PyTorch

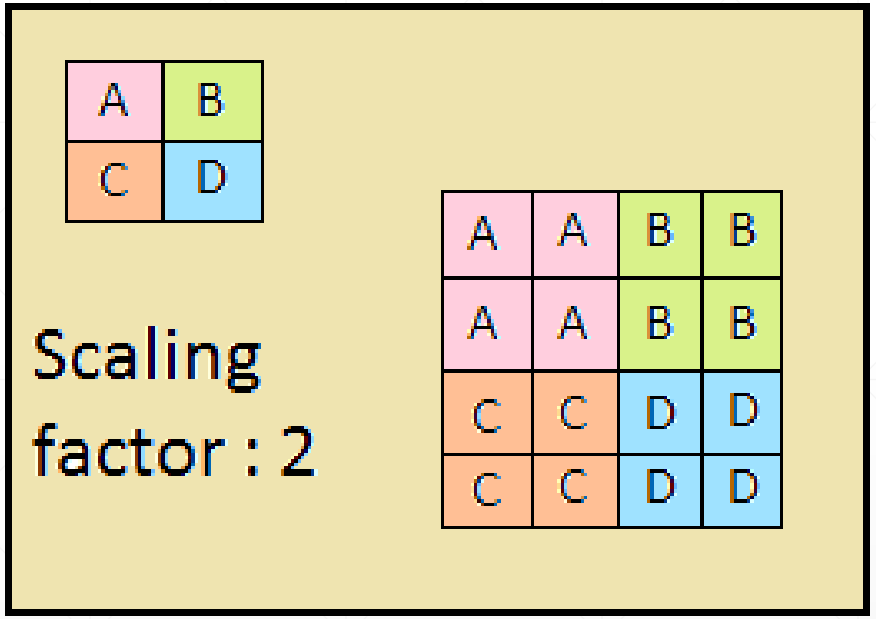
Down/up sample

主讲人：龙良曲

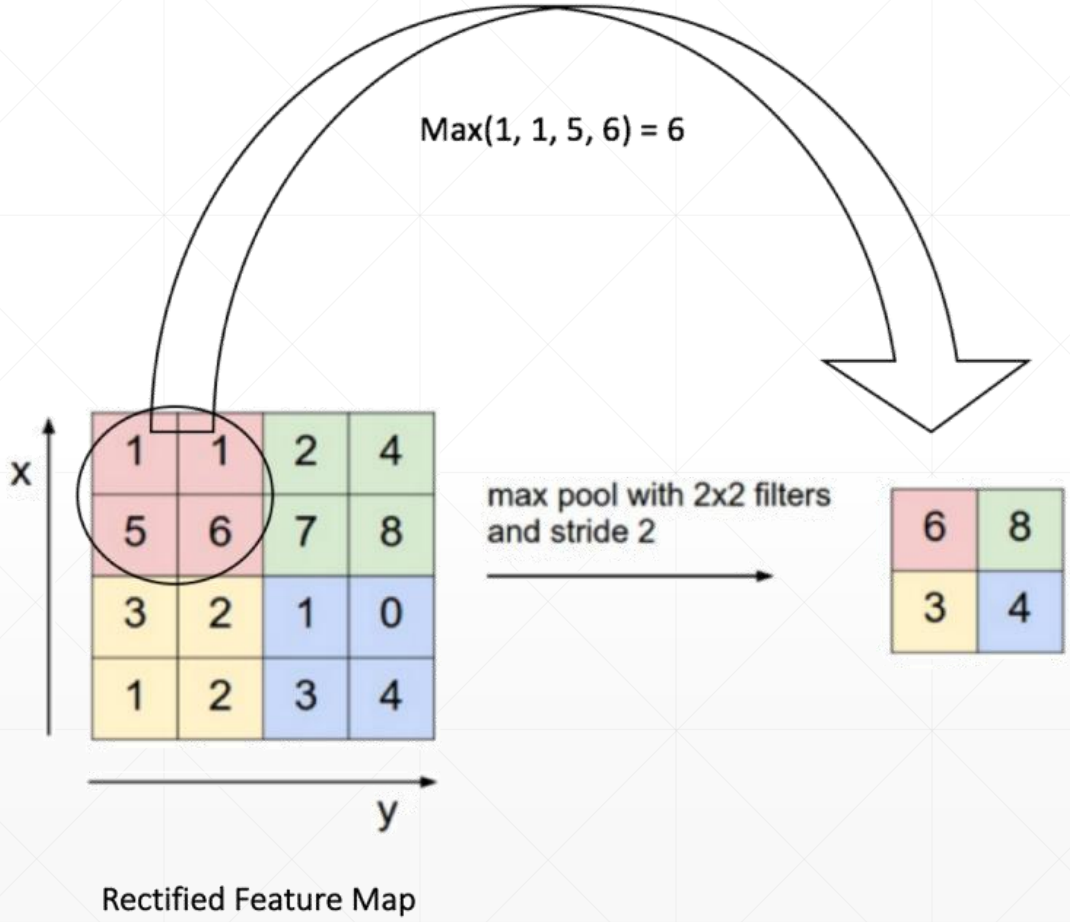
Outline

- Pooling
- upsample
- ReLU

Downsample

