HHCV COURSE

of

Convolutional Neural Networks

孟朝晖

河海大学 **HHCV**

Computer Vision Group of Hohai University

LeNet-5 为例解释卷积网络

原始论文 Gradient-based Learning Applied to Document Recognition, by Yann LeCun etc., 1998.

LeNet-5 基本概念

LeNet-5 为多层前馈有监督型卷积神经网络，最初设计目的为实现手写数字识别功能。

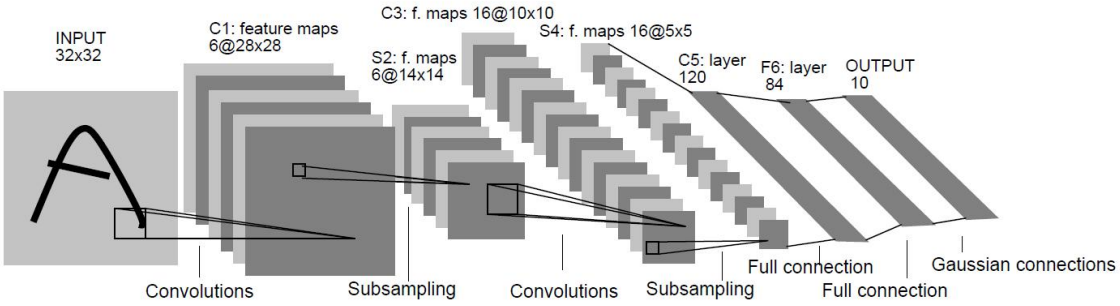


Figure 1 Architecture of LeNet-5, a Convolutional Neural Network, for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

图 1 LeNet-5 的架构图，为一个多层卷积神经网络，用于识别手写数字。每个平面（plane）为一个特征图（feature map），是一个神经元（unit）集合，其中每个不同的神经元其权值（weights）是相同的。

feature map 是特征图，从数据处理角度而言是一幅图（二维图），图中每个点有一个值。

从神经网络系统角度理解，要建立一个以神经元为基本要素的概念体系。

在 LeNet-5 的架构中， feature map 是一个二维布局的神经元 neurons 集合，feature map 的每个点是一个 neuron，这样的 neuron 从前层的特征图中的局部区域获取输入值，即前层局部若干个 neurons 的输出值，在本 neuron 中加工计算后获得输出值。

LeNet-5 comprises 7 layers, not counting the input, all of which contain trainable parameters (weights).

LeNet-5 由 Input Layer 输入层、C1 卷积层、S2 池化层、C3 卷积层、S4 池化层、C5 卷积层、F6 全连接层、OUTPUT 输出层组成，其中 OUTPUT 输出层在 1998 年的原始版本中是径向基函数网络（RBF : Euclidean Radial Basis Function units），目前这一层的网络功能普遍由 softmax 网络完成，所以，本教程用 softmax 网络代替原 RBF 网络，其它部分的网络结构仍然与原 LeNet-5 一致。

MNIST 手写数字数据集

MNIST 手写数字数据集（The MNIST database of handwritten digits）中包含 60000 个训练用数字图片 images（28×28，1 byte per dot，即每个数字图片由 28×28=784 个 unsigned char 0-255 数据组成），10000 个测试用数字图片 images，每个图片均有标记 labels（即数字 0-9），

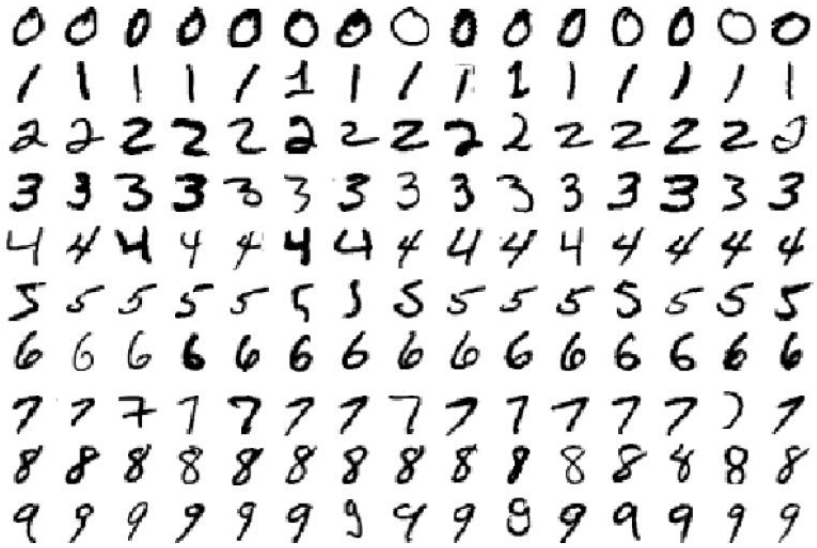
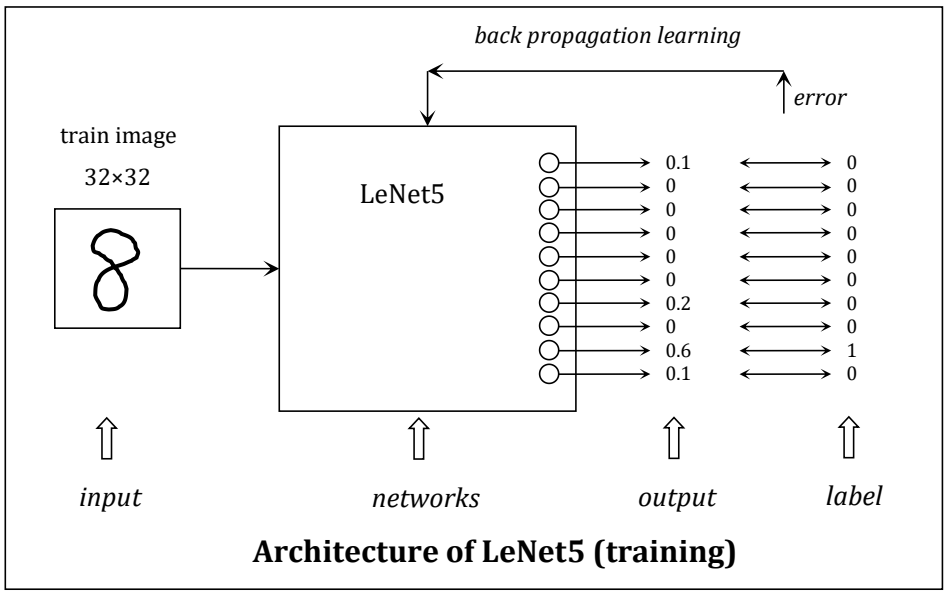


图 2 The MNIST database of handwritten digits

数据存储在以下四个二进制文件中，文件数据格式另文说明：

train-images.idx3-ubyte: training set images (9912422 bytes)
train-labels.idx1-ubyte: training set labels (28881 bytes)
t10k-images.idx3-ubyte: test set images (1648877 bytes)
t10k-labels.idx1-ubyte: test set labels (4542 bytes)

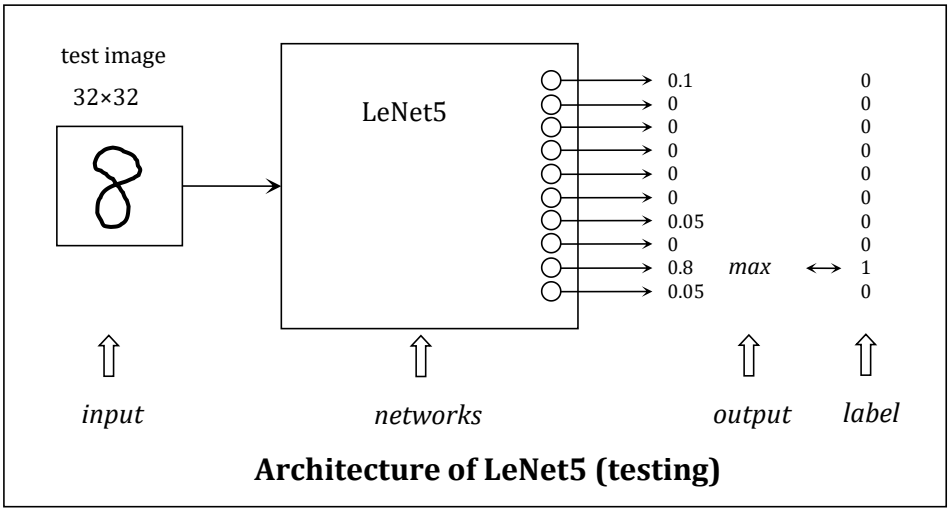
LeNet-5 运行方式

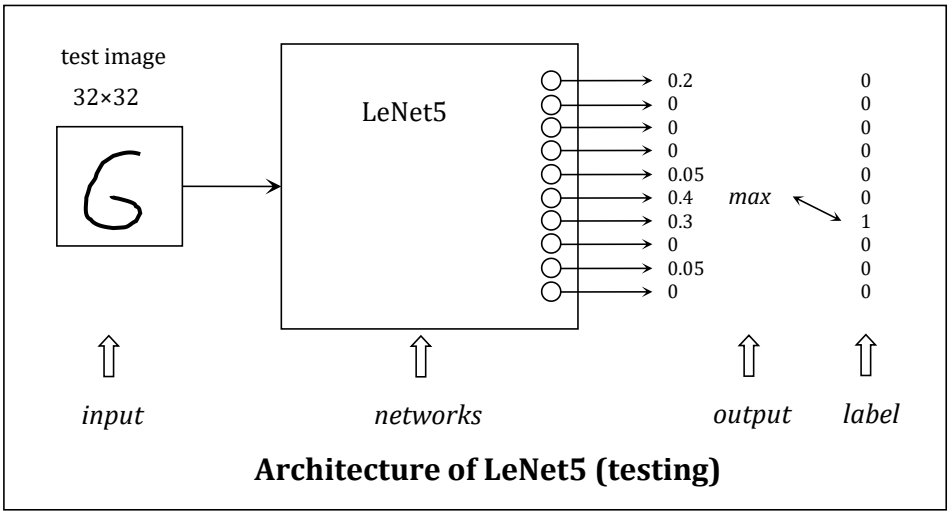


1 训练阶段（training phase）：先将 LeNet-5 看作是一个黑箱模型，输入一张训练图片给网络，网络输出一个 10 维向量，与训练图片的对应标签向量进行比对，差向量进行反向传播，经过学习算法，修正网络的参数，每一张训练图片均进行这样的学习过程；再输入下一张训练图片，直到 60000 张图片学习训练完成。

其中的**标签向量**也为 10 维，称为 one hot vector，就是只有一个分量为 1（该分量位置对应标记数字 0-9），其它分量为 0 的向量。

2 测试阶段（testing phase）：输入一张测试图片给网络，网络输出一个 10 维向量，该向量的最大分量位置对应的标记数字（0-9）即为 LeNet-5 网络对输入测试图片给出的判断，再与测试图片对应的标签向量的最大分量标记数字（0-9）进行比对，统计错判数；再输入下一张图片，直到 10000 张测试图片测试完成给出最终的错误率。





Layer INPUT

The input is a 32×32 pixel image.
输入层为 32×32 像素图片。

This is significantly large than the largest character in the database(at most 20×20 pixels centered in a 28×28 field). The reason is that it is desirable that potential distinctive features such as stroke end-points or corner can appear in the center of the receptive field of the highest-level feature detectors.

原始 MNIST 数据为 28×28 像素图片，神经网络的输入图片扩展为 32×32，目的是在网络的后面层次（C3 层）仍然保证原图片边缘位置的数据得以保留。

卷积网络的 input 层通常不算一个网络层，即不算是 Neuron 层，只提供输入数据，这里 input 是一幅图，其数据可以表达为矩阵 $\left[x_{ij}^{(k)}\right]_{32\times32}$， k 表示第 k 个输入数据（训练数据）。为了表达简洁方便，可以省略 k，表示为 $\left[x_{ij}\right]_{32\times32}$。

In LeNet-5 the set of centers of the receptive fields of the last convolution layer (C3, see below) form a 20×20 area in the center of the 32×32 input.

The values of the input pixels are normalized so that the background level (white) corresponds to a value of -0.1 and the foreground (black) corresponds to 1.175. This makes the mean input roughly 0, and the variance roughly 1 which accelerates learning.

原始的图像像素数值通常不会直接用来做输入数据 $\left[x_{ij}^{(k)}\right]_{32\times32}$，要经过预处理。比如，将背景（白）对应为-0.1，将前景（黑）对应为 1.175，使得输入均值近似为 0，方差近似为 1。

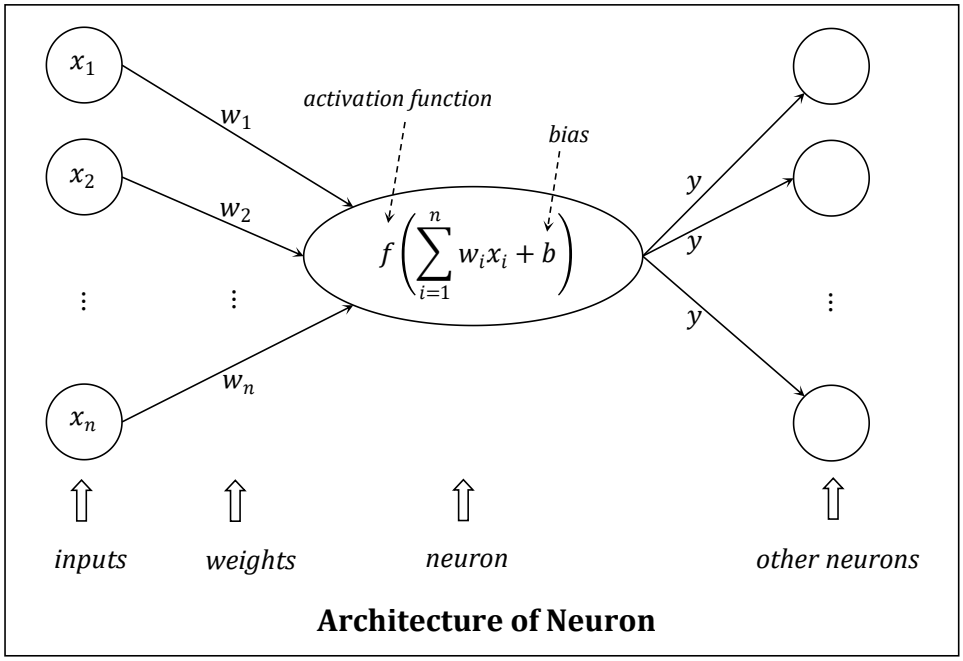
In following, convolutional layers are labeled **Cx**, subsampling layers are labeled **Sx**, and fully-connected layers are labeled **Fx**, where x is the layer index.

convolutional layers 卷积层中的每个 neuron 做卷积操作。subsampling layers 下采样层的每个 neuron 做采样操作，实际上是将大的 feature map 缩为小的 feature map，目前常用 pooling 算法。fully-connected layers 是全连接层，提取全局特征。OUTPUT 输出层为分类层，目前常用 softmax 算法。

<u>神经元 Neuron 基本概念</u>

神经元是神经网络的基本计算单元，一个神经元有若干输入值，设为 $\{x_1,\cdots,x_n\}$，这些输入值来源于其它神经元（或输入层数据），神经元内有与输入值对应的权值，设为 $\{w_1,\cdots,w_n\}$，还有一个偏差值 b，神经元的输出值 y 按下式计算，

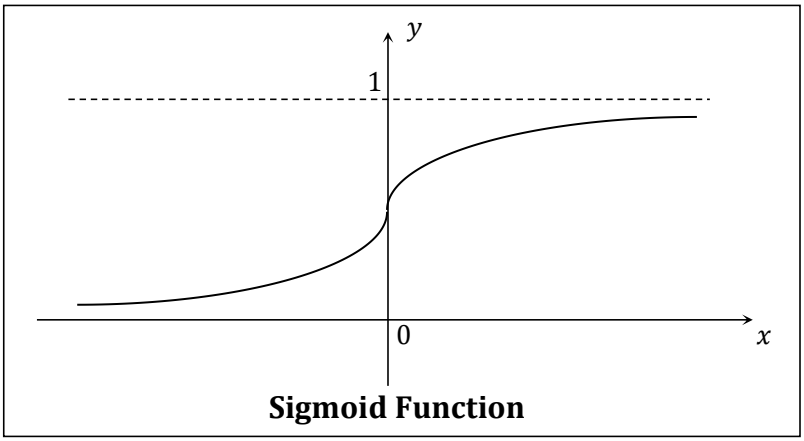
$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$



Architecture of Neuron

其中 f 为激活函数，为非线性函数，通常选取 sigmoid 函数，

$$sigmoid\ function: \qquad f(x) = \frac{1}{1+e^{-ax}} \qquad a > 0$$



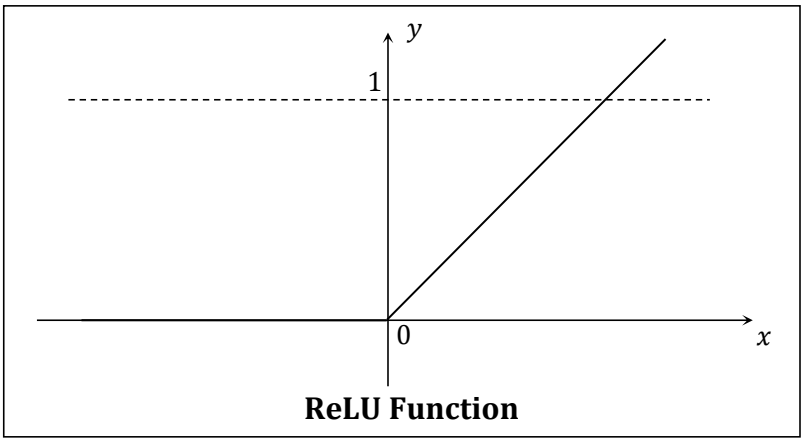
Sigmoid Function

或 tanh 函数，

$$tanh\ function: \qquad f(x) = \frac{1-e^{-ax}}{1+e^{-ax}} \qquad a > 0$$

目前比较常用另外一个非线性激活函数 ReLU（Rectified Linear Unit）定义如下：

$$y = f(x) = \max(0, x)$$



ReLU Function

Layer C1

Layer **C1** is a convolutional layer with 6 **feature maps**. Each **unit** in each feature map is connected to a 5×5 neighborhood in the input. The size of the feature maps is 28×28 which prevents connection from the input from falling off the boundary.

