Computer Vision

Subhasish Basak (MDS201803)

Home_Work Assignment : Lectures 19-21

April 3, 2020

1 Q.(1)

Consider the given feed-forward algorithm for MNIST with a batch size of 1. If we are to examine the bias variables before and after the first training example is processed, describe how the bias variables may change as a result.

Here we have considered a simple Neural Network framework for MNIST handwritten digit recognition. It takes as input a 28×28 image, i.e. the input vector is of length 784×1 and produces an output of size 10×1. It uses the *Cross Entropy* loss function , *Sigmoid* activation function and *Softmax* in the output layer (in order to return the class probabilities).

We compute the feed forward network using the following equations :

$$X(\phi, x) = -\log p(a) \tag{1}$$

$$p(a) = \sigma_a(l) \tag{2}$$

$$l_j = b_j + x * w_j \tag{3}$$

where ϕ is the output and $X(\phi, x)$ is the loss corresponding the input x. σ_a is the *Softmax* function.

To observe the change in the bias variables we note down the *backpropagation* update equation for the bias parameter (b_j).

$$\frac{\delta X(\phi)}{\delta b_j} = \frac{\delta l_j}{\delta b_j} \frac{\delta X(\phi)}{\delta l_j} \tag{4}$$

Now,

$$\frac{\delta l_j}{\delta b_j} = \frac{\delta}{\delta b_j}(b_j + \sum_i x_i w_{j,i}) = 1 \tag{5}$$

$$\frac{\delta X(\phi)}{\delta l_j} = \frac{\delta p_a}{\delta l_j} \frac{\delta X(\phi)}{\delta p_a} = \frac{\delta p_a}{\delta l_j} \frac{\delta}{\delta p_a}(-\log p_a) = \frac{\delta p_a}{\delta l_j} \frac{(-1)}{p_a} \tag{6}$$

and,

$$\frac{\delta p_a}{\delta l_j} = \frac{\delta \sigma_a(l)}{\delta l_j} = \begin{cases} (1 - p_j)p_a & a = j \\ -p_j p_a & a \neq j \end{cases} \tag{7}$$

Hence the update rule (with learning rate η) for the bias parameter is as follows,

$$\Delta b_j = \eta \begin{cases} (1 - p_j) & a = j \\ -p_j & a \neq j \end{cases} \tag{8}$$

Hence, with this update equation we can examine the changes in the bias parameter.

2 Q.(2)

We simplify our MNIST computation by assuming our "image" has two binary-valued pixels, 0 and 1. Further, we assume that there are no bias parameters and that we are performing a binary classification problem.

2.1 (a)

The pixel values are $[0, 1]$, the weight matrix is $\begin{pmatrix} 0.2 & -0.3 \\ -0.1 & 0.4 \end{pmatrix}$ We obtain the logit by multiplying the pixel vector with the weight matrix. The logit is given by $l = [-0.1, 0.4]$.
Now to obtain the probabilities we apply ***softmax*** on the logit and get p = [0.37754067, 0.62245933].

2.2 (b)

With the learning rate $\eta = 0.1$ and true output as 1, the loss is given by,

$$\mathbf{L} = -\log(p) = 0.4740 \tag{9}$$

2.3 (c)

We have the derivative of the loss function w.r.t the weights as follows,

$$\frac{\delta X(\phi)}{\delta w_{j,i}} = \frac{\delta}{\delta w_{j,i}}(b_j + (w_{j,1}x_1 + \cdots + w_{j,i}x_i + \cdots)) = x_i \tag{10}$$

Now, $\frac{\delta X(\phi)}{\delta w_{0,0}} = x_0 = 0$. Thus $\Delta w_{0,0} = 0$.

3 Q.(3)

Assume a ConvNet with an initial convolution layer and a pooling layer that follows. We are required to find the size of the resulting feature map.
In general we have the formula for the size of the feature map as,

$$w_{out} = \left\lfloor \frac{w_{in} - k}{s} \right\rfloor + 1 \tag{11}$$

where, w_{in} is the size of the input, w_{out} is the size of the out put. k is the size of the kernel, s is the stride size.

3.1 (a)

Before pooling we get the output size using formula (11) as $92 \times 92 \times 3$.
After pooling with a 4×4 layer, we basically substitute a 4×4 grid of pixels by a single pixel which contains the maximum among all the pixels in that grid. Hence, after pooling the output size becomes $23 \times 23 \times 3$.

3.2 (b)

Before pooling we get the output size using formula (11) as $32 \times 32 \times 3$. By similar argument as above we can say that after pooling with a 8×8 layer the output size becomes $4 \times 4 \times 3$.