

— 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

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  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: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 64, 56, 56])
  output size: torch.Size([1, 256, 56, 56])

BatchNorm2d
```

Layer 1

3 bottleneck
blocks

shapes not
equal
for shortcut
connection

downsampling

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 64, 56, 56])
  output size: torch.Size([1, 256, 56, 56])

BatchNorm2d

ReLU
```

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 256, 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: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 64, 56, 56])
  output size: torch.Size([1, 256, 56, 56])
```

```
BatchNorm2d
```

```
ReLU
```

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 256, 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: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
```

```
input size: torch.Size([1, 64, 56, 56])
output size: torch.Size([1, 256, 56, 56])
```

BatchNorm2d

ReLU

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

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 256, 56, 56])
output size: torch.Size([1, 128, 56, 56])
```

BatchNorm2d

ReLU

Conv2d

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

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 128, 28, 28])
output size: torch.Size([1, 512, 28, 28])
```

BatchNorm2d

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (2, 2)
input size: torch.Size([1, 256, 56, 56])
output size: torch.Size([1, 512, 28, 28])
```

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 28, 28])
output size: torch.Size([1, 128, 28, 28])
```

BatchNorm2d

layer 2

4 bottleneck blocks

shapes not equal for shortcut

downsampling

2

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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 128, 28, 28])
output size: torch.Size([1, 512, 28, 28])

BatchNorm2d

ReLU

Conv2d

kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 128, 28, 28])
output size: torch.Size([1, 512, 28, 28])

BatchNorm2d

ReLU

Conv2d

kernel: (1, 1)

```
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 128, 28, 28])
output size: torch.Size([1, 512, 28, 28])
```

BatchNorm2d

ReLU

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

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 28, 28])
output size: torch.Size([1, 256, 28, 28])
```

BatchNorm2d

ReLU

Conv2d

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

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
```

layer 3

6 bottleneck
blocks

1

3

```
input size: torch.Size([1, 256, 14, 14])
output size: torch.Size([1, 1024, 14, 14])
```

BatchNorm2d

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (2, 2)
input size: torch.Size([1, 512, 28, 28])
output size: torch.Size([1, 1024, 14, 14])
```

downsampling

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 1024, 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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 256, 14, 14])
output size: torch.Size([1, 1024, 14, 14])
```

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 1024, 14, 14])
output size: torch.Size([1, 256, 14, 14])
```

BatchNorm2d

ReLU

2

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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 256, 14, 14])
output size: torch.Size([1, 1024, 14, 14])

BatchNorm2d

ReLU

4 Conv2d
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 1024, 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: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 256, 14, 14])
output size: torch.Size([1, 1024, 14, 14])

BatchNorm2d

ReLU

5 Conv2d
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 1024, 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: (1, 1)  
padding: (0, 0)  
stride: (1, 1)  
input size: torch.Size([1, 256, 14, 14])  
output size: torch.Size([1, 1024, 14, 14])
```

```
BatchNorm2d
```

```
ReLU
```

6

```
Conv2d
```

```
kernel: (1, 1)  
padding: (0, 0)  
stride: (1, 1)  
input size: torch.Size([1, 1024, 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: (1, 1)  
padding: (0, 0)  
stride: (1, 1)  
input size: torch.Size([1, 256, 14, 14])  
output size: torch.Size([1, 1024, 14, 14])
```

```
BatchNorm2d
```

```
ReLU
```


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

layer 4

3 bottleneck
blocks

4

1

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 1024, 14, 14])
  output size: torch.Size([1, 512, 14, 14])
```

BatchNorm2d

ReLU

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

BatchNorm2d

ReLU

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 512, 7, 7])
  output size: torch.Size([1, 2048, 7, 7])
```

BatchNorm2d

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (2, 2)
  input size: torch.Size([1, 1024, 14, 14])
  output size: torch.Size([1, 2048, 7, 7])
```

downsampling



BatchNorm2d

ReLU

2

```
Conv2d
  kernel: (1, 1)
  padding: (0, 0)
  stride: (1, 1)
  input size: torch.Size([1, 2048, 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

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 7, 7])
output size: torch.Size([1, 2048, 7, 7])
```

BatchNorm2d

ReLU

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 2048, 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

Conv2d

```
kernel: (1, 1)
padding: (0, 0)
stride: (1, 1)
input size: torch.Size([1, 512, 7, 7])
output size: torch.Size([1, 2048, 7, 7])
```

BatchNorm2d

ReLU

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

AvgPool2d

```
kernel: 7
padding: 0
stride: 1
input size: torch.Size([1, 2048, 7, 7])
```

Final
average
pooling

output size: torch.Size([1, 2048, 1, 1])

Linear

input size: torch.Size([1, 2048])

output size: torch.Size([1, 1000])



Fully
connected