如前所述，特征图 feature map 是一个二维布局的神经元 neurons 集合，feature map 的每个点是一个 neuron，这样的 neuron 从前层的特征图中的局部区域获取输入值，即前层局部若干个 neurons 的输出值，在本 neuron 中加工计算后获得输出值。

具体到 **C1** 层，称为一个卷积层，由 6 个特征图组成，每个特征图有 28×28 个二维布局的神经元，每个神经元从输入层（32×32 像素图片）的一个 5×5 的局部邻域获取输入数据，局部邻域的位置由特征图的二维布局座标与输入层的二维座标相对应，输入层尺寸为 32×32，特征图为 28×28，可以保证每个神经元的每个输入都能取到值。

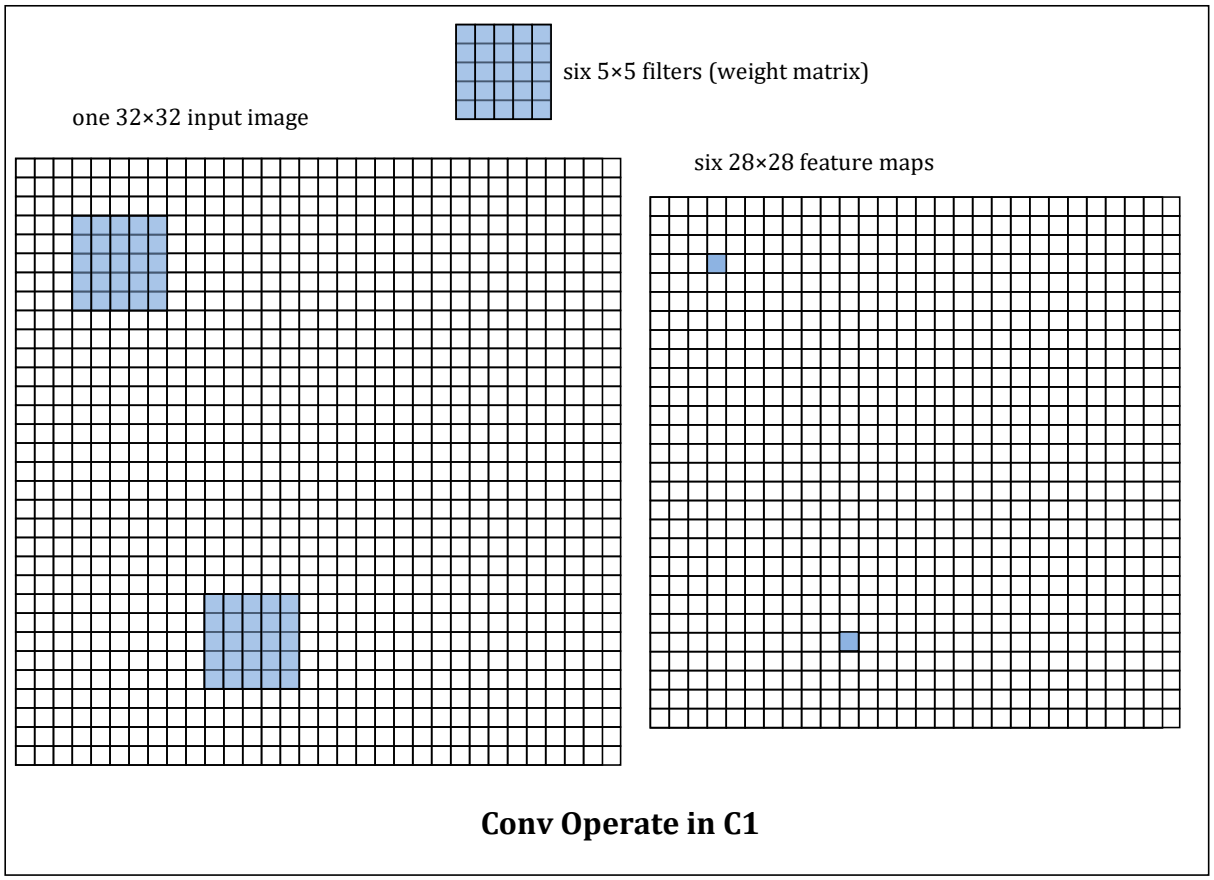
共享权值，每个特征图有 28×28 个二维布局的神经元，所有这些 784 个神经元的权值是一致的，也就是说，同一个特征图中的神经元的计算核（权值 $\{w_1,\cdots,w_n\}$ 和偏差 b）是一样的，只是输入值 $\{x_1,\cdots,x_n\}$ 不同，$n = 5 \times 5 = 25$。

换一个角度看，实际上，就是用 5×5 的卷积核(filter)扫描 input 32×32 pixel image，获得 1 个 28×28 的 feature map(28=32-5+1)，即 5×5 的扫描窗(filter)不超出 32×32 的输入图(pixel image)的边框，这种卷积核有 6 个，所以有 C1 层得到 6 个 28×28 的 feature maps。(C1:feature maps, 6@28×28)

C1 contains 156 trainable parameters, and 122304 connections.

C1 层共有 156 可训练参数，以及 122304 个连接。

Note：156，122304 是如何计算的？
每个卷积核有 5×5=25 个 weight 参数和 1 个 bias 参数，则 6 个卷积核有 26×6=156 个 trainable parameters，feature map 中的每个 unit 都有 26 个 connections，每个 feature map 有 28×28 个 units，一共 6 个 feature maps，共有 28×28×26×6=122304 个连接。



卷积操作（每个神经元）的表达式如下：

$$c1_{ij}^{(k)} = f\left(\sum_{u=-2}^{2}\sum_{v=-2}^{2} w1_{u,v}^{(k)} \cdot x_{i+2+u,j+2+v} + b1^{(k)}\right) \qquad \begin{cases} i,j = 0,\cdots,27 \\ k = 0,\cdots,5 \end{cases}$$

经过以上卷积操作，得到 6 个 feature maps，每个 $c1_{ij}^{(k)} = f(\cdot)$ 就是一个 neuron，共有 28×28×6=4704 个 neurons。$\{k = 0,\cdots,5\}$ 为 6 个 feature maps 的编号。

$$\begin{matrix} \left[c1_{ij}^{(0)}\right]_{28\times28} & \left[c1_{ij}^{(1)}\right]_{28\times28} & \left[c1_{ij}^{(2)}\right]_{28\times28} \\ \left[c1_{ij}^{(3)}\right]_{28\times28} & \left[c1_{ij}^{(4)}\right]_{28\times28} & \left[c1_{ij}^{(5)}\right]_{28\times28} \end{matrix}$$

6 个卷积核，相当于是权值集合，

$$\begin{bmatrix} w1_{-2,-2}^{(k)} & w1_{-2,-1}^{(k)} & w1_{-2,0}^{(k)} & w1_{-2,1}^{(k)} & w1_{-2,2}^{(k)} \\ w1_{-1,-2}^{(k)} & w1_{-1,-1}^{(k)} & w1_{-1,0}^{(k)} & w1_{-1,1}^{(k)} & w1_{-1,2}^{(k)} \\ w1_{0,-2}^{(k)} & w1_{0,-1}^{(k)} & w1_{0,0}^{(k)} & w1_{0,1}^{(k)} & w1_{0,2}^{(k)} \\ w1_{1,-2}^{(k)} & w1_{1,-1}^{(k)} & w1_{1,0}^{(k)} & w1_{1,1}^{(k)} & w1_{1,2}^{(k)} \\ w1_{2,-2}^{(k)} & w1_{2,-1}^{(k)} & w1_{2,0}^{(k)} & w1_{2,1}^{(k)} & w1_{2,2}^{(k)} \end{bmatrix} \qquad k = 0,\cdots,5$$

每个卷积核附加一个 bias 参数，$b1^{(k)}, k = 0,\cdots,5$

C1 层功能解读：卷积核(filter)实际上就是小的特征模版(5×5 feature window)，卷积的作用就是用特征模版在全图(input image)逐点计算每个点及其邻域 $\left(\{x_{i+2+u,j+2+v}\}_{\substack{u=[-2,2]\\v=[-2,2]}}\right)$ 与该模版的符合度(卷积值 $\sum_{u=-2}^{2}\sum_{v=-2}^{2} w1_{u,v}^{(k)} \cdot x_{i+2+u,j+2+v}$)，并通过截距 $b1^{(k)}$ 和激活函数 $f(\cdot)$ 求出输出值 $c1_{ij}^{(k)}$，该值用来衡量此点是否具有该特征，并记录下来，形成一个新的特征图(feature map)，$\left[c1_{ij}^{(k)}\right]_{\substack{i=0,\cdots,27\\j=0,\cdots,27}}$，C1 层有 6 个不同的卷积核($k = 0,\cdots,5$)，就有 6 个特征图。

Layer S2

Layer **S2** is a sub-sampling layer with 6 feature maps of size 14×14. Each unit in each feature map is connected to a 2×2 neighborhood in the corresponding feature map in C1.

S2 层为下采样层，有 6 个特征图，每个特征图为 14×14 二维布局的神经元阵列共 196 个神经元，每个神经元的输入接口与 C1 层对应位置的 2×2 邻域共 4 个神经元的输出接口相连接。

S2 层就是将 28×28 的 feature map 缩减为 14×14 的 feature map，前层的 2×2 四个 neurons 输出给本层一个 neuron，采样域不重叠。

The four inputs to a unit in S2 are added, then multiplied by a **trainable coefficient**, and added to a **trainable bias**. The result is passed through a **sigmoidal function**.

S2 层每个神经元的 4 个输入值相加，然后乘以一个可训练系数，再加上 1 个可训练偏差值，其结果再经过非线性 **sigmoid** 函数的变换作为 S2 层该神经元的输出。

The 2×2 receptive fields are non-overlapping, therefore feature maps in S2 have half the number of rows and column as feature maps in C1. Layer S2 has **12 trainable parameters** and 5880 connections.

S2 层每个神经元的 2×2 接受域不相互重叠，则 S2 层特征图的维度恰好为 C1 层特征图的一半。S2 层共有 12 个可训练参数和总计 5880 个连接。

原始的 LeNet-5 的 sub-sampling 认为每个局部区域的下采样操作也是一个神经元（如上所述），神经元就要有非线性激活步骤，按照标准的 sigmoid 激活模式，表达为

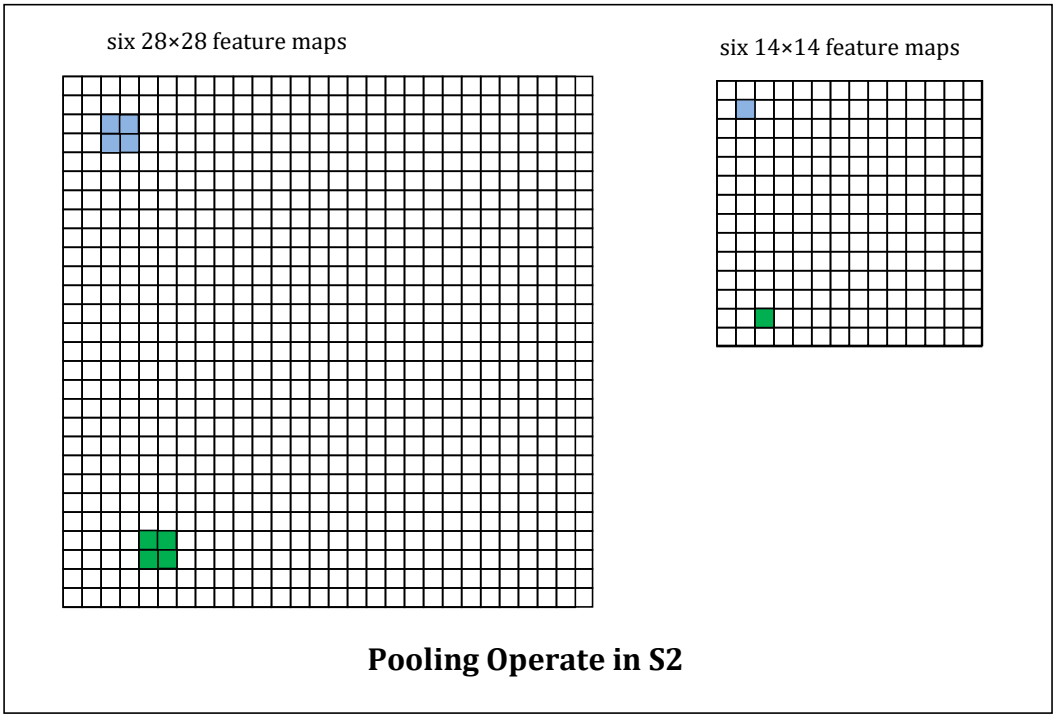
$$s2_{ij}^{(k)} = f\left(a2^{(k)} \cdot \left(\sum_{u=0}^{1}\sum_{v=0}^{1} c1_{2i+u,2j+v}^{(k)}\right) + b2^{(k)}\right) \qquad \begin{cases} i,j = 0,\cdots,13 \\ k = 0,\cdots,5 \end{cases}$$

经过以上 sub-sampling 操作，得到 6 个 feature maps，每个 $s2_{ij}^{(k)} = f(\cdot)$ 就是一个 neuron，共有 14×14×6=1176 个 neurons。$\{k = 0,\cdots,5\}$ 为 6 个 feature maps 的编号。

$$\left[s2_{ij}^{(0)}\right]_{14\times14} \quad \left[s2_{ij}^{(1)}\right]_{14\times14} \quad \left[s2_{ij}^{(2)}\right]_{14\times14}$$
$$\left[s2_{ij}^{(3)}\right]_{14\times14} \quad \left[s2_{ij}^{(4)}\right]_{14\times14} \quad \left[s2_{ij}^{(5)}\right]_{14\times14}$$

$a2^{(k)}$ 为 6 个 **trainable coefficient**， $b2^{(k)}$为 6 个 **trainable bias**。

Note：12，5880 是如何计算的？
有 6 个下采样核，每个核有 1 个 **trainable coefficient** ，1 个 **trainable bias**。共有 12 个 trainable parameters。
计算 connection 时，每个下采样核的 1 个 **trainable coefficient** 算 4 个 connections，加上 1 个 **trainable bias**，每个 unit in S2 有 5 个 connections，共有 5×14×14×6=5880 个 connections。



目前普遍认为，sub-sampling 操作就是一个**池化操作**（pooling），省略可训练系数 $a2^{(k)}$，省略偏差 $b2^{(k)}$ 相加和非线性 sigmoid 激活步骤，直接采用 max-pooling 算法

$$s2_{ij}^{(k)} = \max\left\{c1_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}} \qquad \begin{cases} i,j = 0,\cdots,13 \\ k = 0,\cdots,5 \end{cases}$$

或 mean-pooling 算法

$$s2_{ij}^{(k)} = \frac{1}{4}\cdot\left(\sum_{u=0}^{1}\sum_{v=0}^{1} c1_{2i+u,2j+v}^{(k)}\right) \qquad \begin{cases} i,j = 0,\cdots,13 \\ k = 0,\cdots,5 \end{cases}$$

还有 stochastic pooling 算法等，不做详细介绍了。

S2 层功能解读：S2 层将 C1 层的 6 个特征图分别缩小 1 倍，在尺寸缩小的前提下，要保留 C1 层特征图中的显著特征，这种保留分两个方面解读：

其一，位置如何保留，注意到 S2 层的位置信息的解像度(resolution)下降了一半，即位置更模糊了，同时获得了小范围的平移不变性(invariant of translation)。

其二，输出值如何保留，max-pooling 方法直接保留 C1 层相应领域(2×2 receptive fields)中最显著的特征衡量值，mean-pooling 算法求平均值。

Layer C3

Layer C3 is a convolutional layer with 16 feature maps. Each unit in each feature map is connected to several 5×5 neighborhoods at identical locations in a subset of S2's feature maps. Table 1 shows the set of S2 feature maps combined by each C3 feature map.

