CS5787: Exercises 1

FULL_CODE_FOLDER_URL

Mitchell Krieger
mak483

Meitong (Estelle) He
mh2585

1 Theory: Question 1 [12.5 pts]

a. **What is the shape of the input X?**
There are m samples in the batch with 10 features each so the input shape is $(m, 10)$

b. **What is the shape of the hidden layer's weight vector W_h, and the shape of its bias vector b_h?**
To transform an $(m, 10)$ input into a hidden layer of 50 neurons we need a shape W_h to have a shape of $(10, 50)$ so that we get a $(m, 50)$ result. After the weight matrix is applied, we then add the bias b_h to each row of $W_h X$, which means that the b_h needs to be $(1, 50)$

c. **What is the shape of the output layer's weight vector W_o, and its bias vector b_o?**
$W_h : (50, 3), b_h : (1, 3)$ using the same logic in part a, but now the hidden layer is the input and the output takes the place of the hidden layer.

d. **What is the shape of the network's output matrix Y?**
The shape is $(m, 3)$ because there are m samples and three possible classes to have outputs for.

e. **Write an equation that computes the network's output matrix Y as a function of X, W_h, b_h, W_o, and b_o?**
$Y = W_o a(W_h X + b_h) + b_o$
Where a is the ReLU activation funciton, $a(x) = \max(0, x)$.

2 Theory: Question 2 [12.5 pts]

For the first layer there are 3x3x3 parameters in each of the 100 kernels, and then we need bias parameters for each kernel as well. This totals 3x3x3x100 + 100 = 2800 parameters.

In the second layer there are 3x3x100 parameters in each of the 200 kernels, and then we need bias parameters for each kernel as well. This totals 3x3x100x200 + 200 = 180200.

In the final layer there are 3x3x200 parameters in each of the 200 kernels, and then we need bias parameters for each kernel as well. This totals 3x3x200x400 + 400 = 720400.

Adding all the layers together we get 2800 + 180200 + 720400 = 903400 total parameters.

3 Theory: Question 3 [25 pts]

a.

$$\begin{aligned}\frac{\partial f}{\partial \gamma} &= \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \gamma} \\ &= \sum_{i=i}^{m} \frac{\partial f}{\partial y_i}\widehat{x_i}\end{aligned}$$

b.

$$\begin{aligned}\frac{\partial f}{\partial \beta} &= \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \beta} \\ &= \sum_{i=i}^{m} \frac{\partial f}{\partial y_i}\end{aligned}$$

c.

$$\begin{aligned}\frac{\partial f}{\partial \widehat{x_i}} &= \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \widehat{x_i}} \\ &= \frac{\partial f}{\partial y_i}\gamma\end{aligned}$$

d.

$$\begin{aligned}\frac{\partial f}{\partial \sigma^2} &= \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \widehat{x_i}}\frac{\partial \widehat{x_i}}{\partial \sigma^2} \\ &= \sum_{i=i}^{m}\frac{\partial f}{\partial y_i}\gamma\frac{-1}{2}\frac{(x_i-\mu)}{\sqrt{(\sigma^2+\epsilon)^3}}\end{aligned}$$

e.

$$\begin{aligned}\frac{\partial f}{\partial \mu} &= \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \widehat{x_i}}\frac{\partial \widehat{x_i}}{\partial \mu} + \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \widehat{x_i}}\frac{\partial \widehat{x_i}}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial \mu} \\ &= \sum_{i=i}^{m}\frac{\partial f}{\partial y_i}\gamma\frac{-1}{\sqrt{(\sigma^2+\epsilon)}} \\ &\quad + \sum_{i=i}^{m}\frac{\partial f}{\partial y_i}\gamma\frac{-1}{2}\frac{(x_i-\mu)}{\sqrt{(\sigma^2+\epsilon)^3}}\frac{-2}{m}\sum_{i=1}^{m}(x_i-\mu)\end{aligned}$$

f.

$$\begin{aligned}\frac{\partial f}{\partial x_i} &= \frac{\partial f}{\partial \widehat{x_i}}\frac{\partial \widehat{x_i}}{\partial x_i} + \frac{\partial f}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial x_i} + \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial x_i} \\ \frac{\partial f}{\partial x_i} &= \frac{\partial f}{\partial y_i}\gamma\frac{1}{\sqrt{(\sigma^2+\epsilon)^3}} \\ &\quad + \frac{\partial f}{\partial y_i}\gamma\frac{-1}{2}\frac{(x_i-\mu)}{\sqrt{(\sigma^2+\epsilon)^3}}\frac{2}{m}(x_i-\mu) \\ &\quad + \frac{1}{m}\left(\frac{\partial f}{\partial y_i}\gamma\frac{-1}{\sqrt{(\sigma^2+\epsilon)^3}} + \frac{\partial f}{\partial y_i}\gamma\frac{-1}{2}\frac{(x_i-\mu)}{\sqrt{(\sigma^2+\epsilon)^3}}\frac{-2}{m}\sum_{i=1}^{m}(x_i-\mu)\right)\end{aligned}$$

4 Practical [50 pts]

In this experiment, our goal is to experiment with three different regularization techniques for neural networks and compare their performance results. We used the Lenet5 CNN architecture (LeCun 1998) as a baseline model trained on the FashionMNIST dataset. LeNet5 is composed of two convolutional layers with pooling after each convolution followed by three fully connected layers. Note that we made a few small alterations to the original LeNet5 architecture by switching out the sigmoid activation function for ReLU and used max pooling instead of average pooling. This is because after a little experimenting we found that our models generally performed better on validation accuracy with these changes, they are less computationally expensive, and they converged faster. In addtion, in the 25 years since LeNet was published, there has been research that has shown that these are these changes are in general help performance