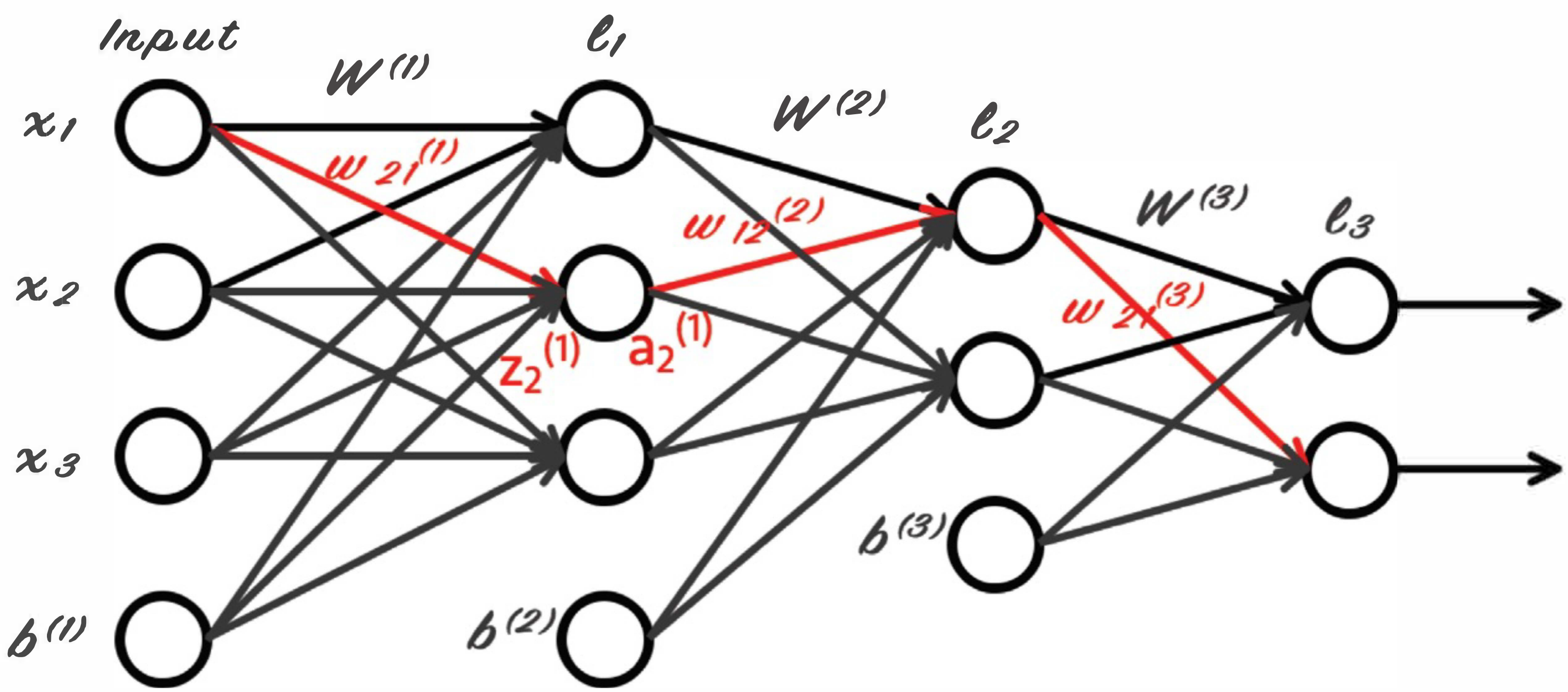
CNN示例：LeNet-5

前向传播计算



$$
\begin{aligned}
z_1^{(1)} &= w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 + b_1^{(1)} & \qquad a_1^{(1)} &= f(z_1^{(1)}) \\
z_2^{(1)} &= w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3 + b_2^{(1)} & \qquad a_2^{(1)} &= f(z_2^{(1)}) \\
z_3^{(1)} &= w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3 + b_3^{(1)} & \qquad a_3^{(1)} &= f(z_3^{(1)})
\end{aligned}
$$