C3 层为卷积层，有 16 个特征图，C3 层特征图中的每个神经元从若干个 S2 特征图中相同位置的 5×5 邻域神经元输出获取输入数据，表 1 表达了每个 C3 层特征图的输入由 S2 特征图组合而成的集合表示。

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

Table 1 Each column indicates which feature map in S2 are combined by the units in a particular feature map of C3.

表 1 每一列表达每个（0-15 个）C3 特征图中的神经元由哪几个 S2 特征图中的数据组合输入。

实际上，就是用 5×5 的单模版卷积核(single filter)扫描 3 个或 4 个或 6 个 14×14 pixel image in S2，再 3 个或 4 个或 6 个 combined 获得 1 个 10×10 的 feature map(10=14-5+1)，这种组合卷积核(combined filters)有 16 个，所以有 C3 层得到 16 个 10×10 的 feature maps.(C3:feature maps, 16@10×10)

这层的卷积操作复杂一些，表中共有 60 个 5×5 的单卷积核（打×的），相当于是 60 个权值矩阵，

$$\begin{bmatrix} w3_{-2,-2}^{(k,m)} & w3_{-2,-1}^{(k,m)} & w3_{-2,0}^{(k,m)} & w3_{-2,1}^{(k,m)} & w3_{-2,2}^{(k,m)} \\ w3_{-1,-2}^{(k,m)} & w3_{-1,-1}^{(k,m)} & w3_{-1,0}^{(k,m)} & w3_{-1,1}^{(k,m)} & w3_{-1,2}^{(k,m)} \\ w3_{0,-2}^{(k,m)} & w3_{0,-1}^{(k,m)} & w3_{0,0}^{(k,m)} & w3_{0,1}^{(k,m)} & w3_{0,2}^{(k,m)} \\ w3_{1,-2}^{(k,m)} & w3_{1,-1}^{(k,m)} & w3_{1,0}^{(k,m)} & w3_{1,1}^{(k,m)} & w3_{1,2}^{(k,m)} \\ w3_{2,-2}^{(k,m)} & w3_{2,-1}^{(k,m)} & w3_{2,0}^{(k,m)} & w3_{2,1}^{(k,m)} & w3_{2,2}^{(k,m)} \end{bmatrix} \quad \begin{cases} k = 0,\cdots,5 \\ m = 0,\cdots,15 \\ (k,m) \in Cbn \end{cases}$$

其中 Cbn 为（打×的）集合，

$$Cbn = \begin{Bmatrix} (0,0) & (\ ) & (\ ) & (\ ) & (0,4) & (0,5) & (0,6) & (\ ) & (\ ) & (0,9) & (0,10) & (0,11) & (0,12) & (\ ) & (0,14) & (0,15) \\ (1,0) & (1,1) & (\ ) & (\ ) & (\ ) & (1,5) & (1,6) & (1,7) & (\ ) & (\ ) & (1,10) & (1,11) & (1,12) & (1,13) & (\ ) & (1,15) \\ (2,0) & (2,1) & (2,2) & (\ ) & (\ ) & (\ ) & (2,6) & (2,7) & (2,8) & (\ ) & (\ ) & (2,11) & (\ ) & (2,13) & (2,14) & (2,15) \\ (\ ) & (3,1) & (3,2) & (3,3) & (\ ) & (\ ) & (3,6) & (3,7) & (3,8) & (3,9) & (\ ) & (\ ) & (3,12) & (\ ) & (3,14) & (3,15) \\ (\ ) & (\ ) & (4,2) & (4,3) & (4,4) & (\ ) & (\ ) & (4,7) & (4,8) & (4,9) & (4,10) & (\ ) & (4,12) & (4,13) & (\ ) & (4,15) \\ (\ ) & (\ ) & (\ ) & (5,3) & (5,4) & (5,5) & (\ ) & (\ ) & (5,8) & (5,9) & (5,10) & (5,11) & (\ ) & (5,14) & (5,14) & (5,15) \end{Bmatrix}$$

为了表达式的统一性，可以将权值矩阵集合进行扩充，共有 6×16=96 个权值矩阵如下，

$$\begin{bmatrix} w3_{-2,-2}^{(k,m)} & w3_{-2,-1}^{(k,m)} & w3_{-2,0}^{(k,m)} & w3_{-2,1}^{(k,m)} & w3_{-2,2}^{(k,m)} \\ w3_{-1,-2}^{(k,m)} & w3_{-1,-1}^{(k,m)} & w3_{-1,0}^{(k,m)} & w3_{-1,1}^{(k,m)} & w3_{-1,2}^{(k,m)} \\ w3_{0,-2}^{(k,m)} & w3_{0,-1}^{(k,m)} & w3_{0,0}^{(k,m)} & w3_{0,1}^{(k,m)} & w3_{0,2}^{(k,m)} \\ w3_{1,-2}^{(k,m)} & w3_{1,-1}^{(k,m)} & w3_{1,0}^{(k,m)} & w3_{1,1}^{(k,m)} & w3_{1,2}^{(k,m)} \\ w3_{2,-2}^{(k,m)} & w3_{2,-1}^{(k,m)} & w3_{2,0}^{(k,m)} & w3_{2,1}^{(k,m)} & w3_{2,2}^{(k,m)} \end{bmatrix} \quad \begin{cases} k = 0,\cdots,5 \\ m = 0,\cdots,15 \end{cases}$$

其中 36 个 $(k,m) \notin Cbn$ 的权值矩阵，其分量元素值 $w3_{u,v}^{(k,m)}$ 都为 0，

$$w3_{u,v}^{(k,m)} = 0 \qquad \begin{cases} k = 0,\cdots,5 \\ m = 0,\cdots,15 \end{cases} \quad \begin{cases} u = -2,-1,0,1,2 \\ v = -2,-1,0,1,2 \end{cases} \quad (k,m) \notin Cbn$$

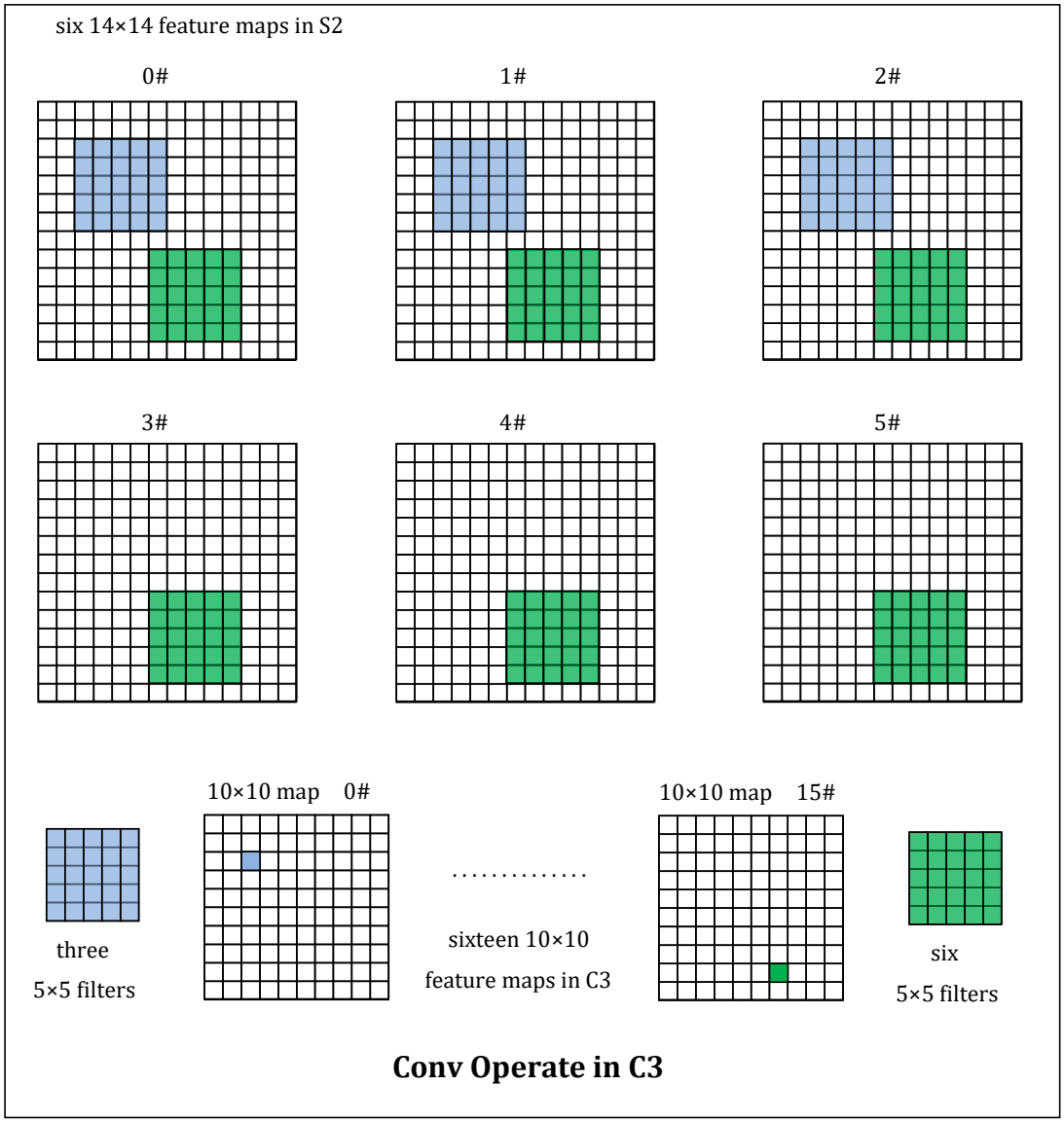
组合卷积操作有 16 个，$m = 0,\cdots,15$，综合表达如下：

$$c3_{ij}^{(m)} = f\left(\sum_{k=0}^{5}\sum_{u=-2}^{2}\sum_{v=-2}^{2} w3_{u,v}^{(k,m)} \cdot s2_{i+2+u,j+2+v}^{(k)} + b3^{(m)}\right) \qquad \begin{cases} i,j = 0,\cdots,9 \\ m = 0,\cdots,15 \end{cases}$$

经过以上卷积操作，得到 16 个 feature maps，每个 $c3_{ij}^{(m)} = f(\cdot)$ 就是一个 neuron，共有 10×10×16=1600 个 neurons。$\{m = 0,\cdots,15\}$ 为 16 个 feature maps 的编号。

$$\left[c3_{ij}^{(0)}\right]_{10\times10} \quad \left[c3_{ij}^{(1)}\right]_{10\times10} \quad \left[c3_{ij}^{(2)}\right]_{10\times10} \quad \left[c3_{ij}^{(3)}\right]_{10\times10}$$
$$\left[c3_{ij}^{(4)}\right]_{10\times10} \quad \left[c3_{ij}^{(5)}\right]_{10\times10} \quad \left[c3_{ij}^{(6)}\right]_{10\times10} \quad \left[c3_{ij}^{(7)}\right]_{10\times10}$$
$$\left[c3_{ij}^{(8)}\right]_{10\times10} \quad \left[c3_{ij}^{(9)}\right]_{10\times10} \quad \left[c3_{ij}^{(10)}\right]_{10\times10} \quad \left[c3_{ij}^{(11)}\right]_{10\times10}$$
$$\left[c3_{ij}^{(12)}\right]_{10\times10} \quad \left[c3_{ij}^{(13)}\right]_{10\times10} \quad \left[c3_{ij}^{(14)}\right]_{10\times10} \quad \left[c3_{ij}^{(15)}\right]_{10\times10}$$

每个组合卷积附加一个 bias 参数，$b3^{(m)}, m = 0,\cdots,15$



注记：原始论文对于为什么不对 S2 feature map 到 C3 feature map 做全参与式组合卷积（相当于前述 6×16=96 个权值矩阵），而采用一个人为设计过的表格这个问题作了一些解释（如下），近几年的深度卷积网络已经不这样做了，不再人工设计特定的层间特征图连接方式，所以本节也是按照统一的扩展权值矩阵组来表述，只是附加一个筛选集合 Cbn 来表达 LeNet5 的特殊模式。

Why not connect every S2 feature map to every C3 feature map? The reason is twofold.

First, a non-complete connection scheme keeps the number of connections within reasonable bounds. More importantly, it forces a break of symmetry in the network. Different feature maps are forced to extract different features because they get different sets of inputs.

The rationale behind the connection scheme in table 1 is the following. **The first** six C3 feature maps take inputs from every contiguous subset of three feature maps in S2. **The next** six take input from every contiguous subset of four. **The next** three take input from some discontinuous subsets of four. **Finally** the last one takes input from all S2 feature maps.

Layer C3 has 1516 trainable parameters and 151600 connections.

Note 1516，151600 是如何计算的？
每个表 1 中的单模版卷积核有 5×5=25 个 weight 参数，表 1 有 60 个这样的单模版卷积核，共有 25×60=1500 个 weight 参数，组合卷积核有 16 个，每个含有 1 个 bias 参数，合计共有 1516 trainable parameters。再乘以 10×10 的 C3_feature maps 尺寸，共计 151600 connections。

C3 层功能解读：这一层将前层的 6 个特征按 16 种方式组合成 16 个组合卷积核(combined filters)，用这种卷积核对前层的 6 个特征图，做卷积、加截距（偏差）、通过激活函数，获得 16 个新的特征图，每个特征图对应一个组合卷积核所表达的组合特征。

Layer S4

Layer S4 is a sub-sampling layer with 16 feature maps of size 5×5. Each unit in each feature map is connected to a 2×2 neighborhood in the corresponding feature map in C3, in a similar way as C1 and S2.

S4 层为下采样层，有 16 个特征图，每个特征图为 5×5 二维布局的神经元阵列共 25 个神经元，每个神经元的输入接口与 C3 层对应位置的 2×2 邻域共 4 个神经元的输出接口相连接。

Layer S4 has 32 trainable parameters and 2000 connections.