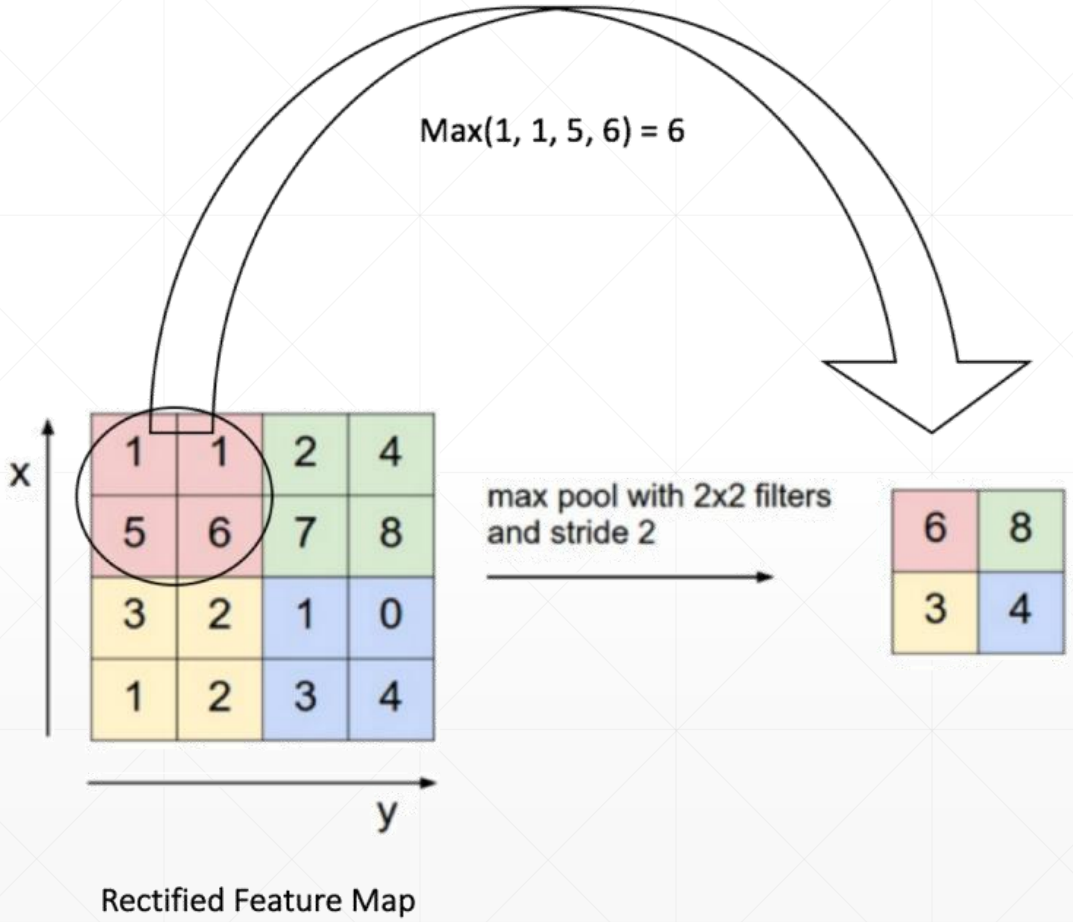


Max pooling


Max(1, 1, 5, 6) = 6
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4
Rectified Feature Map

