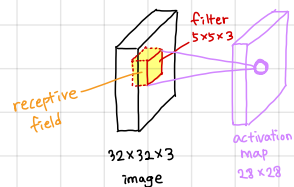


Convolutional Neural Networks

1. Convolution Layer : preserves spatial information



⇒ 여러 개 filter 사용하면? 여러 개의 activation map

e.g. 10개의 5x5 filter ⇒ 10개의 activation maps. 각 map은 28x28 사이즈

⇒ 이 필터의 receptive field : 5x5

⇒ 이런 conv layer 여러 개 쌓으면?

초반 layer는 low-level features를,

후반 layer는 high-level features를 catch

그리고 뒤로 갈수록 receptive field가 커짐.

⇒ activation map size는?

$$H \times W \text{ 사이즈} \xrightarrow[\text{padding}]{\begin{matrix} F \times F \\ \text{stride} \\ P \end{matrix} \text{ filter}} \begin{matrix} (H - F + 2P) / \text{stride} + 1 \\ (W - F + 2P) / \text{stride} + 1 \end{matrix}$$

e.g. 7x9 input, 3x3 filter, stride 2, pad 1

$$\begin{aligned} (7 - 3 + 2) / 2 + 1 &= 4 \\ (9 - 3 + 2) / 2 + 1 &= 5 \end{aligned} \quad \Rightarrow 4 \times 5 \text{ activation map size}$$

⇒ parameter 개수는?

e.g. 32x32x3 input에 10개의 5x5 filter 적용

→ 한 필터 당 (5x5x3) + 1 = 76개의 parameter

총 76 x 10 = 760개

⇒ 여러 개 필터 사용 시 적용 후 size?

e.g. 32x32x3에 16개의 5x5 filter 사용

$$\begin{aligned} \text{height: } (32 - 5) / 1 + 1 &= 28 \\ \text{width: } (32 - 5) / 1 + 1 &= 28 \\ \text{depth: } 16 \end{aligned} \quad \Rightarrow 28 \times 28 \times 16$$

2. Pooling Layer : makes the representations smaller and more manageable ; 각 activation map에 적용

e.g. 224 x 224 x 64 input $\xrightarrow[\text{(pooling)}]{\text{downsampling}}$ 112 x 112 x 64.

⇒ pooling 후 size?

e.g. H x W x D input, pooling size F, stride S $\Rightarrow ((H - F) / S + 1) \times ((W - F) / S + 1) \times D$

⇒ parameter 수는? 0개

⇒ padding? 거의 안함

3. Fully Connected Layer : activation maps를 최종 classification score로 연결하는 층