与 S2 层类似，原始的 LeNet-5 的 sub-sampling 认为每个局部区域的下采样操作也是一个神经元，神经元就要有非线性激活步骤，按照标准的 sigmoid 激活模式，表达为

$$s4_{ij}^{(k)} = f\left(a4^{(k)} \cdot \left(\sum_{u=0}^{1}\sum_{v=0}^{1} c3_{2i+u,2j+v}^{(k)}\right) + b4^{(k)}\right) \qquad \begin{cases} i,j = 0,\cdots,4 \\ k = 0,\cdots,15 \end{cases}$$

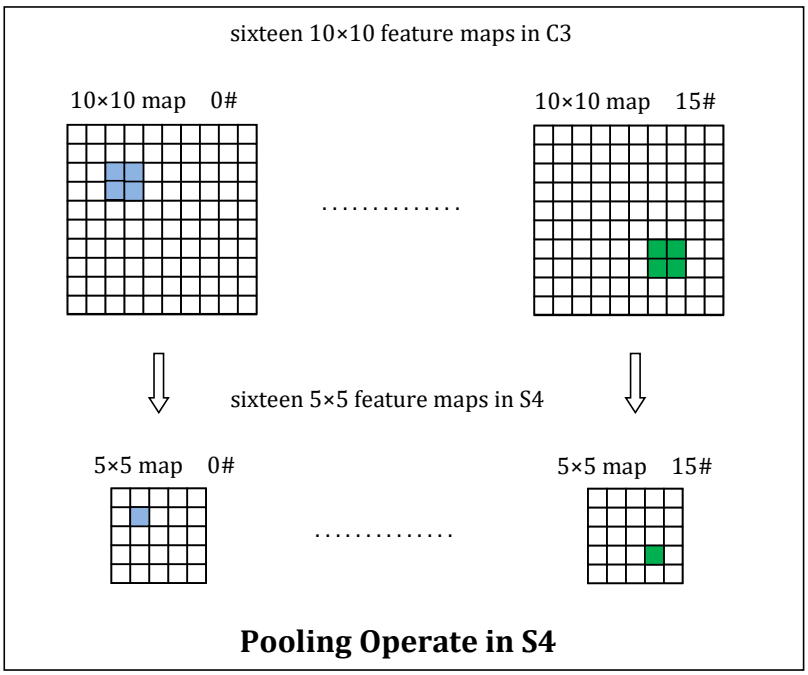
经过以上 sub-sampling 操作，得到 16 个 feature maps，每个 $s4_{ij}^{(k)} = f(\cdot)$ 就是一个 neuron，共有 5×5×16=400 个 neurons。$\{k = 0,\cdots,15\}$ 为 16 个 feature maps 的编号。

$$\begin{matrix}\left[s4_{ij}^{(0)}\right]_{5\times5} & \left[s4_{ij}^{(1)}\right]_{5\times5} & \left[s4_{ij}^{(2)}\right]_{5\times5} & \left[s4_{ij}^{(3)}\right]_{5\times5}\\ \left[s4_{ij}^{(4)}\right]_{5\times5} & \left[s4_{ij}^{(5)}\right]_{5\times5} & \left[s4_{ij}^{(6)}\right]_{5\times5} & \left[s4_{ij}^{(7)}\right]_{5\times5}\\ \left[s4_{ij}^{(8)}\right]_{5\times5} & \left[s4_{ij}^{(9)}\right]_{5\times5} & \left[s4_{ij}^{(10)}\right]_{5\times5} & \left[s4_{ij}^{(11)}\right]_{5\times5}\\ \left[s4_{ij}^{(12)}\right]_{5\times5} & \left[s4_{ij}^{(13)}\right]_{5\times5} & \left[s4_{ij}^{(14)}\right]_{5\times5} & \left[s4_{ij}^{(15)}\right]_{5\times5}\end{matrix}$$

$a4^{(k)}$ 为 16 个 **trainable coefficient**， $b4^{(k)}$为 16 个 **trainable bias**。

Note：32，2000 是如何计算的？
有 16 个下采样核，每个核有 1 个 **trainable coefficient** ，1 个 **trainable bias**。共有 32 个 trainable parameters。
计算 connection 时，每个下采样核的 1 个 **trainable coefficient** 算 4 个 connections，加上 1 个 **trainable bias**，每个 unit in S2 有 5 个 connections，共有 5×(5×5size)×16=2000 个 connections。

与 S2 层类似，目前普遍认为，sub-sampling 操作就是一个**池化操作**（pooling），省略可训练系数 $a4^{(k)}$，省略偏差 $b4^{(k)}$ 相加和非线性 sigmoid 激活步骤，直接采用 max-pooling 算法



目前 sub-sampling 操作大多采用 max-pooling 算法

$$s4_{ij}^{(k)} = \max\left\{c3_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}} \qquad \begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}$$

或 mean-pooling 算法

$$s4_{ij}^{(k)} = \frac{1}{4}\cdot\left(\sum_{u=0}^{1}\sum_{v=0}^{1} c3_{2i+u,2j+v}^{(k)}\right) \qquad \begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}$$

S4 层功能解读：S4 层将 C3 层的 16 个特征图分别缩小 1 倍，在尺寸缩小的前提下，要保留 C3 层特征图中的显著特征。

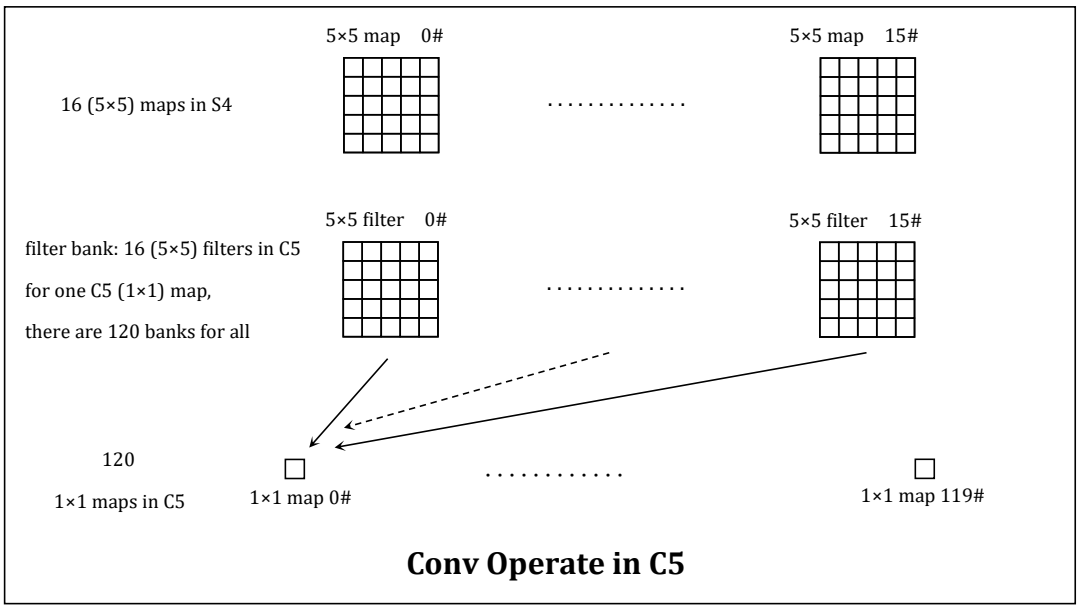
Layer C5

Layer C5 is a convolutional layer with 120 feature maps. Each unit is connected to a 5×5 neighborhood on all 16 of S4's feature maps. Here, because the size of S4 is also 5×5, the size of C5's feature maps is 1×1: this amounts to a full connection between S4 and C5.

C5 层为卷积层，有 120 个特征图。每个神经元连接 S4 特征图中所有的 16 个 5×5 邻域，由于 S4 层每个特征图的尺寸只有 5×5，所以 C5 层 120 个特征图每个的尺寸只能是 1×1，也就是 C5 层的每个特征图只有 1 个神经元。事实上，从 S4 到 C5 为全连接模式。

C5 is labeled as a convolutional layer, instead of a fully-connected layer, because if LeNet-5 input were made bigger with everything else kept constant, the feature map dimension would be larger than 1×1.

C5 层虽然实际上为全连接，但在表达上仍然按照卷积层模式，如果输入层的尺寸更大的话，其特征图就会比 1×1 更大。



Layer C5 has 48120 trainable connections.

卷积操作表达如下：

$$c5_{ij}^{(l)} = f\left(\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2} w5_{u,v}^{(m,l)} \cdot s4_{i+2+u,j+2+v}^{(m)} + b5^{(l)}\right) \qquad \begin{cases} i,j = 0 \\ l = 0,\cdots,119 \end{cases}$$

权值如下：

$$\begin{bmatrix} w5_{-2,-2}^{(m,l)} & w5_{-2,-1}^{(m,l)} & w5_{-2,0}^{(m,l)} & w5_{-2,1}^{(m,l)} & w5_{-2,2}^{(m,l)} \\ w5_{-1,-2}^{(m,l)} & w5_{-1,-1}^{(m,l)} & w5_{-1,0}^{(m,l)} & w5_{-1,1}^{(m,l)} & w5_{-1,2}^{(m,l)} \\ w5_{0,-2}^{(m,l)} & w5_{0,-1}^{(m,l)} & w5_{0,0}^{(m,l)} & w5_{0,1}^{(m,l)} & w5_{0,2}^{(m,l)} \\ w5_{1,-2}^{(m,l)} & w5_{1,-1}^{(m,l)} & w5_{1,0}^{(m,l)} & w5_{1,1}^{(m,l)} & w5_{1,2}^{(m,l)} \\ w5_{2,-2}^{(m,l)} & w5_{2,-1}^{(m,l)} & w5_{2,0}^{(m,l)} & w5_{2,1}^{(m,l)} & w5_{2,2}^{(m,l)} \end{bmatrix} \quad \begin{cases} m = 0,\cdots,15 \\ l = 0,\cdots,119 \end{cases}$$

经过以上卷积操作，得到 120 个 1×1 的 feature maps，每个 $c5_{ij}^{(l)} = f(\cdot)$ 就是一个 neuron，共有 120 个 neurons。相当于有 120 个合成卷积核，120 个 bias 参数，120×16=1920 个小卷积核。

Note　48120 是如何计算的？
1 个 C5 单元对应 S4 中的 16 个 feature maps，每个 map 有卷积核 5×5=25 个 weight 参数，共计 400 个 weight 参数，外加 1 个 bias 参数，即 1 个 C5 单元有 401 个参数。120 个 C5 单元有 48120 个参数。

C5 层功能解读：这一层将前层的 16 个特征按全参加的方式组合成 120 个组合卷积核(combined filters)，用这种卷积核对前层的 16 个特征图，做卷积、加截距、通过激活函数，获得 120 个新的特征图（实际上只是一个特征值），每个特征值对应一个组合卷积核所表达的全图组合特征。

Layer F6

Layer F6, contains 84 units and fully connected to C5. It has 10164(120×84weight + 84bias) trainable parameters.

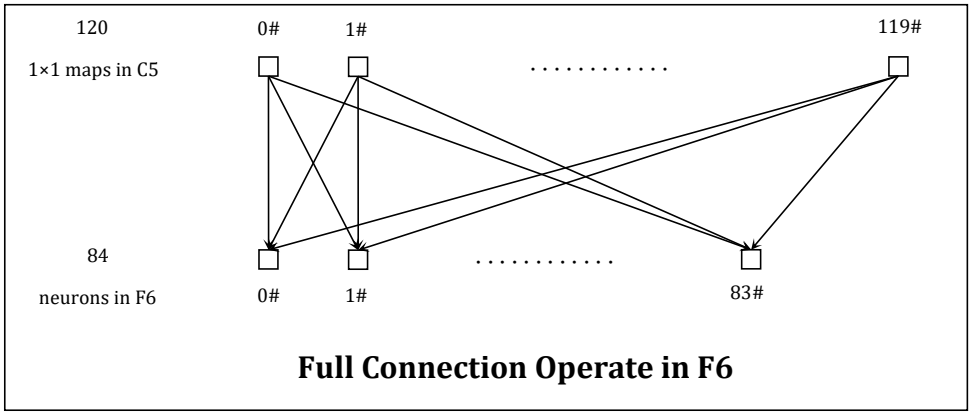
F6 层为全连接层，有 84 个神经元，每个 F6 层神经元从 C5 层所有 120 个神经元获取输入数据，共有 10164 个可训练参数。

F6 层 neuron 按下式计算，

$$f6_i = f\left(\sum_{j=0}^{119} w6_{ij} \cdot c5_{00}^{(j)} + b6_i\right) \qquad i = 0,\cdots,83$$

F6 层有 84 个 neurons，120×84 个 $w6_{ij}$ 权值，84 个 $b6_i$ 偏差。

F6 层功能解读：这一层将前层的 120 个神经元输出值按全连接方式得到本层的 84 个单元。



Layer Output

原 LeNet-5 的 F6 层选用 RBF 全连接网，后接高斯连接做 10 分类。

Output 层功能解读：这一层实现最终的分类功能，接 10 分类器，输入元 84 个，输出元 10 个，实现输出类别 10 种。

目前分类器大多采用 softmax 分类输出层或 MLP 分类层。比如用于手写数字图片分类的网络，最终有 10 个 neurons，分别表示 10 个数字。

本教程不完全按照原 LeNet-5 的模式讲解，在 F6 层之后改为两个层次，分别为 F7 层和 softmax 层，F7 层输入元 84 个，输出元 10 个，softmax 分类层输入元 10 个，输出元 10 个。

Layer F7

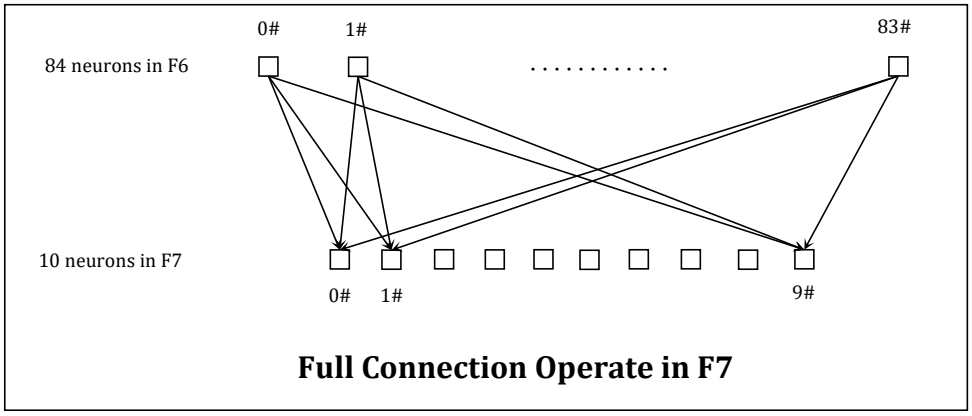
F7 层为全连接层，有 10 个神经元，每个 F7 层神经元从 F6 层所有 84 个神经元获取输入数据。

F7 层 neuron 按下式计算，

$$f7_i = f\left(\sum_{j=0}^{83} w7_{ij} \cdot f6_j + b7_i\right) \qquad i = 0,\cdots,9$$

F7 层有 10 个 neurons，84×10 个 $w7_{ij}$ 权值，10 个 $b7_i$ 偏差，共有 850 个可训练参数。

F7 层功能解读：这一层将前层的 84 个神经元输出值按全连接方式得到本层的 10 个单元。

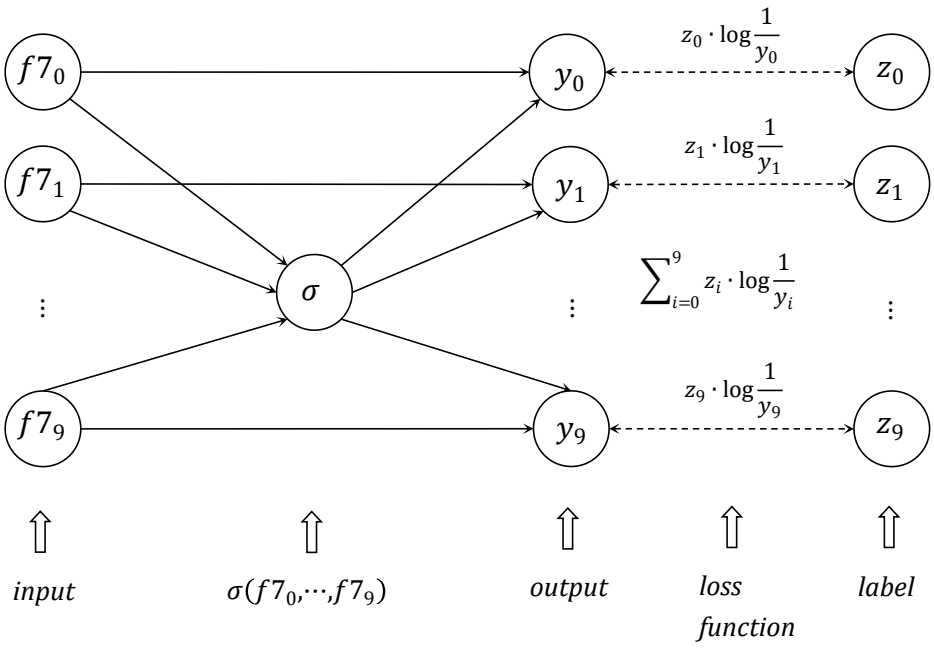


Full Connection Operate in F7

Layer Softmax

现在回顾一下前面的网络运行概图，Figure : Architecture of LeNet5 (training)，从输入层开始的一张 32×32 原始输入图片，到 F7 层已经得到了 10 维输出向量 $\{f7_0,\cdots,f7_9\}$，好像可以与 one-hot 型的 10 维标签相对比了，但是 F7 层的 10 维输出向量不满足归一化条件，还要经过一个归一化层，就是 softmax 层，其输出满足归一化条件。