反向传播算法

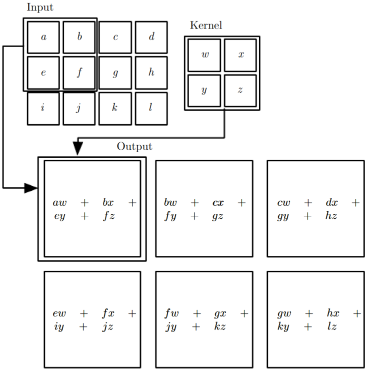
梯度下降法

从数学上的角度来看，梯度的方向是函数增长速度最快的方向，那么梯度的反方向就是函数减少最快的方向。那么，如果想计算一个函数的最小值，就可以使用梯度下降法的思想来做。假设希望求解目标函数 $f(\mathbf{x}) = f(x_1, \cdots, x_n)$ 的最小值，可以从一个初始点 $\mathbf{x}^{(0)} = (x_1^{(0)}, \cdots, x_n^{(0)})$ 开始，基于学习率 $\eta > 0$ 构建一个迭代过程：当 $i \geq 0$ 时，

$$
\begin{aligned}
x_1^{(i+1)} &= x_1^{(i)} - \eta \cdot \frac{\partial f}{\partial x_1}(\mathbf{x}^{(i)}), \\
&\cdots \\
x_n^{(i+1)} &= x_n^{(i)} - \eta \cdot \frac{\partial f}{\partial x_n}(\mathbf{x}^{(i)}).
\end{aligned}
$$

其中 $\mathbf{x}^{(i)} = (x_1^{(i)}, \cdots, x_n^{(i)})$ ，一旦达到收敛条件的话，迭代就结束。

卷积运算



$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

$$S(0,0) = (I * K)(0,0) = \sum_m \sum_n I(0+m, 0+n)K(m,n)$$

$$S(1,0) = (I * K)(1,0) = \sum_m \sum_n I(1+m, 0+n)K(m,n)$$

$$S(2,0) = (I * K)(2,0) = \sum_m \sum_n I(2+m, 0+n)K(m,n)$$

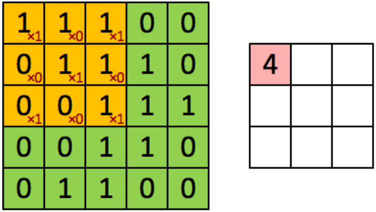
$$S(0,1) = (I * K)(0,1) = \sum_m \sum_n I(0+m, 1+n)K(m,n)$$

$$S(1,1) = (I * K)(1,1) = \sum_m \sum_n I(1+m, 1+n)K(m,n)$$

$$S(2,1) = (I * K)(2,1) = \sum_m \sum_n I(2+m, 1+n)K(m,n)$$

Convolutions

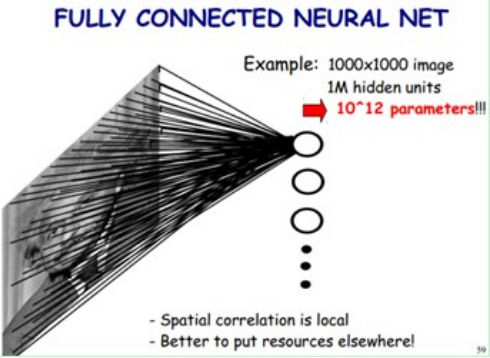
- Suppose you want to learned 9 features from a 5x5 image.
  - With Fully Connected Neural Networks:
    $$5\times5\times9 = 225$$
  - With Locally Connected Neural Networks:
    $$3\times3\times9 = 81$$
  - With Weights Sharing:
    $$3\times3\times1 = 9$$

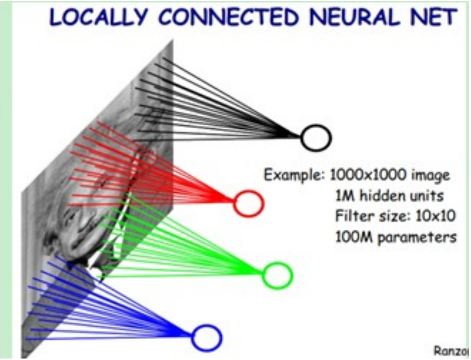


Image

Convolved Feature

Locally Connected Networks



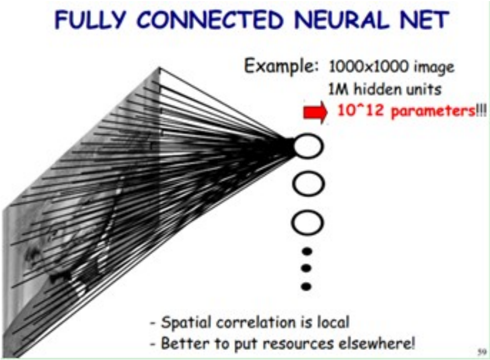


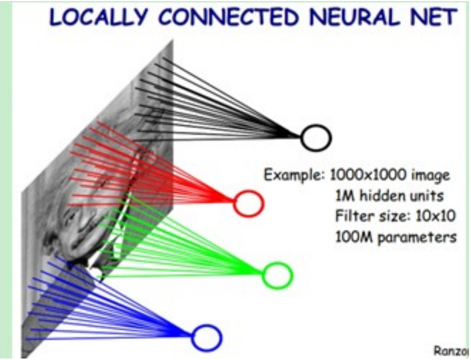
Suppose there are 1M hidden units:

Left: $1000 \times 1000 \times 1M = 10^{12}$

Right: $10 \times 10 \times 1M = 10^{8}$

Weights Sharing





Suppose there are 1M hidden units:

Left: $1000 \times 1000 \times 1M = 10^{12}$

Right: $10 \times 10 \times 1 = 10^{2}$

卷积运算

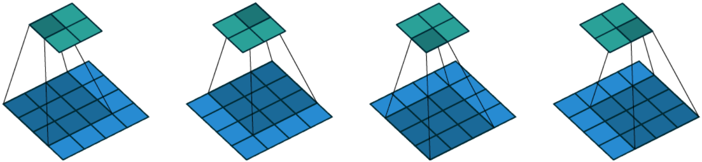


Figure 2.1: (No padding, no strides) Convolving a 3×3 kernel over a 4×4 input using unit strides (i.e., $i = 4$, $k = 3$, $s = 1$ and $p = 0$).

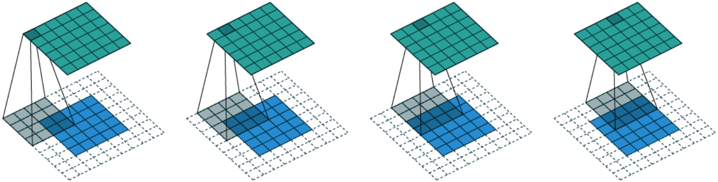


Figure 2.2: (Arbitrary padding, no strides) Convolving a 4×4 kernel over a 5×5 input padded with a 2×2 border of zeros using unit strides (i.e., $i = 5$, $k = 4$, $s = 1$ and $p = 2$).

卷积运算

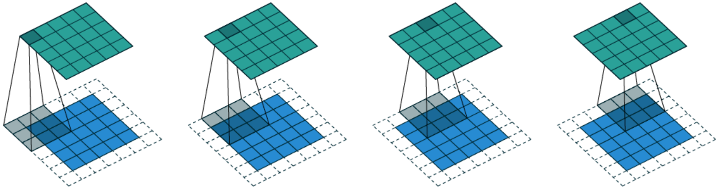


Figure 2.3: (Half padding, no strides) Convolving a 3×3 kernel over a 5×5 input using half padding and unit strides (i.e., $i = 5$, $k = 3$, $s = 1$ and $p = 1$).

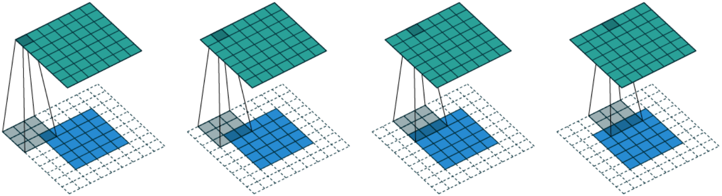


Figure 2.4: (Full padding, no strides) Convolving a 3×3 kernel over a 5×5 input using full padding and unit strides (i.e., $i = 5$, $k = 3$, $s = 1$ and $p = 2$).

卷积运算的物理含义

$$K_{horizontal\_high\_magnitude} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

