tag{17}$$

$$\theta_{MAP} = argmax_\theta - \sum_{i=1}^{m}(y_i - f(x_i;\theta))^2 \tag{18}$$

Then

$$\theta_{MAP} = \theta_{MLE} \tag{19}$$

So, maximum likelihood estimation is different form of maximum a posteriori estimation

8 / 8 pts

b) Cross entropy loss and label smoothing

- **Probability distribution over class labels** For the CIFAR-10, we have 10 candidate labels. For the sample i from the training dataset, (x_i, y_i), we have:
  - $p(y|x_i)$: ground truth distribition p over the 10 labels , and $\sum_{y=1}^{10} p(y|x_i) = 1$. We define $p(y|x_i)$ as one-hot encoded vector:

$$p(y|x_i) = \begin{cases} 1 & \text{if } y = y_i \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

  - Model with parameters θ that represents the predicted label distribution as $q_\theta(y|x_i)$, with $\sum_{y=1}^{10} q_\theta(y|x_i) = 1$. It is computed from softmax function:

$$q_\theta(y|x_i) = \frac{\exp(z_{y_i})}{\sum_{j=1}^{10} \exp(z_j)}, \tag{21}$$

    where z_j is the logit for candidate class i.
  - Cross Entropy:

$$\begin{aligned} H_i(p, q_\theta) &= -\sum_{y=1}^{10} p(y|x_i) \log q_\theta(y|x_i) \\ &= -\log q_\theta(y_i|x_i) \end{aligned} \tag{22}$$

  - Loss function for $n = 60000$ examples in the training dataset:

$$\begin{aligned} L &= \sum_{i=1}^{n} H_i(p, q_\theta) \\ &= -\sum_{i=1}^{n} \sum_{y=1}^{10} p(y|x_i) \log q_\theta(y|x_i) \\ &= -\sum_{i=1}^{n} p(y_i|x_i) \log q_\theta(y_i|x_i) \\ &= -\sum_{i=1}^{n} \log q_\theta(y_i|x_i) \end{aligned}$$

✓

- **Maximum Likelihood Estimation and Cross Entropy**

  As it was deduced above, the cross entropy of the ground truth distribution and the predicted label distribution is given by the equation 22, and the loss function(log loss function) by the

equation 23. We have that:

$$\underset{\theta}{\operatorname{argmin}}\ L = \underset{\theta}{\operatorname{argmin}}\ \sum_{i=1}^{n} H_i(p, q_\theta) \tag{20}$$

$$= \underset{\theta}{\operatorname{argmin}}\ -\sum_{i=1}^{n} \log q_\theta(y_i|x_i) \tag{21}$$

✓

Therefore, minimizing the cross entropy (without label smoothing) is equivalent to maximum likelihood estimation by using this model.

- **Label Smoothing** Considering the noise distribution $u(y|x)$, we obtain:
  - Ground truth label for data (x_i, y_i) for a weight factor $\varepsilon \in [0, 1]$ and $\sum_{y=1}^{10} p'(y|x_i) = 1$:

$$p'(y|x_i) = (1-\varepsilon)p(y|x_i) + \varepsilon u(y|x_i) \tag{22}$$

$$= \begin{cases} 1 - \varepsilon + \varepsilon u(y|x_i) & \text{if } y = y_i \\ \varepsilon u(y|x_i) & \text{otherwise} \end{cases} \tag{23}$$

  - Loss Function:

$$L' = -\sum_{i=1}^{n}\sum_{y=1}^{10} p'(y|x_i) \log q_\theta(y|x_i) \tag{24}$$

$$= -\sum_{i=1}^{n}\sum_{y=1}^{10} \big[(1-\varepsilon)p(y|x_i) + \varepsilon u(y|x_i)\big] \log q_\theta(y|x_i) \tag{25}$$

$$= \sum_{i=1}^{n} \Big\{ (1-\varepsilon)\Big[-\sum_{y=1}^{10} p(y|x_i) \log q_\theta(y|x_i)\Big] + \varepsilon\Big[-\sum_{y=1}^{10} u(y|x_i) \log q_\theta(y|x_i)\Big] \Big\} \tag{26}$$

$$= \sum_{i=1}^{n} \Big[(1-\varepsilon) H_i(p, q_\theta) + \varepsilon H_i(u, q_\theta) \Big] \tag{27}$$