用于 10 分类的数字识别问题的 Softmax 层可以设计为 10 入 10 出的 Softmax Net，如图所示，



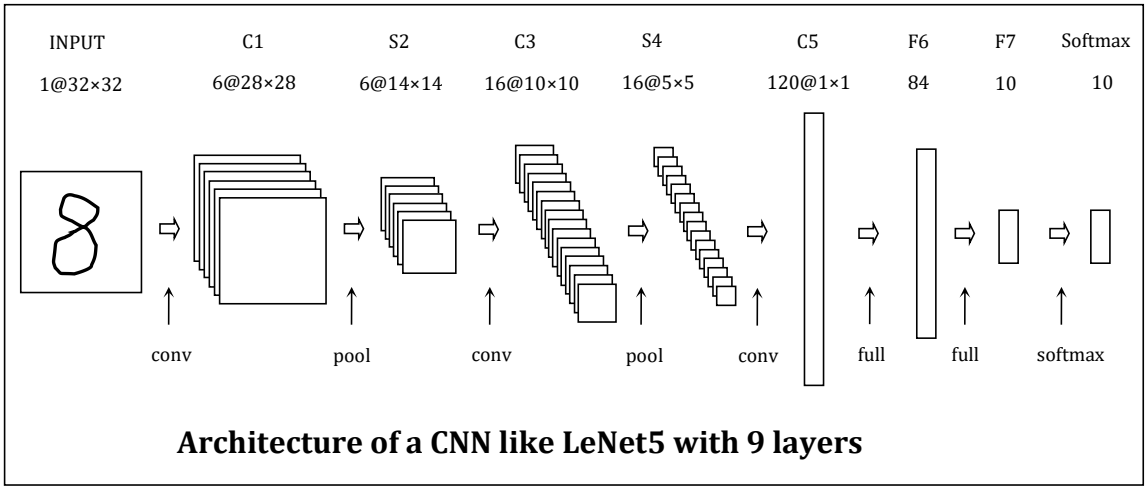
Architecture of Softmax Net

输入为 F7 层的 10 个神经元值 $\{f7_0,\cdots,f7_9\}$，输出为 Softmax 输出 $\{y_0,\cdots,y_9\}$，也是整个网络的输出，

$$\begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix} = \sigma(f7_0,\cdots,f7_9) \cdot \begin{bmatrix} e^{f7_0} \\ \vdots \\ e^{f7_9} \end{bmatrix} = \frac{1}{\sum_{j=0}^{9} e^{f7_j}} \cdot \begin{bmatrix} e^{f7_0} \\ \vdots \\ e^{f7_9} \end{bmatrix}$$

Architecture of this Course

到此为止，本教程的类 LeNet5 的多层卷积网络结构就完整了。



Architecture of a CNN like LeNet5 with 9 layers

Target Supervisor Labels

目标向量，或称监督向量、标签向量，为 $\{z_0,\cdots,z_9\}$，表示为与输入 image 图片的配对形式，$\left\{\left[x_{ij}^{(k)}\right]_{32\times32},\left[z_i^{(k)}\right]_{10\times1}\right\}$，简记为 $\left\{x^{(k)},z^{(k)}\right\}$，上标 (k) 表示第 k 对训练数据。如果用于 10 分类的数字识别问题，标签向量可取值：

$$z^{(k)}\in\left\{\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}\right\}$$

分别表示训练样本的真实标签（ground truth）为 0，1，2，3，4，5，6，7，8，9，简记为

$$z^{(k)}\in\{\vec{e}_0,\vec{e}_1,\vec{e}_2,\vec{e}_3,\vec{e}_4,\vec{e}_5,\vec{e}_6,\vec{e}_7,\vec{e}_8,\vec{e}_9\}$$

Cost function

对于一个训练样本及其标签 $\left\{x^{(k)},z^{(k)}\right\}$，衡量整个网络的输出向量 $\{y_0,\cdots,y_9\}$ 与标签向量 $\{z_0,\cdots,z_9\}$ 之间的差值的函数称为**代价函数**。

这个代价函数要达到一种效果，当输出向量与标签向量之间越接近，说明网络对当前输入样本的分类判断越正确，代价函数值要越小；反之，当输出向量与标签向量之间差距越大，网络对当前输入样本的分类判断越错误，代价函数值就要越大，而网络学习的目的就是要通过修改网络权值，从而降低这个代价值，使得网络输出尽量靠近对应的标签。

假设整个网络所有的可训练权值整合简记为参数 $\theta = (\theta_1,\cdots,\theta_N)$，所有的训练样本及其标签整合简记为 $s = \left(s^{(1)},\cdots,s^{(M)}\right)$，$s^{(k)} = \left\{x^{(k)},z^{(k)}\right\},\ k = 1,\cdots,M$，$s^{(k)}$ 表示第 k 对训练样本（训练数据），M 表示训练样本的总数，则针对一个样本的代价函数可以记为 $J\left(s^{(k)};\theta\right)$。

Gradient Descent 梯度下降算法

We want to choose θ so as to minimize $J\left(s^{(k)};\theta\right)$.

神经网络训练的目的就是选择合适的 θ 使得代价函数 $J\left(s^{(k)};\theta\right)$ 取得综合极小。所谓综合极小，是个平均概念，不见得每个 $J\left(s^{(k)};\theta\right)$ 均取得极小，但平均而言，总的代价函数取极小，取决于训练数据集 s 是如何选取的，训练方案是如何设计的，等等。

To do so, let's use a search algorithm that starts with some "initial guess" for θ, and that repeatedly changes θ to make $J\left(s^{(k)};\theta\right)$ smaller, until hopefully we converge to a value of θ that minimizes $J\left(s^{(k)};\theta\right)$.

梯度下降算法的基本思想：先猜测一个初始的参数 θ，然后不断改变 θ，使得 $J\left(s^{(k)};\theta\right)$ 逐渐变小，直到满足收敛条件。

Specifically, let's consider the **gradient descent algorithm**, which starts with some initial θ, and repeatedly performs the update:

$$\theta_j :=\ \theta_j - \alpha \cdot \frac{\partial J\left(s^{(k)};\theta\right)}{\partial \theta_j} \qquad j = 1,\cdots,N$$

This update is simultaneously performed for all values of j. Here, α is called the **learning rate**. This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of $J(s;\theta)$.

$\alpha > 0$ 称为**学习速率 learning rate**，代价函数 $J\left(s^{(k)};\theta\right)$ 相对于个体参数 θ_j 的导数 $\frac{\partial J\left(s^{(k)};\theta\right)}{\partial \theta_j}$ 称为**梯度 gradient**，这个公式称为**学习公式**，这个方法称为**最陡方向下降法**。

注意一个基本事实：

$$if \quad \frac{\partial J\left(s^{(k)};\theta\right)}{\partial \theta_j} > 0 \quad then \quad \begin{cases} J\left(s^{(k)};\theta\right)\uparrow & when\ \ \theta_j \uparrow \\ J\left(s^{(k)};\theta\right)\downarrow & when\ \ \theta_j \downarrow \end{cases}$$

$$if \quad \frac{\partial J\left(s^{(k)};\theta\right)}{\partial \theta_j} < 0 \quad then \quad \begin{cases} J\left(s^{(k)};\theta\right)\downarrow & when\ \ \theta_j \uparrow \\ J\left(s^{(k)};\theta\right)\uparrow & when\ \ \theta_j \downarrow \end{cases}$$

再来理解 $\theta_j: = \theta_j - \alpha \cdot \frac{\partial J(s^{(k)};\theta)}{\partial \theta_j}$ ，就很清楚了。

Cross Entropy Loss Function

交叉熵代价函数（cross entropy loss function）是与 Softmax 网络相匹配的代价函数。

对于一个训练样本及其标签 $s^{(k)} = \{x^{(k)},z^{(k)}\},\ k = 1,\cdots,M$，整个网络的输出向量为 $\{y_0,\cdots,y_9\}$，标签向量为 $\{z_0,\cdots,z_9\}$，均满足归一化条件 $\sum_{0=1}^{9} y_i = 1$，$\sum_{i=0}^{9} z_i = 1$，定义交叉熵代价函数 $J(s^{(k)};\theta)$ 为

$$J(s^{(k)};\theta) = -\sum_{i=0}^{9} z_i \cdot \log y_i = \sum_{i=0}^{9} z_i \cdot \log\frac{1}{y_i}$$

为了表述方便，后面省略 $s^{(k)}$，简记为 $J(\theta)$，但是要谨记，代价函数的计算是针对一个一个训练样本的。

基于信息熵的解释

因为有归一化条件 $\sum_{0=1}^{9} y_i = 1$，$\sum_{i=0}^{9} z_i = 1$，则可以将网络输出记为随机变量 Y 的分布 $\{y_0,\cdots,y_9\}$，将监督标签记为随机变量 Z 的分布 $\{z_0,\cdots,z_9\}$。

根据交叉熵的定义和性质，代价函数 $J(\theta)$ 相当于两项之和，即网络输出 Y 与监督标签 Z 的差异度信息 $H_{KL}(Z||Y)$（**KL 散度**）加上监督标签 Z 的**信息** $H(Z)$。

$$J(\theta) = \sum_{i=0}^{9} z_i \cdot \log\frac{1}{y_i} = H_{cross}(Z||Y) = H_{KL}(Z||Y) + H(Z)$$

相对熵（Kullback-Leibler 散度）$H_{KL}(Z||Y)$ 表示和之间的差异度，

$$H_{KL}(Z||Y) = \sum_{i=0}^{9} z_i \cdot log\left(\frac{z_i}{y_i}\right)$$

相对熵比较两个分布不同的程度，相对熵 $H_{KL}(Z||Y) \geq 0$，当两个分布完全一致时，$z_i = y_i, i = 0,\cdots,9$，有 $H_{KL}(Z||Y) = 0$。

离散随机变量 Z 的信息熵 $H(Z)$ 定义为

$$H(Z) = \sum_{i=0}^{9} z_i \cdot log\left(\frac{1}{z_i}\right) > 0$$

$H(Z)=0$ 当且仅当 $\{z_0,\cdots,z_9\}$ 为 one-hot 型（只有一个分量为 1，其它为 0），即确定性分布，对应确定性的 Z。$H(Z)=1$ 当且仅当 Z 的分布为均匀分布 $\begin{cases} i=0,\cdots,9 \\ z_i=\frac{1}{10} \end{cases}$。

网络的学习过程就是不断地减小代价函数 $J(\theta)$ 即减小差异度信息 $H_{KL}(Z||Y)$ 的过程。

对于 one-hot 型监督标签，其信息 $H(Z)=0$，表示确定性的信息，则网络的学习过程就是不断地减小网络输出 Y 的信息量 $H(Y)$，并将其分布 $\{y_1,\cdots,y_n\}$ 逐步收敛到 one-hot 型监督标签 Z 的分布 $\{z_1,\cdots,z_n\}$ 的过程。

梯度计算的基本原理

从学习公式

$$\theta_j:=\theta_j-\alpha\cdot\frac{\partial J(\theta)}{\partial\theta_j}\qquad j=1,\cdots,N$$

可知，网络训练的关键是求出代价函数 $J(\theta)$ 相对于每个可训练参数（权值）θ_j 的偏导数 $\frac{\partial J(\theta)}{\partial\theta_j}$，方法就是链式求导法则（chain rule）。

实际上，代价函数 $J(\theta)$ 相对于网络中的每个量（不一定都是可训练的）的偏导数都可以称为梯度（gradient），习惯上，gradient 也可以称为敏感值（sensitivity），意为网络中的某个量对代价函数 $J(\theta)$ 的影响力。后面统一用 δ 表示 sensitivity。

比如，定义 $\delta(u_A)$ 为代价函数 $J(\theta)$ 相对于神经元 neuron A 的激活函数 $f(\cdot)$ 的输入 u_A 的导数：

$$\delta(u_A)=\frac{\partial J(\theta)}{\partial u_A}$$

称其为神经元 neuron A 的**输入敏感值**。神经网络中的每个神经元均有相应的输入敏感值。u_A 实际上就是激活函数括号里面的内容，比如，F7 层 neuron 按下式计算，

$$f7_i=f\left(\sum_{j=0}^{83}w7_{ij}\cdot f6_j+b7_i\right)\qquad i=0,\cdots,9$$

则有

$$\delta(u_{f7_i})=\frac{\partial J(\theta)}{\partial u_{f7_i}}\qquad u_{f7_i}=\sum_{j=0}^{83}w7_{ij}\cdot f6_j+b7_i\qquad i=0,\cdots,9$$

为什么要单独定义和计算神经元的输入敏感值？如果每个神经元的输入敏感值已经计算得到了，则可以由此直接求出相关的权值梯度值（权值敏感值），比如，

$$\delta(w7_{ij}) = \frac{\partial J(\theta)}{\partial w7_{ij}} = \frac{\partial J(\theta)}{\partial u_{f7_i}} \cdot \frac{\partial u_{f7_i}}{\partial w7_{ij}} = \delta(u_{f7_i}) \cdot f6_j$$

神经网络通过神经元的连接关系逐层向前传播激活值（activation forward），而每个神经元的输入敏感值 δ 也通过神经元的反向连接关系逐层向后传播（sensitivity backward），其计算服从链式求导法则（chain rule），统称为反向传播算法（Back Propagation Algorithm）。

Gradient in Softmax Net $\delta(f7_i)$

在 Softmax Net 中，代价函数为

$$J(\theta) = -\sum_{i=0}^{9} z_i \cdot \log y_i = \sum_{i=0}^{9} z_i \cdot \log \frac{1}{y_i}$$

求 Softmax Net 的输入值 $f7_i$ 对应的**敏感值** $\delta(f7_i)$，该值实为代价函数 $J(\theta)$ 相对于输入值 $f7_i$ 的导数，

$$\delta(f7_i) = \frac{\partial J(\theta)}{\partial (f7_i)} = -(z_i - y_i) \qquad i = 0,\cdots,9$$

证明：

根据 Softmax Net 的网络图结构，应用反向 chain rules，依次求出梯度值 $\frac{\partial J(\theta)}{\partial (y_i)}$，$\frac{\partial J(\theta)}{\partial \sigma}$，$\frac{\partial J(\theta)}{\partial (f7_i)}$，

$$\frac{\partial J(\theta)}{\partial (y_i)} = -\frac{z_i}{y_i} \qquad i = 0,\cdots,9$$

$$\frac{\partial J(\theta)}{\partial \sigma} = \sum_{j=0}^{9} \frac{\partial J(\theta)}{\partial (y_j)} \cdot \frac{\partial (y_j)}{\partial \sigma} = \sum_{j=9}^{9} \left(-\frac{z_j}{y_j}\right) \cdot e^{f7_j} = \frac{-1}{\sigma} \sum_{j=0}^{9} z_j$$

$$note: \qquad y_i = \sigma(f7_1,\cdots,f7_n) \cdot e^{f7_i} = \frac{e^{f7_i}}{\sum_{j=0}^{9} e^{f7_j}} \qquad i = 0,\cdots,9$$

$$\frac{\partial J(\theta)}{\partial (f7_i)} = \frac{\partial J(\theta)}{\partial (y_i)} \cdot \frac{\partial (y_i)}{\partial (f7_i)} + \frac{\partial J(\theta)}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial (f7_i)} \qquad i = 0,\cdots,9$$

$$\frac{\partial J(\theta)}{\partial (f7_i)} = \left(-\frac{z_i}{y_i}\right) \cdot \left(\sigma \cdot e^{f7_i}\right) + \left(\frac{-1}{\sigma} \sum_{j=0}^{9} z_j\right) \cdot \left(\frac{-e^{f7_i}}{\left(\sum_{j=0}^{9} e^{f7_j}\right)^2}\right) \qquad i = 0,\cdots,9$$

$$note: \qquad \sigma = \frac{1}{\sum_{j=0}^{9} e^{f7_j}} \qquad \Rightarrow \qquad \frac{\partial \sigma}{\partial (f7_i)} = \frac{-\, e^{f7_i}}{\left(\sum_{j=0}^{9} e^{f7_j}\right)^2} \qquad i = 0,\cdots,9$$

$$\frac{\partial J(\theta)}{\partial (f7_i)} = (-\, z_i) + \left(\frac{1}{\sigma}\sum_{j=0}^{9} z_j\right) \cdot \left(\sigma^2 \cdot e^{f7_i}\right) \qquad i = 0,\cdots,9$$

$$note: \qquad y_i = \sigma \cdot e^{f7_i} \qquad i = 0,\cdots,9$$

$$\frac{\partial J(\theta)}{\partial (f7_i)} = (-\, z_i) + \left(\sum_{j=0}^{9} z_j\right) \cdot (y_i) \qquad i = 0,\cdots,9$$

$$note: \qquad \sum_{j=0}^{9} z_j = 1$$

$$\delta(f7_i) = \frac{\partial J(\theta)}{\partial (f7_i)} = -\,(z_i - y_i) \qquad i = 0,\cdots,9$$