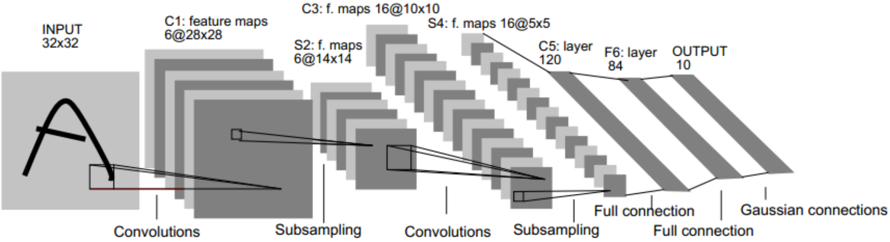
(a) Lenna

(b) Horizontal edge

(c) Vertical edge

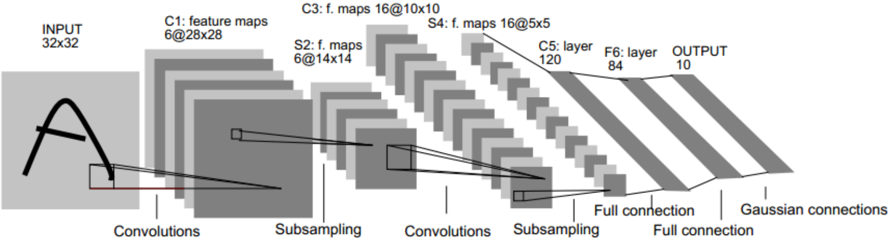
LeNet-5

- C1层是一个卷积层
  - 6个特征图，每个特征图中的每个神经元与输入中5*5的邻域相连，特征图大小为28*28，
  - 每个卷积神经元的参数数目：5*5=25个unit参数和一个bias参数，
  - 连接数目： (5*5+1)*6*(28*28)=122,304个连接
  - 参数共享：每个特征图内共享参数，因此参数总数：共(5*5+1)*6=156个参数



LeNet-5

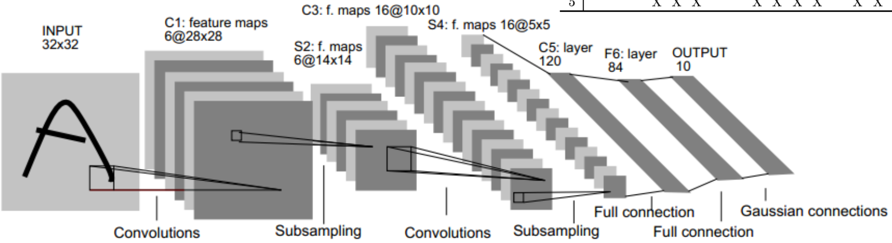
- S2层是一个下采样层
  - 6个14*14的特征图，每个图中的每个单元与C1特征图中的一个2*2邻域相连接，不重叠。因此，S2中每个特征图的大小是C1中特征图大小的1/4.
  - S2层每个单元的4个输入相加，乘以一个可训练参数w，再加上一个可训练偏置b，结果通过sigmoid函数计算。
  - 连接数：（2*2+1）*1*14*14*6 = 5880个
  - 参数共享：每个特征图内共享参数，因此有(2*2+1)*6=30个可训练参数



LeNet-5

- C3层是一个卷积层
  - 16个卷积核，得到16张特征图，特征图大小为10*10；
  - 每个特征图中的每个神经元与S2中某几层的多个5*5的邻域相连；

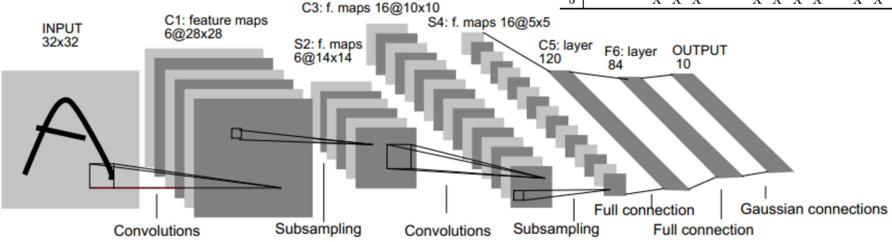
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X



LeNet-5

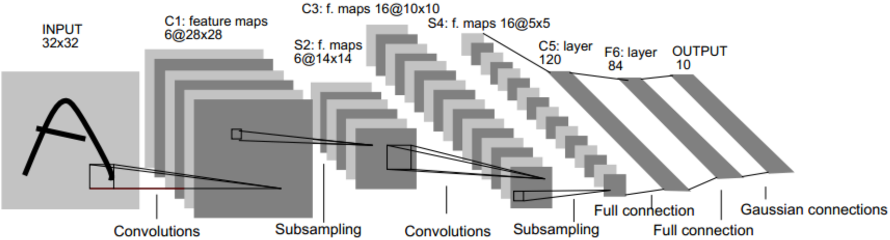
- C3层是一个卷积层
  - 16个卷积核，得到16张特征图，特征图大小为10*10；
  - 每个特征图中的每个神经元与S2中某几层的多个5*5的邻域相连；
    - 例如，对于C3层第0张特征图，其每一个节点与S2层的第0张特征图，第1张特征图，第2张特征图，总共3个5x5个节点相连接。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X



