Parameter Calculation of Neural Network

CSE 4237
Soft Computing

Model Architecture

```
NeuralNetwork(
  (cnn_layer_1): Conv2d(1, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (cnn_layer_2): Conv2d(16, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (flatten): Flatten(start_dim=1, end_dim=-1)
  (maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (linear_layer_1): Linear(in_features=1568, out_features=512, bias=True)
  (linear_layer_2): Linear(in_features=512, out_features=128, bias=True)
  (linear_layer_3): Linear(in_features=128, out_features=10, bias=True)
  (relu): ReLU()
  (sigmoid): Sigmoid()
  (dropout): Dropout(p=0.2, inplace=False)
)
```

```
def forward(self, x):
        x = self.cnn_layer_1(x)
        x = self.dropout(x)
        x = self.relu(x)
        x = self.maxpool(x)
        x = self.cnn_layer_2(x)
        x = self.dropout(x)
        x = self.relu(x)
        x = self.maxpool(x)
        x = self.flatten(x)
        x = self.linear_layer_1(x)
        x = self.dropout(x)
        x = self.relu(x)
        x = self.linear_layer_2(x)
        x = self.dropout(x)
        x = self.relu(x)

        x = self.linear_layer_3(x)
        return x
```

Number of parameters in a CONV layer would be :

((m * n * d)+1)* k)

added 1 because of the bias term for each filter.

The same expression can be written as follows:

((shape of width of the filter * shape of height of the filter * number of filters in the previous layer+1)*number of filters).

Calculation of CNN Output Shape

$$\text{Floor}\left(\frac{(W-F+2P)}{S}+1\right)$$

Model Summary

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 28, 28]	416
Dropout-2	[-1, 16, 28, 28]	0
ReLU-3	[-1, 16, 28, 28]	0
MaxPool2d-4	[-1, 16, 14, 14]	0
Conv2d-5	[-1, 32, 14, 14]	12,832
Dropout-6	[-1, 32, 14, 14]	0
ReLU-7	[-1, 32, 14, 14]	0
MaxPool2d-8	[-1, 32, 7, 7]	0
Flatten-9	[-1, 1568]	0
Linear-10	[-1, 512]	803,328
Dropout-11	[-1, 512]	0
ReLU-12	[-1, 512]	0
Linear-13	[-1, 128]	65,664
Dropout-14	[-1, 128]	0
ReLU-15	[-1, 128]	0
Linear-16	[-1, 10]	1,290

Model Summary

```
==================================================================
Total params: 883,530
Trainable params: 883,530
Non-trainable params: 0
------------------------------------------------------------------
```

Example

Calculate the number of parameters for the following architecture for the input shape of 1024.

input→FCwB(256)→Dropout(0.1)→FC(128)→tanh→Maxpool(3)→FC(3)→output

Here, FCwB(x) denotes a Fully Connected Layer of x neurons with Bias,

FC(x) denotes a Fully Connected Layer of x neurons without Bias,

Dropout (y) denotes a Dropout Layer that drops input at a rate of y, and

Maxpool(x) denotes a flat Maxpooling layer that has both window size and stride size of x.

Example

Layer	Output Shape	No. of Parameter
input	1024	0
FCwB	256	(1024*256+256) = 2,62,400
Dropout	256	0
FC	128	256*128 = 32,768
tanh	128	0
Maxpool	floor((128-3+2*0)/3) = 42	0
FC	3	42*3=126
Output	3	0
	Total	2,95,294

Reference

https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d#:~:text=Number%20of%20parameters%20in%20a%20CONV%20layer%20would%20be%20%3A%20((,1)*number%20of%20filters).