Gradient in Layer F7

各层提要：在每一层，都要计算 4 种敏感值，
（1）本层神经元的激活函数括号里面的输入值的敏感值 $\boldsymbol{\delta(u)}$，
（2）可学习权值参数 $\boldsymbol{w}$ 的敏感值 $\boldsymbol{\delta(w)}$，
（3）可学习偏差参数 $\boldsymbol{b}$ 的敏感值 $\boldsymbol{\delta(b)}$，
（4）前一层输出值即本层输入值的敏感值 $\boldsymbol{\delta(f)}$。

F7 层为全连接层，有 10 个神经元，每个 F7 层神经元从 F6 层所有 84 个神经元获取输入数据，按下式计算，

$$f7_i = f\left(\sum_{j=0}^{83} w7_{ij} \cdot f6_j + b7_i\right) = f\left(u_{f7_i}\right) \qquad i = 0,\cdots,9$$

$$u_{f7_i} = \sum_{j=0}^{83} w7_{ij} \cdot f6_j + b7_i \qquad i = 0,\cdots,9$$

（1）每个 F7 层神经元的激活函数括号里面的输入值 u_{f7_i} 的敏感值 $\delta\left(u_{f7_i}\right)$，

$$\delta\left(u_{f7_i}\right) = \frac{\partial J(\theta)}{\partial\left(u_{f7_i}\right)} = \frac{\partial J(\theta)}{\partial (f7_i)} \cdot \frac{\partial (f7_i)}{\partial\left(u_{f7_i}\right)} = \delta(f7_i) \cdot f'\left(u_{f7_i}\right) \qquad i = 0,\cdots,9$$

注记：注意到，u_{f7_i} 与 $f7_i$ 是一对一的，依据链式求导法则（chain rule），求导链是单一链，即可以表达为如下单链关系图，

$$u_{f7_i} \overset{1\ to\ 1}{\longleftrightarrow} f7_i \leftrightarrow J(\theta) \qquad i = 0,\cdots,9$$

（2）可学习权值参数 $w7_{ij}$ 的敏感值 $\delta(w7_{ij})$，求导链是单一链，

$$\delta(w7_{ij}) = \frac{\partial J(\theta)}{\partial(w7_{ij})} = \frac{\partial J(\theta)}{\partial(u_{f7_i})} \cdot \frac{\partial(u_{f7_i})}{\partial(w7_{ij})} = \delta(u_{f7_i}) \cdot f6_j \qquad \begin{cases} i = 0,\cdots,9 \\ j = 0,\cdots,83 \end{cases}$$

（3）可学习偏差参数 $b7_i$ 的敏感值 $\delta(b7_i)$，求导链是单一链，

$$\delta(b7_i) = \frac{\partial J(\theta)}{\partial(b7_i)} = \frac{\partial J(\theta)}{\partial(u_{f7_i})} \cdot \frac{\partial(u_{f7_i})}{\partial(b7_i)} = \delta(u_{f7_i}) \cdot 1 \qquad i = 0,\cdots,9$$

注记：对于全连接层（full connection），权值参数和偏差参数不是共享的，所以求导链都是单一链。

注记：权值 $w7_{ij}$ 与偏差 $b7_i$ 是可学习参数，敏感值 $\delta(w7_{ij})$ 和 $\delta(b7_i)$ 就是学习公式中的 $\frac{\partial J(\theta)}{\partial \theta_j}$，

$$\theta_j := \theta_j - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta_j} \qquad j = 1,\cdots,N$$

（4）前一层输出值即本层输入值 $f6_j$ 的敏感值 $\delta(f6_j)$，求解 $\delta(f6_j)$ 时要注意到，每个 $f6_j$ 均与 10 个 u_{f7_i}, $i = 0,\cdots,9$ 有关，求导链是并行多链，

$$f6_j \overset{1\ to\ 10}{\longleftrightarrow} \{u_{f7_i}\}_{i=0,\cdots,9} \leftrightarrow J(\theta) \qquad j = 0,\cdots,83$$

所以有如下求和式，

$$\delta(f6_j) = \frac{\partial J(\theta)}{\partial(f6_j)} = \sum_{i=0}^{9} \frac{\partial J(\theta)}{\partial(u_{f7_i})} \cdot \frac{\partial(u_{f7_i})}{\partial(f6_j)} = \sum_{i=0}^{9} \delta(u_{f7_i}) \cdot w7_{ij} \qquad j = 0,\cdots,83$$

注意到这个表达式，这种**多通道模式**是敏感值按照反向网络连接进行反向传播的主要模式。

Gradient in Layer F6

F6 层为全连接层，有 84 个神经元，每个 F6 层神经元从 C5 层所有 120 个神经元获取输入数据，按下式计算，

$$f6_i = f\left(\sum_{j=0}^{119} w6_{ij} \cdot c5_{00}^{(j)} + b6_i\right) = f\left(u_{f6_i}\right) \qquad i = 0,\cdots,83$$

$$u_{f6_i} = \sum_{j=0}^{119} w6_{ij} \cdot c5_{00}^{(j)} + b6_i \qquad i = 0,\cdots,83$$

依次求得各个敏感值。

（1）每个 F6 层神经元的激活函数括号里面的输入值 u_{f6_i} 的敏感值 $\delta\left(u_{f6_i}\right)$，求导链是单一链，

$$\delta\left(u_{f6_i}\right) = \frac{\partial J(\theta)}{\partial\left(u_{f6_i}\right)} = \frac{\partial J(\theta)}{\partial(f6_i)} \cdot \frac{\partial(f6_i)}{\partial\left(u_{f6_i}\right)} = \delta(f6_i) \cdot f'\left(u_{f6_i}\right) \qquad i = 0,\cdots,83$$

（2）可学习权值参数 $w6_{ij}$ 的敏感值 $\delta\left(w6_{ij}\right)$，求导链是单一链，

$$\delta\left(w6_{ij}\right) = \frac{\partial J(\theta)}{\partial\left(w6_{ij}\right)} = \frac{\partial J(\theta)}{\partial\left(u_{f6_i}\right)} \cdot \frac{\partial\left(u_{f6_i}\right)}{\partial\left(w6_{ij}\right)} = \delta\left(u_{f6_i}\right) \cdot c5_{00}^{(j)} \qquad \begin{cases} i = 0,\cdots,83 \\ j = 0,\cdots,119 \end{cases}$$

（3）可学习偏差参数 $b6_i$ 的敏感值 $\delta(b6_i)$，求导链是单一链，

$$\delta(b6_i) = \frac{\partial J(\theta)}{\partial(b6_i)} = \frac{\partial J(\theta)}{\partial\left(u_{f6_i}\right)} \cdot \frac{\partial\left(u_{f6_i}\right)}{\partial(b6_i)} = \delta\left(u_{f6_i}\right) \cdot 1 \qquad i = 0,\cdots,83$$

（4）前一层输出值即本层输入值 $c5_{00}^{(j)}$ 的敏感值 $\delta\left(c5_{00}^{(j)}\right)$，求解 $\delta\left(c5_{00}^{(j)}\right)$ 时要注意到，每个 $c5_{00}^{(j)}$ 均与 84 个 u_{f6_i}, $i = 0,\cdots,83$ 有关，求导链是并行多链，

$$c5_{00}^{(j)} \quad \overset{1\ to\ 84}{\longleftrightarrow} \quad \left\{u_{f6_i}\right\}_{i=0,\cdots,83} \quad \leftrightarrow \quad J(\theta) \qquad j = 0,\cdots,119$$

所以有如下求和式，

$$\delta\left(c5_{00}^{(j)}\right) = \frac{\partial J(\theta)}{\partial\left(c5_{00}^{(j)}\right)} = \sum_{i=0}^{83} \frac{\partial J(\theta)}{\partial\left(u_{f6_i}\right)} \cdot \frac{\partial\left(u_{f6_i}\right)}{\partial\left(c5_{00}^{(j)}\right)} = \sum_{i=0}^{83} \delta\left(u_{f6_i}\right) \cdot w6_{ij} \qquad j = 0,\cdots,119$$

<u>Gradient in Layer C5</u>

C5 层 neuron 按如下式计算，

$$c5_{ij}^{(l)} = f\left(\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2} w5_{u,v}^{(m,l)} \cdot s4_{i+2+u,j+2+v}^{(m)} + b5^{(l)}\right) = f\left(u_{c5_{ij}^{(l)}}\right) \qquad \begin{cases} i,j = 0 \\ l = 0,\cdots,119 \end{cases}$$

简化为

$$c5_{00}^{(l)}=f\left(\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w5_{u,v}^{(m,l)}\cdot s4_{2+u,2+v}^{(m)}+b5^{(l)}\right)=f\left(u_{c5_{00}^{(l)}}\right)\qquad l=0,\cdots,119$$

其激活函数的输入部分为 $u_{c5_{00}^{(l)}}$

$$u_{c5_{00}^{(l)}}=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w5_{u,v}^{(m,l)}\cdot s4_{2+u,2+v}^{(m)}+b5^{(l)}\qquad l=0,\cdots,119$$

（1）每个 C5 层神经元的激活函数括号里面的输入值 $u_{c5_{00}^{(l)}}$ 的敏感值 $\delta\left(u_{c5_{00}^{(l)}}\right)$，求导链是单一链，

$$\delta\left(u_{c5_{00}^{(l)}}\right)=\frac{\partial J(\theta)}{\partial\left(u_{c5_{00}^{(l)}}\right)}=\frac{\partial J(\theta)}{\partial\left(c5_{00}^{(l)}\right)}\cdot\frac{\partial\left(c5_{00}^{(l)}\right)}{\partial\left(u_{c5_{00}^{(l)}}\right)}=\delta\left(c5_{00}^{(l)}\right)\cdot f'\left(u_{c5_{00}^{(l)}}\right)\qquad l=0,\cdots,119$$

（2）则该层参数 $w5_{u,v}^{(m,l)},\begin{cases}m=0,\cdots,15\\l=0,\cdots,119\\u,v=-2,-1,0,1,2\end{cases}$ 的梯度为（求导链是单一链）

$$\delta\left(w5_{u,v}^{(m,l)}\right)=\frac{\partial J(\theta)}{\partial\left(w5_{u,v}^{(m,l)}\right)}=\frac{\partial J(\theta)}{\partial\left(u_{c5_{00}^{(l)}}\right)}\cdot\frac{\partial\left(u_{c5_{00}^{(l)}}\right)}{\partial\left(w5_{u,v}^{(m,l)}\right)}=\delta\left(u_{c5_{00}^{(l)}}\right)\cdot s4_{2+u,2+v}^{(m)}\qquad\begin{cases}m=0,\cdots,15\\l=0,\cdots,119\\u,v=-2,-1,0,1,2\end{cases}$$

（3）则该层参数 $\left\{b5^{(l)},l=0,\cdots,119\right\}$ 的梯度为（求导链是单一链）

$$\delta\left(b5^{(l)}\right)=\frac{\partial J(\theta)}{\partial\left(b5^{(l)}\right)}=\frac{\partial J(\theta)}{\partial\left(u_{c5_{00}^{(l)}}\right)}\cdot\frac{\partial\left(u_{c5_{00}^{(l)}}\right)}{\partial\left(b5^{(l)}\right)}=\delta\left(u_{c5_{00}^{(l)}}\right)\qquad l=0,\cdots,119$$

注记：C5 层是卷积层，卷积层的卷积核（filter）原则上是共享重复使用的，但是 C5 层的输入特征图与卷积核的尺寸是一样的，都是 5×5 窗口，所以每个卷积核及其权值和偏差都只使用了 1 次，所以其求导链都是单一链。

（4）C5 层的输入即 S4 层的输出 $s4_{ij}^{(m)},\begin{cases}i,j=0,\cdots,4\\m=0,\cdots,15\end{cases}$，其敏感值 $\delta\left(s4_{ij}^{(m)}\right)$ 为

$$\delta\left(s4_{ij}^{(m)}\right)=\frac{\partial J(\theta)}{\partial\left(s4_{ij}^{(m)}\right)}=\sum_{l=0}^{119}\delta\left(u_{c5_{00}^{(l)}}\right)\cdot w5_{i-2,j-2}^{(m,l)}\qquad\begin{cases}i,j=0,\cdots,4\\m=0,\cdots,15\end{cases}$$

证明：注意到从 S4 到 C5 的前向卷积公式为

$$c5_{00}^{(l)}=f\left(\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w5_{u,v}^{(m,l)}\cdot s4_{2+u,2+v}^{(m)}+b5^{(l)}\right)=f\left(u_{c5_{00}^{(l)}}\right)\qquad l=0,\cdots,119$$

$$u_{c5_{00}^{(l)}}=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w5_{u,v}^{(m,l)}\cdot s4_{2+u,2+v}^{(m)}+b5^{(l)}\qquad l=0,\cdots,119$$

由此两个公式，可以得到前向关系图，

$$\left\{s4_{2+u,2+v}^{(m)}\right\}_{\substack{m=0,\cdots,15\\u=-2,-1,0,1,2\\v=-2,-1,0,1,2}}\xrightarrow{400\ to\ 400}\left\{w5_{u,v}^{(m,l)}\right\}_{\substack{m=0,\cdots,15\\u=-2,-1,0,1,2\\v=-2,-1,0,1,2}}\xrightarrow{400\ to\ 1}\left\{u_{c5_{00}^{(l)}}\right\}\qquad l=0,\cdots,119$$

做下标变换，$2+u=i,2+v=j$，

$$\begin{cases}2+u=i\\2+v=j\end{cases}\qquad\begin{cases}u=-2,-1,0,1,2\\v=-2,-1,0,1,2\end{cases}\qquad\begin{cases}i=0,1,2,3,4\\j=0,1,2,3,4\end{cases}$$

得到等价的前向关系图，

$$\left\{s4_{i,j}^{(m)}\right\}_{\substack{m=0,\cdots,15\\i=0,1,2,3,4\\j=0,1,2,3,4}}\xrightarrow{400\ to\ 400}\left\{w5_{i-2,j-2}^{(m,l)}\right\}_{\substack{m=0,\cdots,15\\i=0,1,2,3,4\\j=0,1,2,3,4}}\xrightarrow{400\ to\ 1}\left\{u_{c5_{00}^{(l)}}\right\}\qquad l=0,\cdots,119$$

推出反向关系图，

$$\left\{u_{c5_{00}^{(l)}}\right\}_{l=0,\cdots,119}\xrightarrow{120\ to\ 120}\left\{w5_{i-2,j-2}^{(m,l)}\right\}_{l=0,\cdots,119}\xrightarrow{120\ to\ 1}\left\{s4_{i,j}^{(m)}\right\}\qquad\begin{cases}m=0,\cdots,15\\i=0,1,2,3,4\\j=0,1,2,3,4\end{cases}$$

从这两个公式以及结构图（**Conv Operate in C5**）以及反向关系图中可知，每个 $s4_{ij}^{(m)},\begin{cases}i,j=0,\cdots,4\\m=0,\cdots,15\end{cases}$ 与 120 个 $c5_{00}^{(l)},l=0,\cdots,119$ 有关联，$c5_{00}^{(l)}$ 与 $u_{c5_{00}^{(l)}}$ 是一一对应的，即每个 $s4_{ij}^{(m)}$ 与 120 个 $u_{c5_{00}^{(l)}},l=0,\cdots,119$ 有关联，这个关联对应的权值为 $w5_{i-2,j-2}^{(m,l)},l=0,\cdots,119$。

也可以写出矩阵式进行直观比对，得到同样的结果。事实上，公式中的 $\left[w5_{u,v}^{(m,l)}\right]_{u,v=-2,-1,0,1,2}$ 如下

$$\begin{bmatrix} w5_{-2,-2}^{(m,l)} & w5_{-2,-1}^{(m,l)} & w5_{-2,0}^{(m,l)} & w5_{-2,1}^{(m,l)} & w5_{-2,2}^{(m,l)} \\ w5_{-1,-2}^{(m,l)} & w5_{-1,-1}^{(m,l)} & w5_{-1,0}^{(m,l)} & w5_{-1,1}^{(m,l)} & w5_{-1,2}^{(m,l)} \\ w5_{0,-2}^{(m,l)} & w5_{0,-1}^{(m,l)} & w5_{0,0}^{(m,l)} & w5_{0,1}^{(m,l)} & w5_{0,2}^{(m,l)} \\ w5_{1,-2}^{(m,l)} & w5_{1,-1}^{(m,l)} & w5_{1,0}^{(m,l)} & w5_{1,1}^{(m,l)} & w5_{1,2}^{(m,l)} \\ w5_{2,-2}^{(m,l)} & w5_{2,-1}^{(m,l)} & w5_{2,0}^{(m,l)} & w5_{2,1}^{(m,l)} & w5_{2,2}^{(m,l)} \end{bmatrix} \quad \begin{cases} m = 0,\cdots,15 \\ l = 0,\cdots,119 \end{cases}$$

