AI6126 Advanced Computer Vision

Chen Yongquan (G2002341D)
Nanyang Technological University

Assignment 1

Question 1

Given:

$$\text{Output Size} = \frac{N - F + 2P}{S} + 1$$

Where:

$$\begin{aligned} \mathrm{N} &= \text{Input Size} \\ \mathrm{F} &= \text{Filter Size} \\ \mathrm{P} &= \text{Padding} \\ \mathrm{S} &= \text{Stride} \end{aligned}$$

i. Conv2d-1

Input Shape = [1,32,32]

Output Shape = [6,28,28]

$$28 = \frac{32 - \mathrm{F} + 0}{1} + 1$$

F = 5

For six 1x5x5 filters with stride 1, pad 0:

$$\text{Number of Parameters} = [(1 \times 5 \times 5) + 1] \times 6 = 156$$

ii. ReLU-2

Output Shape = [6,28,28]

$$\text{Number of Parameters} = 0$$

iii. MaxPool2d-3

Output Shape = [6,14,14]

$$14 = \frac{28 - \mathrm{F} + 0}{S} + 1$$

Assuming $F \equiv S$ for max pooling layers,

$$\mathrm{F} = 2$$

$$\text{Number of Parameters} = 0$$

iv. Conv2d-4

Output Shape = [16,10,10]

$$10 = \frac{14 - \mathrm{F} + 0}{1} + 1$$

F = 5

For sixteen 6x5x5 filters with stride 1, pad 0:

$$\text{Number of Parameters} = [(6 \times 5 \times 5) + 1] \times 16 = 2416$$

v. ReLU-5

Output Shape = [16,10,10]

$$\text{Number of Parameters} = 0$$

vi. MaxPool2d-6

Output Shape = [16,5,5]

$$5 = \frac{10 - \mathrm{F} + 0}{S} + 1$$

Assuming $F \equiv S$ for max pooling layers,

$$\mathrm{F} = 2$$

$$\text{Number of Parameters} = 0$$

vii. Conv2d-7

Output Shape = [120,1,1]

$$1 = \frac{5 - \mathrm{F} + 0}{1} + 1$$

F = 5

For one hundred and twenty 16x5x5 filters with stride 1, pad 0:

$$\text{Number of Parameters} = [(16 \times 5 \times 5) + 1] \times 120 = 48120$$

viii. ReLU-8

Output Shape = [120,1,1]

$$\text{Number of Parameters} = 0$$

ix. Linear-9

Output Shape = [84]

$$\text{Number of Parameters} = (120 \times 84) + 84 = 10164$$

x. ReLU-10

Output Shape = [84]

$$\text{Number of Parameters} = 0$$

xi. Linear-11

Output Shape = [10]

$$\text{Number of Parameters} = (84 \times 10) + 10 = 850$$

xii. LogSoftmax-12

Output Shape = [10]

$$\text{Number of Parameters} = 0$$

xiii. Total Number of Parameters

$$\text{Total Number of Parameters} = 156 + 2416 + 48120 + 10164 + 850 = \ 61706$$

Question 2

```
class HelloCNN(nn.Module):
  def __init__(self):
    super(HelloCNN, self).__init__()
    self.conv1 = nn.Conv2d(1, 6, 5, 1)
    self.conv2 = nn.Conv2d(6, 16, 5, 1)
    self.conv3 = nn.Conv2d(16, 120, 5, 1)
    self.fc1 = nn.Linear(120, 84)
    self.fc2 = nn.Linear(84, 10)
    self.pool = nn.MaxPool2d(2)
    self.relu = nn.ReLU()
    self.lsm = nn.LogSoftmax(1)

  def forward(self, x):
    x = self.conv1(x) #1
    x = self.relu(x)  #2
    x = self.pool(x)  #3
    x = self.conv2(x) #4
    x = self.relu(x)  #5
    x = self.pool(x)  #6
    x = self.conv3(x) #7
    x = self.relu(x)  #8
    x = x.view(-1,self.fc1.weight.shape[1])
    x = self.fc1(x)   #9
    x = self.relu(x)  #10
    x = self.fc2(x)   #11
    x = self.lsm(x)   #12
    return x
```

Question 3

i.

