CNN

注：该卷积和数学的卷积有区别实际上是数学的互相关，不翻转的卷积

数学：

$$s(i,j)=(X*W)(i,j)=\sum_m\sum_n x(i-m,j-n)\,w(m,n)$$

会旋转180°后再相乘求和

实际CNN：

$$s(i,j)=(X*W)(i,j)=\sum_m\sum_n x(i+m,j+n)\,w(m,n)$$

当输入是多维的时候，卷积核也是多维，相乘求和再对最后一维求和。

一般使用 $ReLu(x)=\max(0,x)$。

池化 $\begin{cases}\text{max}\\ \text{average}\end{cases}$

前向传播：

$$a^2=\sigma(z^2)=\sigma(a^1*W^2+b^2)$$

1) 卷积核一般不止一个，k个，那么下一层的输入就是k个

2) Padding：更好的识别边缘

反向传播：

1. 已知池化层的 δ^l，推导上一层 δ^{l-1}

前向传播过程中，用max，average池化了

采用upsample把采样后的误差还原

~~max~~：先把 δ^l 的所有子矩阵矩阵大小还原成池化之前的大小。

①max：把 δ^l 所有子矩阵的各个池化局域的值放在之前做前向传播得最大值的位置

②average：把 δ^l 所有子矩阵各个池化局域的值取平均后放在还原后的子矩阵位置

$$\begin{pmatrix}2&8\\4&6\end{pmatrix}\to\begin{pmatrix}0&0&0&0\\0&2&8&0\\0&4&6&0\\0&0&0&0\end{pmatrix}\xrightarrow{\text{max}}\begin{pmatrix}2&0&0&0\\0&0&0&8\\0&4&0&0\\0&0&6&0\end{pmatrix}$$

$$\begin{pmatrix}0&0&0&0\\0&2&8&0\\0&4&6&0\\0&0&0&0\end{pmatrix}\xrightarrow{\text{average}}\begin{pmatrix}0.5&0.5&2&2\\0.5&0.5&2&2\\1&1&1.5&1.5\\1&1&1.5&1.5\end{pmatrix}$$

$$\delta^{l-1}=\left(\frac{\partial a_k^{l-1}}{\partial z_k^{l-1}}\right)^T\frac{\partial J}{\partial a_k^{l-1}}=upsample(\delta_k^l)\odot\sigma'(z_k^{l-1})$$

upsample完成池化误差矩阵放大与误差重新分配

2. 已知卷积层的 δ^l 推导上一层 δ^{l-1}

$$\delta^l=\left(\frac{\partial z^{l+1}}{\partial z^l}\right)\delta^{l+1}$$

$$z^l=a^{l-1}*W^l+b^l=\sigma(z^{l-1})*W^l+b^l$$

$$\delta^{l-1}=\left(\frac{\partial z^l}{\partial z^{l-1}}\right)^T\delta^l=\delta^l*rot180(W^l)\odot\sigma'(z^{l-1})$$

卷积核被旋转了180°，上下翻转，左右翻转

3. 已知卷积层的 δ^l，推导该层的W，b梯度。

池化层无W，b

卷积层有W，b，$z^l=a^{l-1}*W+b$

$$\frac{\partial J}{\partial W^l}=a^{l-1}*\delta^l$$ 卷积核没有反转，层内求导

$$\frac{\partial J}{\partial b^l}=\sum_{u,v}(\delta^l)_{u,v}$$

1、局部连接，权重共享，汇聚

(1) 参数太多
(2) 局部不变性 } 全连接解决不了

2. 导数：$Y = W \otimes X$，$X \in R^{M\times N}$，$W \in R^{m\times n}$

$Y \in R^{(M-m+1)\times(N-n+1)}$　$f(Y) \in R$

$$y_{ij} = \sum_{u,v} w_{uv}\, x_{i+u-1,\,j+v-1}$$

$$\frac{\partial f(Y)}{\partial w_{uv}} = \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial y_{ij}}{\partial w_{uv}} \frac{\partial f(Y)}{\partial y_{ij}}$$

$$= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} x_{i+u-1,\,j+v-1} \frac{\partial f(Y)}{\partial y_{ij}}$$

$$= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial f(Y)}{\partial y_{ij}} x_{u+i-1,\,j+v-1}$$

$\cancel{= rot180(}$

$$\frac{\partial f(Y)}{\partial W} = \frac{\partial f(Y)}{\partial Y} \otimes X$$

同理：
$$\frac{\partial f(Y)}{\partial x_{st}} = \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} \frac{\partial y_{ij}}{\partial x_{st}} \frac{\partial f(Y)}{\partial y_{ij}}$$

$$= \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} w_{s-i+1,\,t-j+1} \frac{\partial f(Y)}{\partial y_{ij}}$$

∴ $\frac{\partial f(Y)}{\partial X} = rot180(\frac{\partial f(Y)}{\partial Y}) \tilde{\otimes} W$

CNN：

1) $z^{(l)}$：第l层净输入

$a^{(l-1)}$：第l-1活性层

$w^{(l)} \in R^m$：filter

$$z^{(l)} = w^{(l)} \otimes a^{(l-1)} + b^{(l)}$$

① 局部连接：$n^{(l)} \times n^{(l-1)} \to n^{(l)} \times m$

② 权重共享：m+1个参数与神经元个数无关，$n^{(l)} = n^{(l-1)} + m - 1$

$$Z^p = W^p \otimes X + b^p$$
$$= \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p$$

$Y^p = f(Z^p)$　　$f(\cdot) = ReLU(\cdot)$

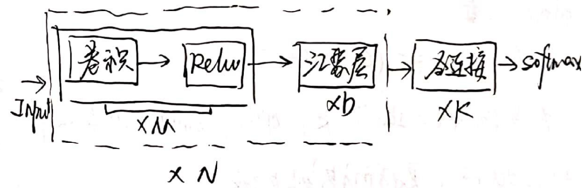
共 $P\times D\times(m\times n)+P$ 个参数

2) 汇聚层

{ max-pooling
  mean pooling

{ ① 减小神经元的数量
  ② 小的局部形态保持不变性，并拥有更大的感受野 ☆

☆ 特殊的卷积层，核函数是max或mean



{ 更小的卷积核
  更深的结构

3) 残差网络

$$h(x) = x + (h(x) - x)$$
$$= x + f(\theta, x)$$

4) Inception网络

常用卷积.

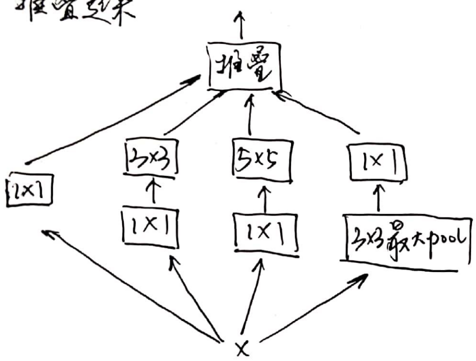
① 窄卷积: 步长 $s=1$, 两端不补0,
输出长 $n-m+1$

② 宽卷积: 步长 $s=1$, 两端补 $p=m-1$
输出长 $n+m-1$

③ 等宽卷积: 步长 $s=1$, 两端补 $p=\frac{m-1}{2}$
输出长度 n

Inception

同时使用1×1, 3×3, 5×5不同大小的卷积核, 得到的特征映射在深度上堆叠起来



1×1卷积用来减少特征映射的深度, 如果存在冗余信息, 1×1相当于先进行一次特征抽取.