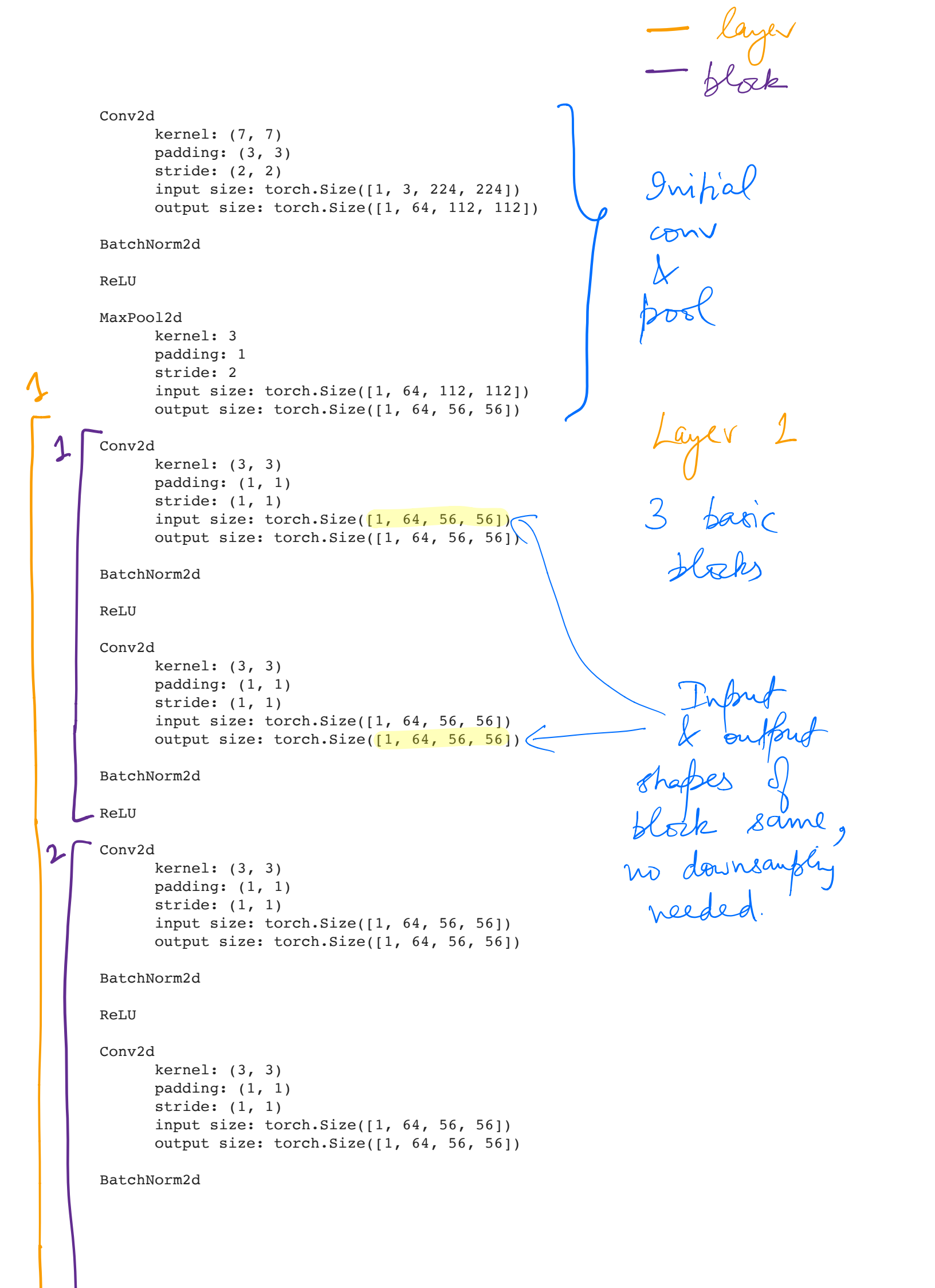
layer
block
Conv2d
kernel: (7, 7)
padding: (3, 3)
stride: (2, 2)
input size: torch.Size([1, 3, 224, 224])
output size: torch.Size([1, 64, 112, 112])
BatchNorm2d
ReLU
MaxPool2d
kernel: 3
padding: 1
stride: 2
input size: torch.Size([1, 64, 112, 112])
output size: torch.Size([1, 64, 56, 56])
Initial conv & pool
1
1
Conv2d
kernel: (3, 3)
padding: (1, 1)
stride: (1, 1)
input size: torch.Size([1, 64, 56, 56])
output size: torch.Size([1, 64, 56, 56])
BatchNorm2d
ReLU
Conv2d
kernel: (3, 3)
padding: (1, 1)
stride: (1, 1)
input size: torch.Size([1, 64, 56, 56])
output size: torch.Size([1, 64, 56, 56])
BatchNorm2d
ReLU
Layer 1
3 basic blocks
Input & output shapes of block same, no downsampling needed.
2
Conv2d
kernel: (3, 3)
padding: (1, 1)
stride: (1, 1)
input size: torch.Size([1, 64, 56, 56])
output size: torch.Size([1, 64, 56, 56])
BatchNorm2d
ReLU
Conv2d
kernel: (3, 3)
padding: (1, 1)
stride: (1, 1)
input size: torch.Size([1, 64, 56, 56])
output size: torch.Size([1, 64, 56, 56])
BatchNorm2d

```
ReLU
Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 64, 56, 56])
      output size: torch.Size([1, 64, 56, 56])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 64, 56, 56])
      output size: torch.Size([1, 64, 56, 56])

BatchNorm2d

ReLU

            ------ Layer boundary ------

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (2, 2)
      input size: torch.Size([1, 64, 56, 56])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

Conv2d
      kernel: (1, 1)
      padding: (0, 0)
      stride: (2, 2)
      input size: torch.Size([1, 64, 56, 56])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
```

Layer 2

4 basic blocks

Shapes not equal for shortcut connection

downsampling

```
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 128, 28, 28])
```

```
BatchNorm2d

ReLU

              ------ Layer boundary ------

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (2, 2)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

Conv2d
      kernel: (1, 1)
      padding: (0, 0)
      stride: (2, 2)
      input size: torch.Size([1, 128, 28, 28])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU
```

3

1

2

Layer 3

6 basic blocks

Shapes not equal for shortcut connection

downsampling

3

```
Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU
```

4

```
Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU
```

5

```
Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
```

```
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 256, 14, 14])

BatchNorm2d

ReLU

             ------ Layer boundary ------

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (2, 2)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

Conv2d
      kernel: (1, 1)
      padding: (0, 0)
      stride: (2, 2)
      input size: torch.Size([1, 256, 14, 14])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d
```

layer 4
3 basic blocks

```
ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

ReLU
```

2

```
Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

ReLU

Conv2d
      kernel: (3, 3)
      padding: (1, 1)
      stride: (1, 1)
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 7, 7])

BatchNorm2d

ReLU
```

3

```
             ------ Layer boundary ------

AvgPool2d
      kernel: 7
      padding: 0
      stride: 1
      input size: torch.Size([1, 512, 7, 7])
      output size: torch.Size([1, 512, 1, 1])
```

Final avg. pooling

```
Linear
      input size: torch.Size([1, 512])
      output size: torch.Size([1, 1000])
```

Fully connected