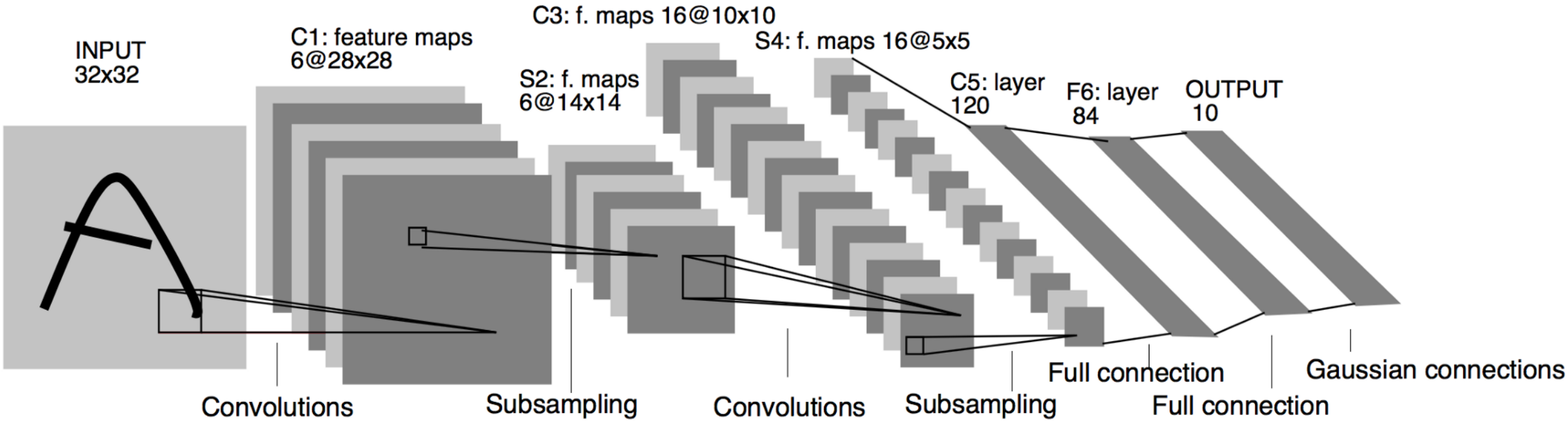
Avg pooling

- ?


Max(1, 1, 5, 6) = 6
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4
Rectified Feature Map

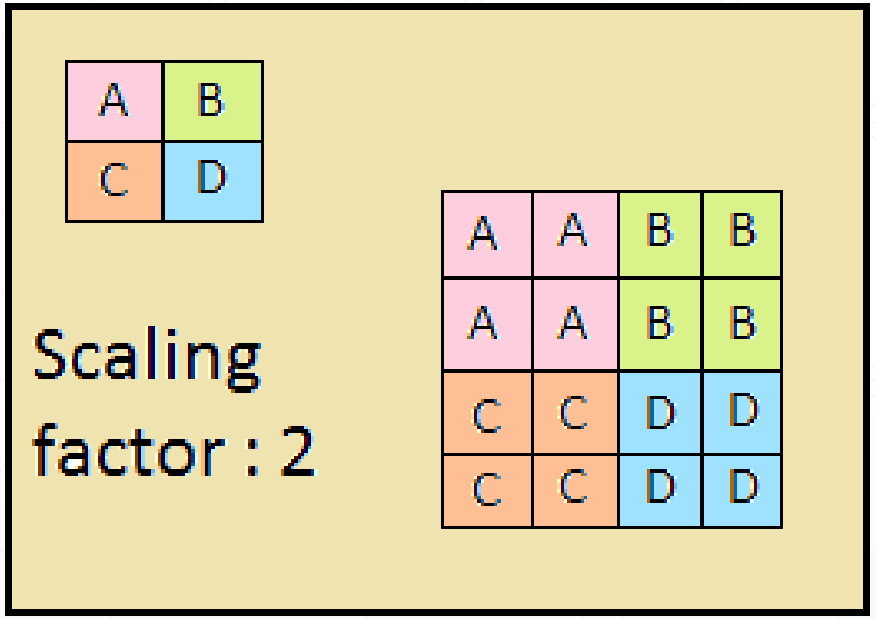
Pooling

- reduce size



```
In [33]: x=out
Out[31]: torch.Size([1, 16, 14, 14])

In [32]: layer=nn.MaxPool2d(2,stride=2)

In [34]: out=layer(x)
Out[35]: torch.Size([1, 16, 7, 7])

In [36]: out=F.avg_pool2d(x,2,stride=2)
Out[37]: torch.Size([1, 16, 7, 7])
```