✓

$$\tag{28}$$

  - Considering the uniform distribution $u = 1/K = 1/10$, we have:

$$p'(y|x_i) = (1-\varepsilon)p(y|x_i) + \frac{\varepsilon}{10} \tag{29}$$

$$= \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{10} & \text{if } y = y_i \\ \frac{\varepsilon}{10} & \text{otherwise} \end{cases} \tag{30}$$

    and the loss function,

$$L' = \sum_{i=1}^{n} \Big\{ (1-\varepsilon)\Big[-\sum_{y=1}^{10} p(y|x_i) \log q_\theta(y|x_i)\Big] + \varepsilon\Big[-\sum_{y=1}^{10} \frac{1}{10} \log q_\theta(y|x_i)\Big] \Big\} \tag{31}$$

Therefore, for u as a the uniform distribution, $H(u, p)$ is a measure of how dissimilar the predicted distribution p is to uniform.

For each example in the training dataset, the loss function has the contribution of the cross entropy between p and q_θ and the cross entropy between u and q_θ. If the model predicts the distribution confidently, $H_i(p, q_\theta)$ will approach to zero, but $H_i(u, q_\theta)$ will increase and "regularize" the model. Therefore, the label smoothing deals with the problem of overfitting and prevents the model from predicting too confidently.

Reference:
`https://leimao.github.io/blog/Label-Smoothing/` - Acessed on 09.01.2021
`https://arxiv.org/pdf/1512.00567.pdf` - Acessed on 09.01.2021

2 / 5 pts

c) Batch normalization

1)Without batch normalization

n	o_1	o_2	S_1	S_2	Loss
1	0.43	0.41	0.512	0.488	0.1697
2	0.44	0.58	0.431	0.569	0.2694
3	0.51	0.58	0.468	0.532	0.0046
4	0.6	0.57	0.513	0.487	0.0561
5	0.48	0.49	0.495	0.505	0.164

o2 values are incorrect. Did you use w1 and w3 again?

$$o_1 = x_1 * w_1 + x_2 * w_3 + b \tag{32}$$

$$S_1 = \frac{o_1}{o_1 + o_2} \quad \text{you have to apply exp to o1 and o2 here} \tag{33}$$

$$L = 1/2((\Sigma S_1 - p1) + (\Sigma S_2 - p2)) \tag{34}$$

$$TotalLoss = \frac{\Sigma_i L_i}{i} \tag{35}$$

$$= \frac{0.1697 + 0.2694 + 0.0046 + 0.0561 + 0.164}{5} \tag{36}$$

$$= 0.1328 \tag{37}$$

2) With batch normalization

$$H = \begin{pmatrix} 0.43 & 0.41 \\ 0.44 & 0.58 \\ 0.51 & 0.58 \\ 0.6 & 0.57 \\ 0.48 & 0.49 \end{pmatrix} \tag{38}$$

$$\mu = \frac{1}{m} * \Sigma_{i=1}^{m} H_i \tag{39}$$

$$= \begin{pmatrix} 0.43 & 0.41 \\ 0.44 & 0.58 \\ 0.51 & 0.58 \\ 0.6 & 0.57 \\ 0.48 & 0.49 \end{pmatrix} \tag{40}$$

$$\sigma = \sqrt{\delta + \frac{1}{m} * \Sigma_{i=1}^{m} (H - \mu)_i^2} \tag{41}$$

$$= 1e^{-4} \tag{42}$$

$$H' = \frac{H - \mu}{\sigma} \tag{43}$$

$$= 0 \tag{44}$$

This should not be 0.

Gradients missing.