Both classification and regressions problems involve mapping a given m-dimensional input x to a scalar output y.

However, in classification the range of y consists of discrete values only, where each value corresponds to a class or label. On the other hand, in regression the range of y consists of continuous real values.

Also, for classification problems, output y corresponds to the value that gives the highest posterior probability given input x, while for regression problems, output y is the estimated expected value given input x.

Geometrically, in binary linear classification problems our goal can be interpreted as trying to find the hyperplane (i.e. decision boundary) that divides the problem space into individual regions for each class, while for regression our goal is to fit a curve to the problem based on the available training data points.

ii.

Depending on how we formulate the architecture of the network, age estimation can be tackled as either a classification problem or a regression problem. For instance, our network can extract facial features from the images and attempt to classify them into appropriate discrete age groups or ages. Conversely, our network can also perform regression on the features and estimate a real value in the human age range. Finally, a fusion of both classification and regression can also be used in the model to improve the accuracy of age estimation (Jiang, Zhang, & Yang, 2018).

iii.

We also include a validation set when dividing the full set of sample data into training and testing sets as we can use the validation set for fine tuning the hyper-parameters of the model.
If we use only the training set to train the model, it may result in overfitting and yield poorer accuracy on the testing set or in the real-world.
If we use both the training set and testing set to train the model, it would positively bias our performance metric which uses the testing set and give a poor representation of its real-world performance.
Thus, we also divide out a validation set which we use for calculating the validation error. We then try to minimize and balance the validation error with the training error, by fine tuning hyper-parameters of the model, but we do not use the validation set to update the weights and biases i.e. “learn” from it. We cannot perform fine-tuning without “learning” with the testing set as the hyper-parameters still indirectly affects our model and would thus still bias our performance metric taken from it.

Question 4

$$\boldsymbol{w} \star \boldsymbol{x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} -1 & 0 & 1 & 0 & -2 & 0 & 2 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & -2 & 0 & 2 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -2 & 0 & 2 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -2 & 0 & 2 & 0 & -1 & 0 & 1 \end{pmatrix}$$

$$v(\mathbf{x}) = (10 \quad 10 \quad 0 \quad 0 \quad 10 \quad 10 \quad 0 \quad 0 \quad 10 \quad 10 \quad 0 \quad 0 \quad 10 \quad 10 \quad 0 \quad 0)^{\mathrm{T}}$$

$$\mathbf{W}v(\mathbf{x}) = \begin{pmatrix} (-1 \times 10) + (-2 \times 10) + (-1 \times 10) \\ (-1 \times 10) + (-2 \times 10) + (-1 \times 10) \\ (-1 \times 10) + (-2 \times 10) + (-1 \times 10) \\ (-1 \times 10) + (-2 \times 10) + (-1 \times 10) \end{pmatrix}$$

$$= \begin{pmatrix} -40 \\ -40 \\ -40 \\ -40 \end{pmatrix}$$

After reshaping,

$$\boldsymbol{w} \star \boldsymbol{x} = \begin{pmatrix} -40 & -40 \\ -40 & -40 \end{pmatrix}$$

Question 5

$$\text{Mean Absolute Error} = \sum_{i=1}^{n} |y_{True} - y_{Predicted}|$$

$$\text{Mean Squared Error} = \sum_{i=1}^{n} (y_{True} - y_{Predicted})^2$$

We might prefer to minimize L1 loss (MAE) instead of L2 loss (MSE) when we know our sample data contains many outlier data points that we want to "ignore" and not affect our weights training too much. For instance, given an outlier input x which gives an output y of 10 with a target of 1, L1 loss is only 9 however L2 loss is 81. When minimizing L2 loss, the much higher loss due to outliers are then backpropagated through the model and the weights and biases are corrected to account for outliers. Consequently, the accuracy of the model on regular data points deteriorates. Using L1 loss makes training less sensitive to these outliers and thus provides more robustness to our model.

References

Jiang, F., Zhang, Y., & Yang, G. (2018). Facial Age Estimation Method Based on Fusion Classification and Regression Model. *MATEC Web of Conferences*, *232*, p. 02021. doi:10.1051/matecconf/201823202021