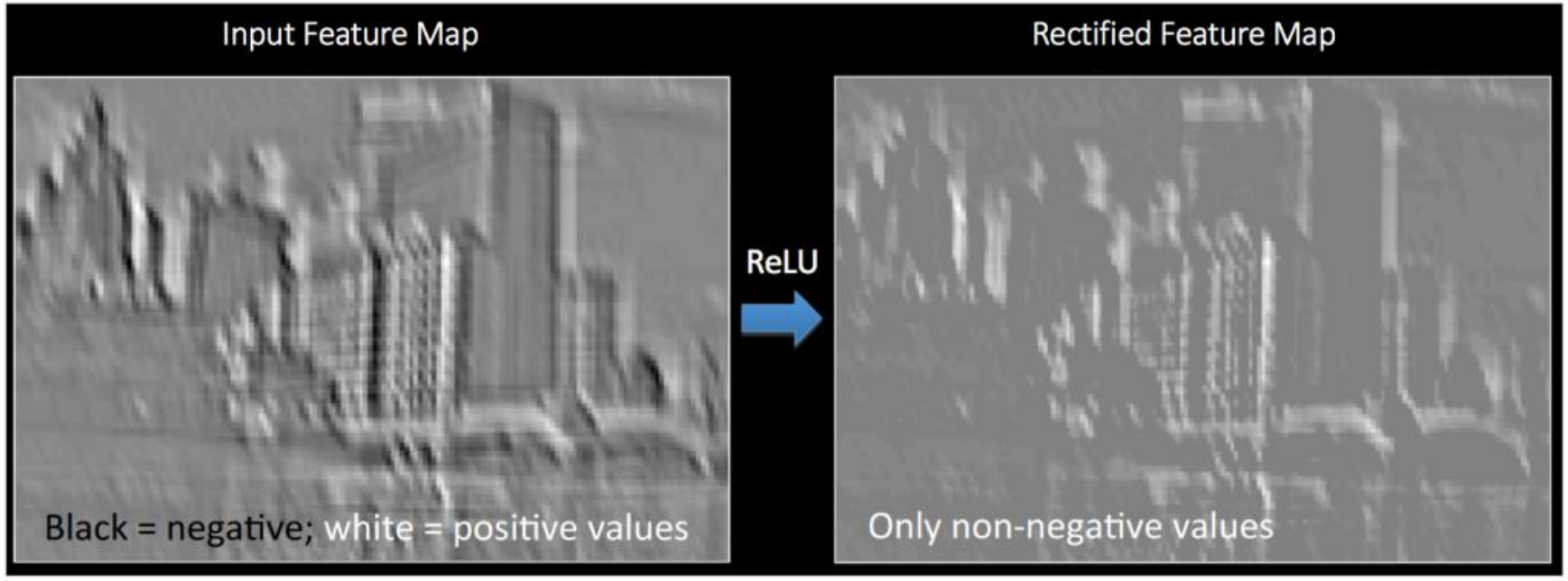
upsample



F.interpolate

```
In [38]: x=out

In [39]: out=F.interpolate(x,scale_factor=2,mode='nearest')

In [40]: out.shape
Out[40]: torch.Size([1, 16, 14, 14])

In [41]: out=F.interpolate(x,scale_factor=3,mode='nearest')

In [42]: out.shape
Out[42]: torch.Size([1, 16, 21, 21])
```

ReLU


Input Feature Map
Rectified Feature Map
ReLU
Black = negative; white = positive values
Only non-negative values

```
In [43]: x.shape
Out[43]: torch.Size([1, 16, 7, 7])

In [44]: layer=nn.ReLU(inplace=True)

In [45]: out=layer(x)
In [46]: out.shape
Out[46]: torch.Size([1, 16, 7, 7])

In [47]: out=F.relu(x)
In [48]: out.shape
Out[48]: torch.Size([1, 16, 7, 7])
```

下一课时

Batch-Norm

Thank You.