其对应卷积的 $\left[s4_{2+u,2+v}^{(m)}\right]_{u,v=-2,-1,0,1,2}$，实际为 $\left[s4_{i,j}^{(m)}\right]_{i,j=0,1,2,3,4}$

$$\begin{bmatrix} s4_{0,0}^{(m)} & s4_{0,1}^{(m)} & s4_{0,2}^{(m)} & s4_{0,3}^{(m)} & s4_{0,4}^{(m)} \\ s4_{1,0}^{(m)} & s4_{1,1}^{(m)} & s4_{1,2}^{(m)} & s4_{1,3}^{(m)} & s4_{1,4}^{(m)} \\ s4_{2,0}^{(m)} & s4_{2,1}^{(m)} & s4_{2,2}^{(m)} & s4_{2,3}^{(m)} & s4_{2,4}^{(m)} \\ s4_{3,0}^{(m)} & s4_{3,1}^{(m)} & s4_{3,2}^{(m)} & s4_{3,3}^{(m)} & s4_{3,4}^{(m)} \\ s4_{4,0}^{(m)} & s4_{4,1}^{(m)} & s4_{4,2}^{(m)} & s4_{4,3}^{(m)} & s4_{4,4}^{(m)} \end{bmatrix} \quad m = 0,\cdots,15$$

将以上两个矩阵进行比对，即可知道每个 $s4_{ij}^{(m)}$ 与 120 个 $u_{c5_{00}^{(l)}}, l = 0,\cdots,119$ 有关联并且其对应的权值为 $w5_{i-2,j-2}^{(m,l)}$，即

$$\frac{\partial\left(u_{c5_{00}^{(l)}}\right)}{\partial\left(s4_{ij}^{(m)}\right)} = w5_{i-2,j-2}^{(m,l)} \qquad \begin{cases} m = 0,\cdots,15 \\ i,j = 0,\cdots,4 \\ l = 0,\cdots,119 \end{cases}$$

所以有

$$\delta\left(s4_{ij}^{(m)}\right) = \frac{\partial J(\theta)}{\partial\left(s4_{ij}^{(m)}\right)} = \sum_{l=0}^{119} \frac{\partial J(\theta)}{\partial\left(u_{c5_{00}^{(l)}}\right)} \cdot \frac{\partial\left(u_{c5_{00}^{(l)}}\right)}{\partial\left(s4_{ij}^{(m)}\right)} = \sum_{l=0}^{119} \delta\left(u_{c5_{00}^{(l)}}\right) \cdot w5_{i-2,j-2}^{(m,l)} \qquad \begin{cases} i,j = 0,\cdots,4 \\ m = 0,\cdots,15 \end{cases}$$

证明完成。

注记：（1）每个 $s4_{ij}^{(m)}$ 与 120 个 $u_{c5_{00}^{(l)}}$ 有关联，所以 $\delta\left(s4_{ij}^{(m)}\right)$ 一定是一个求和式。

注记：（2）求导式

$$\frac{\partial\left(u_{c5_{00}^{(l)}}\right)}{\partial\left(s4_{ij}^{(m)}\right)}=w5_{i-2,j-2}^{(m,l)} \qquad \begin{cases} m=0,\cdots,15 \\ i,j=0,\cdots,4 \\ l=0,\cdots,119 \end{cases}$$

并不能直接从关系式

$$u_{c5_{00}^{(l)}}=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w5_{u,v}^{(m,l)}\cdot s4_{2+u,2+v}^{(m)}+b5^{(l)} \qquad l=0,\cdots,119$$

中简单地按照求导法则而得到，需要通过结构图研究其实际的前向与反向连接关系而得到正确的结论。

Gradient in Layer S4

目前普遍认为，S4 层的 sub-sampling 操作就是一个池化操作（pooling），本节只给出 max-pooling 算法的反传公式。

S4 层将 C3 层的 16 个特征图分别缩小 1 倍，不涉及可学习参数，只有输入数据 $c3_{ij}^{(k)}$, $\begin{cases} i,j=0,\cdots,9 \\ k=0,\cdots,15 \end{cases}$ 和输出数据 $s4_{ij}^{(k)}$, $\begin{cases} i,j=0,\cdots,4 \\ k=0,\cdots,15 \end{cases}$。其中，$\delta\left(s4_{ij}^{(k)}\right)$ 已经在前一个小结（**Gradient in Layer C5**）计算出来，

$$\delta\left(s4_{ij}^{(k)}\right)=\frac{\partial J(\theta)}{\partial\left(s4_{ij}^{(k)}\right)}=\sum_{l=0}^{119}\delta\left(u_{c5_{00}^{(l)}}\right)\cdot w5_{i-2,j-2}^{(k,l)} \qquad \begin{cases} i,j=0,\cdots,4 \\ k=0,\cdots,15 \end{cases}$$

本小结（**Gradient in Layer S4**）只需要计算出 $\delta\left(c3_{ij}^{(k)}\right)=\frac{\partial J(\theta)}{\partial\left(c3_{ij}^{(k)}\right)}$ $\begin{cases} i,j=0,\cdots,9 \\ k=0,\cdots,15 \end{cases}$。

池化操作（pooling）操作大多采用 max-pooling 算法，

$$s4_{ij}^{(k)}=\max\left\{c3_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}} \qquad \begin{cases} i,j=0,\cdots,4 \\ k=0,\cdots,15 \end{cases}$$

则只对每四个 $c3_{2i+u,2j+v}^{(k)}$ $\begin{cases} u=0,1 \\ v=0,1 \end{cases}$ $\begin{cases} i,j=0,\cdots,4 \\ k=0,\cdots,15 \end{cases}$ 中最大的一个 $\max\left\{c3_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}}\begin{cases} i,j=0,\cdots,4 \\ k=0,\cdots,15 \end{cases}$ 有唯一的一个 $s4_{ij}^{(k)}$ 与其对应， 为此，制作一个与 $c3_{2i+u,2j+v}^{(k)}$ 一一对应的掩膜矩阵(mask matrix for max-pooling) $msk3_{2i+u,2j+v}^{(k)}$

$$msk3_{2i+u,2j+v}^{(k)}=\begin{cases}1 & c3_{2i+u,2j+v}^{(k)}=\max\left\{c3_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}} \\ 0 & c3_{2i+u,2j+v}^{(k)}\neq\max\left\{c3_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}}\end{cases} \qquad \begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}$$

另外，本小结需要计算出 $\delta\left(c3_{ij}^{(k)}\right)\ \begin{cases}i,j=0,\cdots,9\\k=0,\cdots,15\end{cases}$，也就是说，要修改 $s4_{ij}^{(k)},\begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}$ 的下标以适应左边，方法如下，

对每一个 $c3_{2i+u,2j+v}^{(k)}$ 有唯一的一个 $s4_{ij}^{(k)}$ 与其对应（严谨来说，是可能唯一的对应，因为 max-pooling 是从 4 个 $c3_{2i+u,2j+v}^{(k)}\ \begin{cases}u=0,1\\v=0,1\end{cases}$ 中挑一个最大的与 $s4_{ij}^{(k)}$ 对应），

$$c3_{2i+u,2j+v}^{(k)}\ \begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\\u,v=0,1\end{cases}\quad\rightarrow\quad s4_{ij}^{(k)}\ \begin{cases}i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}$$

相当于：

$$c3_{i,j}^{(k)}\ \begin{cases}i,j=0,\cdots,9\\k=0,\cdots,15\end{cases}\quad\rightarrow\quad s4_{\text{floor}\left(\frac{i}{2}\right),\text{floor}\left(\frac{j}{2}\right)}^{(k)}\ \begin{cases}i,j=0,\cdots,9\\k=0,\cdots,15\end{cases}$$

注意其中下标范围的变化，实质对应关系没有改变。掩膜矩阵也做相应修改为

$$msk3_{2i+u,2j+v}^{(k)}\ \begin{cases}u=0,1\\v=0,1\\i,j=0,\cdots,4\\k=0,\cdots,15\end{cases}\quad\rightarrow\quad msk3_{i,j}^{(k)}\ \begin{cases}i,j=0,\cdots,9\\k=0,\cdots,15\end{cases}$$

在做好上述掩膜矩阵对应关系与下标的变换后，就可以直接给出 $\delta\left(c3_{ij}^{(k)}\right)$ 的计算公式了，

$$\delta\left(c3_{ij}^{(k)}\right)=msk3_{i,j}^{(k)}\cdot\delta\left(s4_{\text{floor}\left(\frac{i}{2}\right),\text{floor}\left(\frac{j}{2}\right)}^{(k)}\right)\qquad\begin{cases}i,j=0,\cdots,9\\k=0,\cdots,15\end{cases}$$

Gradient in Layer C3

C3 层有 16 个 feature maps，每个 $\left\{c3_{ij}^{(m)}\in\left[c3_{ij}^{(m)}\right]_{10\times10},m=0,\cdots,15\right\}$ 就是一个 neuron，共有 10×10×16=1600 个 neurons。如下所列：

$$\begin{matrix}\left[c3_{ij}^{(0)}\right]_{10\times10} & \left[c3_{ij}^{(1)}\right]_{10\times10} & \left[c3_{ij}^{(2)}\right]_{10\times10} & \left[c3_{ij}^{(3)}\right]_{10\times10}\\ \left[c3_{ij}^{(4)}\right]_{10\times10} & \left[c3_{ij}^{(5)}\right]_{10\times10} & \left[c3_{ij}^{(6)}\right]_{10\times10} & \left[c3_{ij}^{(7)}\right]_{10\times10}\\ \left[c3_{ij}^{(8)}\right]_{10\times10} & \left[c3_{ij}^{(9)}\right]_{10\times10} & \left[c3_{ij}^{(10)}\right]_{10\times10} & \left[c3_{ij}^{(11)}\right]_{10\times10}\\ \left[c3_{ij}^{(12)}\right]_{10\times10} & \left[c3_{ij}^{(13)}\right]_{10\times10} & \left[c3_{ij}^{(14)}\right]_{10\times10} & \left[c3_{ij}^{(15)}\right]_{10\times10}\end{matrix}$$

C3 层 neuron 按如下式计算，

$$c3_{ij}^{(m)}=f\left(\sum_{k=0}^{5}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w3_{u,v}^{(k,m)}\cdot s2_{i+2+u,j+2+v}^{(k)}+b3^{(m)}\right)\qquad\begin{cases}i,j=0,\cdots,9\\m=0,\cdots,15\end{cases}$$

其激活函数的输入部分为 $u_{c3_{ij}^{(m)}}$

$$u_{c3_{ij}^{(m)}}=\sum_{k=0}^{5}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w3_{u,v}^{(k,m)}\cdot s2_{i+2+u,j+2+v}^{(k)}+b3^{(m)}\qquad\begin{cases}i,j=0,\cdots,9\\m=0,\cdots,15\end{cases}$$

（1）每个 C3 层神经元的激活函数括号里面的输入值 $u_{c3_{ij}^{(m)}}$ 的敏感值 $\delta\left(u_{c3_{ij}^{(m)}}\right)$，求导链是单一链，

$$\delta\left(u_{c3_{ij}^{(m)}}\right)=\frac{\partial J(\theta)}{\partial\left(u_{c3_{ij}^{(m)}}\right)}=\frac{\partial J(\theta)}{\partial\left(c3_{ij}^{(m)}\right)}\cdot\frac{\partial\left(c3_{ij}^{(m)}\right)}{\partial\left(u_{c3_{ij}^{(m)}}\right)}=\delta\left(c3_{ij}^{(m)}\right)\cdot f'\left(u_{c3_{ij}^{(m)}}\right)\qquad\begin{cases}i,j=0,\cdots,9\\m=0,\cdots,15\end{cases}$$

（2）下面求该层权值参数 $w3_{u,v}^{(k,m)},\begin{cases}k=0,\cdots,5\\m=0,\cdots,15\\u,v=-2,-1,0,1,2\end{cases}$ 的梯度，注意，这里卷积核是共用的，即每个 $w3_{u,v}^{(k,m)}$ 会使用 100 次（$i,j=0,\cdots,9$），或者说，每个 $w3_{u,v}^{(k,m)}$ 会对 100 个 $\left\{u_{c3_{ij}^{(m)}},i,j=0,\cdots,9\right\}$ 都有贡献，则有链式（并行多链）求导公式如下：

$$\delta\left(w3_{u,v}^{(k,m)}\right)=\frac{\partial J(\theta)}{\partial\left(w3_{u,v}^{(k,m)}\right)}=\sum_{i=0}^{9}\sum_{j=0}^{9}\frac{\partial J(\theta)}{\partial\left(u_{c3_{ij}^{(m)}}\right)}\cdot\frac{\partial\left(u_{c3_{ij}^{(m)}}\right)}{\partial\left(w3_{u,v}^{(k,m)}\right)}$$

$$\delta\left(w3_{u,v}^{(k,m)}\right)=\sum_{i=0}^{9}\sum_{j=0}^{9}\delta\left(u_{c3_{ij}^{(m)}}\right)\cdot s2_{i+2+u,j+2+v}^{(k)}\qquad\begin{cases}(k,m)\in Cbn\\m=0,\cdots,15\\k=0,\cdots,5\\u,v=-2,-1,0,1,2\end{cases}$$

注意，所有 96 个权值矩阵中 36 个 $(k,m)\notin Cbn$ 的权值矩阵，其分量元素值 $w3_{u,v}^{(k,m)}$ 都为 0，

$$w3_{u,v}^{(k,m)}=0 \qquad \begin{cases} k=0,\cdots,5 \\ m=0,\cdots,15 \end{cases} \qquad \begin{cases} u=-2,-1,0,1,2 \\ v=-2,-1,0,1,2 \end{cases} \qquad (k,m)\notin Cbn$$

则有

$$\delta\left(w3_{u,v}^{(k,m)}\right)=0 \qquad \begin{cases} (k,m)\notin Cbn \\ m=0,\cdots,15 \\ k=0,\cdots,5 \\ u,v=-2,-1,0,1,2 \end{cases}$$

（3）下面求该层偏差参数 $\{b3^{(m)},m=0,\cdots,15\}$ 的梯度，同理，截距 $b3^{(m)}$ 也是共用的，即每个 $b3^{(m)}$ 也会使用 100 次（$i,j=0,\cdots,9$），或者说，每个 $b3^{(m)}$ 会对 100 个 $\left\{u_{c3_{ij}^{(m)}},i,j=0,\cdots,9\right\}$ 都有贡献，则有链式（并行多链）求导公式如下：

$$\delta\left(b3^{(m)}\right)=\frac{\partial J(\theta)}{\partial\left(b3^{(m)}\right)}=\sum_{i=0}^{9}\sum_{j=0}^{9}\frac{\partial J(\theta)}{\partial\left(u_{c3_{ij}^{(m)}}\right)}\cdot\frac{\partial\left(u_{c3_{ij}^{(m)}}\right)}{\partial\left(b3^{(m)}\right)}=\sum_{i=0}^{9}\sum_{j=0}^{9}\delta\left(u_{c3_{ij}^{(m)}}\right)\cdot 1 \qquad m=0,\cdots,15$$

（4）下面求 C3 层的输入部分 $s2_{ij}^{(k)},\begin{cases} i,j=0,\cdots,13 \\ k=0,\cdots,5 \end{cases}$ 的敏感值 $\delta\left(s2_{ij}^{(k)}\right)$，

$$\delta\left(s2_{ij}^{(k)}\right)=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}\delta\left(u_{c3_{i-2+u,j-2+v}^{(m)}}\right)\cdot w3_{-u,-v}^{(k,m)} \qquad \begin{cases} i,j=0,\cdots,13 \\ k=0,\cdots,5 \end{cases}$$

证明：首先，需要理清 $s2_{ij}^{(k)},\begin{cases} i,j=0,\cdots,13 \\ k=0,\cdots,5 \end{cases}$ 与 $c3_{ij}^{(m)},\begin{cases} i,j=0,\cdots,9 \\ m=0,\cdots,15 \end{cases}$ 的对应关系链，以及关系链上对应的权值。

注记：这里比 C5 层要复杂，C5 层有 120 个特征图，但每个特征图只有一个元，向 S4 层的反向定位关系链比较容易确定。而 C3 层有 16 个特征图，每个特征图有 10×10=100 个元，向 S2 层的反向定位关系链要复杂一些。