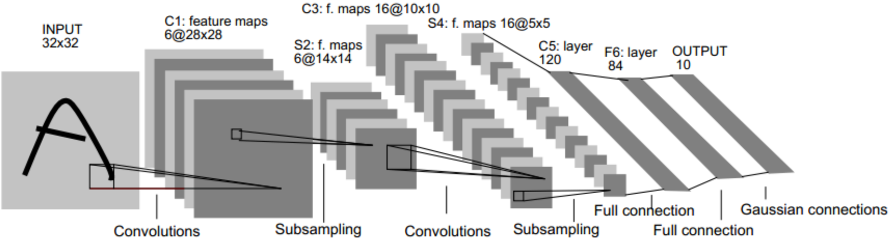
LeNet-5

- S4层是一个下采样层
  - 由16个5*5大小的特征图构成，特征图中的每个单元与C3中相应特征图的2*2邻域相连接；
  - 连接数：（2*2+1）*5*5*16=2000个
  - 参数共享：特征图内共享参数，每张特征图中的每个神经元需要1个因子和一个偏置，因此有 2*16个可训练参数



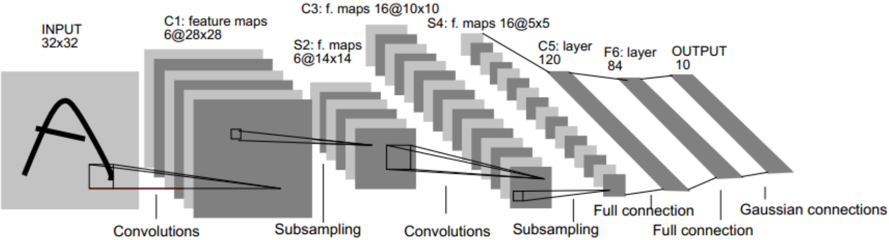
LeNet-5

- C5层
  - 120个神经元，可以看作120个特征图，每张特征图的大小为1*1
  - 每个单元与S4层的全部16个单元的5*5邻域相连（S4和C5之间的全连接）
  - 连接数=可训练参数：（5*5*16+1）*120=48120个



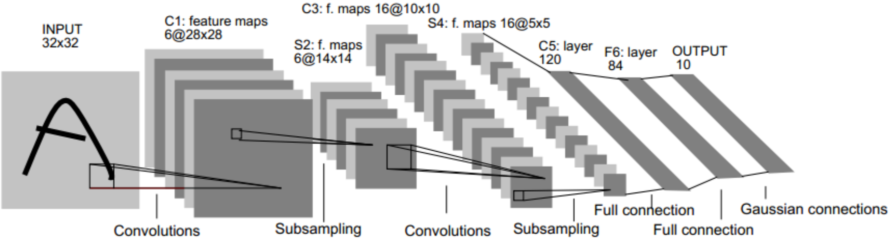
LeNet-5

- F6层
  - 有84个单元（之所以选这个数字的原因来自于输出层的设计），与C5层全相连。
  - F6层计算输入向量和权重向量之间的点积，再加上一个偏置。
  - 连接数=可训练参数：（120+1）*84=10164
  - 84：stylized image：7*12



LeNet-5

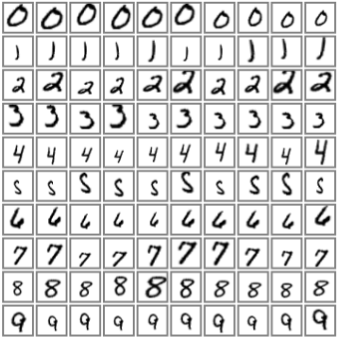
- 输出层采用欧式径向基函数（Euclidean Radial Basis Function）单元
  - 给定一个输入模式，损失函数应能使得F6的配置与RBF参数向量（即模式的期望分类）足够接近。
  - 每类一个单元，每个单元连接84个输入；每个输出RBF单元计算输入向量和参数向量之间的欧式距离。
  - RBF输出可以被理解为F6层配置空间的高斯分布的【-log-likelihood】



LeNet-5 on MNIST

60,000 original datasets

Test error: 0.95%



540,000 artificial distortions

+ 60,000 original

Test error: 0.8%

错误识别分析


4->6 3->5 8->2 2->1 5->3 4->8 2->8 3->5 6->5 7->3
9->4 8->0 7->8 5->3 8->7 0->6 3->7 2->7 8->3 9->4
8->2 5->3 4->8 3->9 6->0 9->8 4->9 6->1 9->4 9->1
9->4 2->0 6->1 3->5 3->2 9->5 6->0 6->0 6->0 6->8
4->6 7->3 9->4 4->6 2->7 9->7 4->3 9->4 9->4 9->4
8->7 4->2 8->4 3->5 8->4 6->5 8->5 3->8 3->8 9->8
1->5 9->8 6->3 0->2 6->5 9->5 0->7 1->6 4->9 2->1
2->8 8->5 4->9 7->2 7->2 6->5 9->7 6->1 5->6 5->0
4->9 2->8