

# AI6126: Homework 1

## Question 1:

1. Conv2d:
  - Input:  $1 * 32 * 32$ , output  $6 * 28 * 28$
  - `nn.Conv2d(1, 6, (5, 5), stride=(1, 1), bias=True)`
  - 6 kernels, each size  $5 * 5$ , weights:  $6 * 5 * 5 = 150$ , bias = 6, total parameters: 156
2. ReLu: 0 parameters
3. MaxPool2d:
  - input  $6 * 28 * 28$ , output:  $6 * 14 * 14$
  - `nn.MaxPool2d((15, 15), stride = (1, 1))`
  - 0 parameters
4. Conv2d:
  - input:  $6 * 14 * 14$ , output:  $16 * 10 * 10$
  - `nn.Conv2d(6, 16, (5, 5), stride=(1, 1), bias = True)`
  - $6 * 16$  kernels, each size  $5 * 5$ , weight =  $16 * 6 * 5 * 5 = 2400$ , bias = 16, total parameters: 2416
5. Relu: 0 parameters
6. MaxPool2d:
  - input:  $16 * 10 * 10$ , output:  $16 * 5 * 5$
  - `nn.MaxPool2d((6, 6), stride= (1, 1))`
  - 0 parameters
7. Conv2d:
  - input:  $16 * 5 * 5$ , output:  $120 * 1 * 1$
  - `nn.Conv2d(16, 120, (5, 5), stride=(1, 1), bias = True)`
  - $16 * 120$  kernels, each size  $5 * 5$ , weight =  $16 * 120 * 5 * 5$ , bias= 120, total parameters: 48120
8. Relu: 0 parameters
9. Linear:
  - input  $120 * 1 * 1$ , output 84
  - `nn.Linear(120, 84, bias= True)`
  - weight =  $120 * 84 = 10080$ , bias = 84, total parameters: 10164
10. Relu: 0 parameters
11. Linear:
  - input 84, output 10
  - `nn.Linear(84, 10, bias= True)`
  - weight =  $84 * 10 = 840$ , bias = 10, total parameters: 850
12. LogSoftmax: 0 parameters Total Parameters =  $156 + 2416 + 48120 + 10164 + 850 = 61706$

## Question 2:

```
import torch.nn as nn
class HelloCNN(nn.Module):
    def __init__(self):
        super(HelloCNN, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(1, 6, (5, 5), stride=(1, 1), bias=True),
            nn.ReLU(),
            nn.MaxPool2d((15, 15), stride=(1, 1)),
            nn.Conv2d(6, 16, (5, 5), stride=(1, 1), bias=True),
            nn.ReLU(),
            nn.MaxPool2d((6, 6), stride= (1, 1)),
            nn.Conv2d(16, 120, (5, 5), stride=(1, 1), bias=True),
```

```

        nn.ReLU(),
    )
    self.linear = nn.Sequential(
        nn.Linear(120, 84, bias=True),
        nn.ReLU(),
        nn.Linear(84, 10, bias=True),
        nn.LogSoftmax(dim=1),
    )

    def forward(self, x):
        return self.linear(self.conv(x).squeeze())

```

### Question3:

#### 1. Explain the difference between regression and classification:

**Classification** is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes i.e. discrete values. **Regression** is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values.

#### 2. You need to train a neural network that predicts the age of a person. Is this a regression or classification problem: The problem of predicting age can be both Regression or classification. But Regression is better

#### 3. Why do we need a validation set? Validation dataset is a dataset of examples used to tune the hyperparameters (i.e. the architecture) of a model to avoid overfitting

### Question 4

Flatten the input into vector by the row:

$$input = \begin{pmatrix} 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 0 \\ 0 \\ 10 \\ 10 \\ 0 \\ 0 \\ 10 \\ 10 \\ 0 \\ 0 \\ 10 \\ 10 \\ 0 \\ 0 \end{pmatrix}$$

unroll the Kernel:

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

### Question 5

Why might we prefer to minimize the sum of absolute residuals (L1 loss) instead of the residual sum of squares for some data sets (L2 loss)?

L1 loss that is less sensitive to outliers than the L2 loss, which means the L1 loss is more robust on the data with more outliers