注意到从 S2 到 C3 的前向卷积公式为

$$u_{c3_{ij}^{(m)}}=\sum_{k=0}^{5}\sum_{u=-2}^{2}\sum_{v=-2}^{2}w3_{u,v}^{(k,m)}\cdot s2_{i+2+u,j+2+v}^{(k)}+b3^{(m)} \qquad \begin{cases} i,j=0,\cdots,9 \\ m=0,\cdots,15 \end{cases}$$

此式表达的**正向传播关系图**为，此图为从 S2 到 C3 的多对一关系图，

$$\left\{s2_{i+2+u,j+2+v}^{(k)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\k=0,\cdots,5}} \xrightarrow{150\ to\ 150} \left\{w3_{u,v}^{(k,m)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\k=0,\cdots,5}} \xrightarrow{150\ to\ 1} \left\{u_{c3_{ij}^{(m)}}\right\} \qquad \begin{cases} i,j=0,\cdots,9 \\ m=0,\cdots,15 \end{cases}$$

左边表示 6 个 5×5 窗口，共 150 个元，与右边一个元的多一正向对应关系，而这样的多一对应关系共有 10×10×16=1600 个，为使 S2 层特征图下标为主下标，令位置下标变换

$$\begin{cases} i^{'}=i+2+u \\ j^{'}=j+2+v \end{cases} \Rightarrow \begin{cases} i^{'}-2-u=i \\ j^{'}-2-v=j \end{cases} \Rightarrow \begin{cases} i^{'}=0,\cdots,13 \\ j^{'}=0,\cdots,13 \end{cases}$$

前向公式变为

$$u_{c3_{i^{'}-2-u,j^{'}-2-v}^{(m)}}=\sum_{k=0}^{5}\sum_{u=-2}^{2}\sum_{v=-2}^{2} w3_{u,v}^{(k,m)}\cdot s2_{i^{'},j^{'}}^{(k)}+b3^{(m)} \qquad \begin{cases} i^{'},j^{'}=0,\cdots,13 \\ m=0,\cdots,15 \end{cases}$$

则正向传播关系图变为，此图仍为从 S2 到 C3 的多对一关系图，

$$\left\{s2_{i^{'},j^{'}}^{(k)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\k=0,\cdots,5}} \xrightarrow{150\ to\ 150} \left\{w3_{u,v}^{(k,m)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\k=0,\cdots,5}} \xrightarrow{150\ to\ 1} \left\{u_{c3_{i^{'}-2-u,j^{'}-2-v}^{(m)}}\right\} \qquad \begin{cases} i^{'},j^{'}=0,\cdots,13 \\ m=0,\cdots,15 \end{cases}$$

要根据此图得到从 S2 到 C3 的一对多关系图。

方法原理为，**固定左端 $s2_{i^{'},j^{'}}^{(k)}$ 的下标 $(i^{'},j^{'},k)$**，并且在约束关系 $\begin{cases} i^{'}=i+2+u \\ j^{'}=j+2+v \end{cases}$ 条件下，小窗口坐标 (u,v) 为卷积窗口与 $s2_{i^{'},j^{'}}^{(k)}$ 对应的权值的下标，相当于卷积窗口移动时，即 (u,v) 变化时，满足约束关系 $\begin{cases} i^{'}=i+2+u \\ j^{'}=j+2+v \end{cases}$ 的 $s2_{i^{'},j^{'}}^{(k)}$ 的对应点 $u_{c3_{i^{'}-2-u,j^{'}-2-v}^{(m)}}$ 也是变化的，对应关系就体现在下标关系式中，并且当固定 k 时，$m=0,\cdots,15$ 变为游标，从而得到从 S2 到 C3 的一对多关系图如下：

$$\left\{s2_{i^{'},j^{'}}^{(k)}\right\} \xrightarrow{1\ to\ 400} \left\{w3_{u,v}^{(k,m)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\m=0,\cdots,15}} \xrightarrow{400\ to\ 400} \left\{u_{c3_{i^{'}-2-u,j^{'}-2-v}^{(m)}}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\m=0,\cdots,15}} \qquad \begin{cases} i^{'},j^{'}=0,\cdots,13 \\ k=0,\cdots,5 \end{cases}$$

将此图左右翻转，则推出反向传播关系图为，

$$\left\{u_{c3_{i^{'}-2-u,j^{'}-2-v}^{(m)}}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\m=0,\cdots,15}} \xrightarrow{400\ to\ 400} \left\{w3_{u,v}^{(k,m)}\right\}_{\substack{u=[-2,2]\\v=[-2,2]\\m=0,\cdots,15}} \xrightarrow{400\ to\ 1} \left\{s2_{i^{'},j^{'}}^{(k)}\right\} \qquad \begin{cases} i^{'},j^{'}=0,\cdots,13 \\ k=0,\cdots,5 \end{cases}$$

做卷积核下标变换 $u^{'}=-u$，$v^{'}=-v$，反向传播关系图变为，

$$\left\{u_{c3_{i'-2+u',j'-2+v'}^{(m)}}\right\}_{\substack{u'=[-2,2]\\ v'=[-2,2]\\ m=0,\cdots,15}} \xrightarrow{400\ to\ 400} \left\{w3_{-u',-v'}^{(k,m)}\right\}_{\substack{u'=[-2,2]\\ v'=[-2,2]\\ m=0,\cdots,15}} \xrightarrow{400\ to\ 1} \left\{s2_{i',j'}^{(k)}\right\} \qquad \begin{cases} i',j'=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

再换回下标 $i'=i$，$j'=j$，$u'=u$，$v'=v$，**反向传播关系图**变为，

$$\left\{u_{c3_{i-2+u,j-2+v}^{(m)}}\right\}_{\substack{u=[-2,2]\\ v=[-2,2]\\ m=0,\cdots,15}} \xrightarrow{400\ to\ 400} \left\{w3_{-u,-v}^{(k,m)}\right\}_{\substack{u=[-2,2]\\ v=[-2,2]\\ m=0,\cdots,15}} \xrightarrow{400\ to\ 1} \left\{s2_{i,j}^{(k)}\right\} \qquad \begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

由此反向传播关系图，得到敏感值反向关系式为

$$\delta\left(s2_{ij}^{(k)}\right)=\frac{\partial J(\theta)}{\partial\left(s2_{ij}^{(k)}\right)}=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}\frac{\partial J(\theta)}{\partial\left(u_{c3_{i-2+u,j-2+v}^{(m)}}\right)}\cdot\frac{\partial\left(u_{c3_{i-2+u,j-2+v}^{(m)}}\right)}{\partial\left(s2_{ij}^{(k)}\right)} \qquad \begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

$$\delta\left(s2_{ij}^{(k)}\right)=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}\delta\left(u_{c3_{i-2+u,j-2+v}^{(m)}}\right)\cdot w3_{-u,-v}^{(k,m)} \qquad \begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

证明完成。

<u>Gradient in Layer S2</u>

与 S4 层类似，目前普遍认为，S2 层的 sub-sampling 操作就是一个池化操作（pooling），本节只给出 max-pooling 算法的反传公式。

S2 层将 C1 层的 6 个特征图分别缩小 1 倍，不涉及可学习参数，只有输入数据 $c1_{ij}^{(k)}$，$\begin{cases} i,j=0,\cdots,27\\ k=0,\cdots,5 \end{cases}$ 和输出数据 $s2_{ij}^{(k)}$，$\begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$。其中，$\delta\left(s2_{ij}^{(k)}\right)$ 已经在前一个小结（**<u>Gradient in Layer C3</u>**）计算出来，

$$\delta\left(s2_{ij}^{(k)}\right)=\sum_{m=0}^{15}\sum_{u=-2}^{2}\sum_{v=-2}^{2}\delta\left(u_{c3_{i-2+u,j-2+v}^{(m)}}\right)\cdot w3_{-u,-v}^{(k,m)} \qquad \begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

本小结（**<u>Gradient in Layer S2</u>**）只需要计算出 $\delta\left(c1_{ij}^{(k)}\right)=\frac{\partial J(\theta)}{\partial\left(c1_{ij}^{(k)}\right)}$ $\begin{cases} i,j=0,\cdots,27\\ k=0,\cdots,5 \end{cases}$。

池化操作（pooling）操作大多采用 max-pooling 算法，

$$s2_{ij}^{(k)}=\max\left\{c1_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\ v=0,1}} \qquad \begin{cases} i,j=0,\cdots,13\\ k=0,\cdots,5 \end{cases}$$

则只对每四个 $c1_{2i+u,2j+v}^{(k)}\begin{cases}u=0,1\\v=0,1\end{cases}\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}$ 中最大的一个 $\max\left\{c1_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}}\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}$ 有唯一的一个 $s2_{ij}^{(k)}$ 与其对应， 为此，制作一个与 $c1_{2i+u,2j+v}^{(k)}$ 一一对应的掩膜矩阵(mask matrix for max-pooling) $msk1_{2i+u,2j+v}^{(k)}$

$$msk1_{2i+u,2j+v}^{(k)}=\begin{cases}1 & c1_{2i+u,2j+v}^{(k)}=\max\left\{c1_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}}\\ 0 & c1_{2i+u,2j+v}^{(k)}\neq\max\left\{c1_{2i+u,2j+v}^{(k)}\right\}_{\substack{u=0,1\\v=0,1}}\end{cases}\qquad\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}$$

另外，本小结需要计算出 $\delta\left(c1_{ij}^{(k)}\right)\begin{cases}i,j=0,\cdots,27\\k=0,\cdots,5\end{cases}$，也就是说，要修改 $s2_{ij}^{(k)},\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}$ 的下标以适应左边，方法如下，

对每一个 $c1_{2i+u,2j+v}^{(k)}$ 有唯一的一个 $s2_{ij}^{(k)}$ 与其对应（严谨来说，是可能唯一的对应，因为 max-pooling 是从 4 个 $c1_{2i+u,2j+v}^{(k)}\begin{cases}u=0,1\\v=0,1\end{cases}$ 中挑一个最大的与 $s2_{ij}^{(k)}$ 对应），

$$c1_{2i+u,2j+v}^{(k)}\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\\u,v=0,1\end{cases}\quad\rightarrow\quad s2_{ij}^{(k)}\begin{cases}i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}$$

相当于：

$$c1_{i,j}^{(k)}\begin{cases}i,j=0,\cdots,27\\k=0,\cdots,5\end{cases}\quad\rightarrow\quad s2_{\text{floor}\left(\frac{i}{2}\right),\text{floor}\left(\frac{j}{2}\right)}^{(k)}\begin{cases}i,j=0,\cdots,27\\k=0,\cdots,5\end{cases}$$

注意其中下标范围的变化，实质对应关系没有改变。掩膜矩阵也做相应修改为

$$msk1_{2i+u,2j+v}^{(k)}\begin{cases}u=0,1\\v=0,1\\i,j=0,\cdots,13\\k=0,\cdots,5\end{cases}\quad\rightarrow\quad msk1_{i,j}^{(k)}\begin{cases}i,j=0,\cdots,27\\k=0,\cdots,5\end{cases}$$

在做好上述掩膜矩阵对应关系与下标的变换后，就可以直接给出 $\delta\left(c1_{ij}^{(k)}\right)$ 的计算公式了，

$$\delta\left(c1_{ij}^{(k)}\right)=msk1_{i,j}^{(k)}\cdot\delta\left(s2_{\text{floor}\left(\frac{i}{2}\right),\text{floor}\left(\frac{j}{2}\right)}^{(k)}\right)\qquad\begin{cases}i,j=0,\cdots,27\\k=0,\cdots,5\end{cases}$$

Gradient in Layer C1

C1 层有 6 个 feature maps，每个 $\left\{c1_{ij}^{(k)} \in \left[c1_{ij}^{(k)}\right]_{28\times28}, k=0,\cdots,5\right\}$ 就是一个 neuron，共有 28×28×6=4704 个 neurons。如下所列：

$$\begin{matrix}\left[c1_{ij}^{(0)}\right]_{28\times28} & \left[c1_{ij}^{(1)}\right]_{28\times28} & \left[c1_{ij}^{(2)}\right]_{28\times28}\\ \left[c1_{ij}^{(3)}\right]_{28\times28} & \left[c1_{ij}^{(4)}\right]_{28\times28} & \left[c1_{ij}^{(5)}\right]_{28\times28}\end{matrix}$$

C1 层 neuron 按如下式计算，

$$c1_{ij}^{(k)} = f\left(\sum_{u=-2}^{2}\sum_{v=-2}^{2} w1_{u,v}^{(k)} \cdot x_{i+2+u,j+2+v} + b1^{(k)}\right) \qquad \begin{cases} i,j = 0,\cdots,27 \\ k = 0,\cdots,5 \end{cases}$$

其激活函数的输入部分为 $u_{c1_{ij}^{(k)}}$

$$u_{c1_{ij}^{(k)}} = \sum_{u=-2}^{2}\sum_{v=-2}^{2} w1_{u,v}^{(k)} \cdot x_{i+2+u,j+2+v} + b1^{(k)} \qquad \begin{cases} i,j = 0,\cdots,27 \\ k = 0,\cdots,5 \end{cases}$$

（1）每个 C1 层神经元的激活函数括号里面的输入值 $u_{c1_{ij}^{(k)}}$ 的敏感值 $\delta\left(u_{c1_{ij}^{(k)}}\right)$，求导链是单一链，

$$\delta\left(u_{c1_{ij}^{(k)}}\right) = \frac{\partial J(\theta)}{\partial\left(u_{c1_{ij}^{(k)}}\right)} = \frac{\partial J(\theta)}{\partial\left(c1_{ij}^{(k)}\right)} \cdot \frac{\partial\left(c1_{ij}^{(k)}\right)}{\partial\left(u_{c1_{ij}^{(k)}}\right)} = \delta\left(c1_{ij}^{(k)}\right) \cdot f'\left(u_{c1_{ij}^{(k)}}\right) \qquad \begin{cases} i,j = 0,\cdots,27 \\ k = 0,\cdots,5 \end{cases}$$

（2）下面求该层权值参数 $w1_{u,v}^{(k)}$，$\begin{cases} k = 0,\cdots,5 \\ u,v = -2,-1,0,1,2 \end{cases}$ 的梯度，注意，这里卷积核是共用的，即每个 $w1_{u,v}^{(k)}$ 会使用 784 次（$i,j = 0,\cdots,27$），或者说，每个 $w1_{u,v}^{(k)}$ 会对 784 个 $\left\{u_{c1_{ij}^{(k)}}, i,j = 0,\cdots,27\right\}$ 都有贡献，则有链式（并行多链）求导公式如下：

$$\delta\left(w1_{u,v}^{(k)}\right) = \frac{\partial J(\theta)}{\partial\left(w1_{u,v}^{(k)}\right)} = \sum_{i=0}^{27}\sum_{j=0}^{27} \frac{\partial J(\theta)}{\partial\left(u_{c1_{ij}^{(k)}}\right)} \cdot \frac{\partial\left(u_{c1_{ij}^{(k)}}\right)}{\partial\left(w1_{u,v}^{(k)}\right)} \qquad \begin{cases} k = 0,\cdots,5 \\ u,v = -2,-1,0,1,2 \end{cases}$$

$$\delta\left(w1_{u,v}^{(k)}\right) = \sum_{i=0}^{27}\sum_{j=0}^{27} \delta\left(u_{c1_{ij}^{(k)}}\right) \cdot x_{i+2+u,j+2+v} \qquad \begin{cases} k = 0,\cdots,5 \\ u,v = -2,-1,0,1,2 \end{cases}$$

（3）下面求该层偏差参数 $\{b1^{(k)}, k=0,\cdots,5\}$ 的梯度，同理，截距 $b1^{(k)}$ 也是共用的，即每个 $b1^{(k)}$ 也会使用 784 次（$i,j=0,\cdots,27$），或者说，每个 $b1^{(k)}$ 会对 784 个 $\left\{u_{c1_{ij}^{(k)}}, i,j=0,\cdots,27\right\}$ 都有贡献，则有链式（并行多链）求导公式如下：

$$\delta\left(b1^{(k)}\right)=\frac{\partial J(\theta)}{\partial\left(b1^{(k)}\right)}=\sum_{i=0}^{27}\sum_{j=0}^{27}\frac{\partial J(\theta)}{\partial\left(u_{c1_{ij}^{(k)}}\right)}\cdot\frac{\partial\left(u_{c1_{ij}^{(k)}}\right)}{\partial\left(b1^{(k)}\right)}=\sum_{i=0}^{27}\sum_{j=0}^{27}\delta\left(u_{c1_{ij}^{(k)}}\right)\cdot 1 \qquad k=0,\cdots,5$$