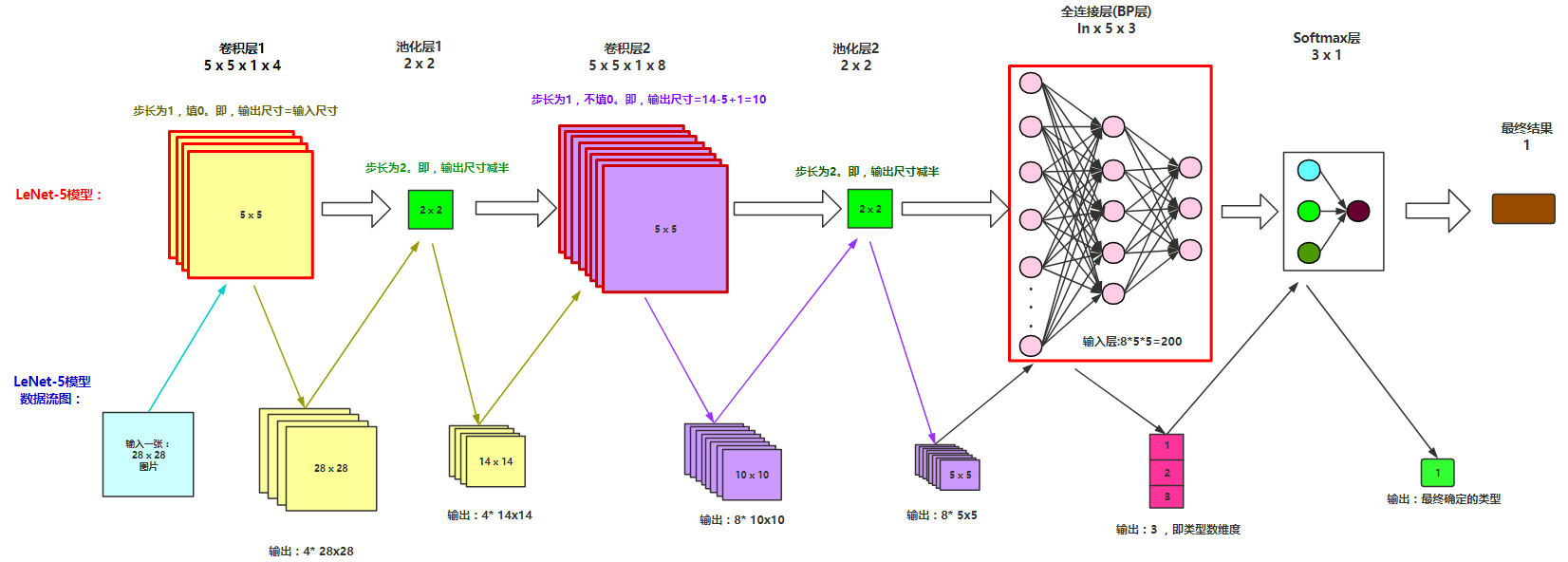
CNN算法

# CNN网络结构



# 卷积算法

卷积总结：

1. 目的：取样，提取输入特征；
2. 运算规则：y=x(n)\*h(n)，\*代表卷积运算，x,h对应项相乘，结果相加。x,h维度一定相同，一次卷积运算得出的只是一个标量，要得到其它值，需要移动卷积窗口。
3. 步长移动：根据输入输出是否变化，有两种类型（一般W=L）：
   * + - 维度不变：步长为1，填0，则输出维度=输入，可变卷积核尺寸，多用这种
       - 降维输出：步长=1，不填0；或者步长>1；输出维度<输入
4. 填0操作：因为“卷积核维度=取样窗口”时，才能成功运算，当取到右或下边缘，余下取样数据不足时，可以采用填0补足；如果不填0，则意味着抛弃余下的边缘。
5. 多通道运算规则：对于多通道输入（N通道），卷积核与每一个通道作卷积运算，得到的N个结果，再将这N个结果作累加+偏置，最后得到唯一输出；所以最终输出结果的深度=卷积层深度；

卷积运算：;（z:卷积结果项，w:卷积权重，x:输入，: 卷积核维数）

多通道输入结果累加：;（:卷积输出项，:该卷积偏置，:输入通道数(也可以定制，不用全组合)）

最终卷积输出：; （:卷积深度，即卷积核数量）

# 二、交叉熵 & Softmax算法

假如有两组正确分类结果：

**结果一：[0.98, 0.97], max==> [1,0]**

**结果二：[0.9, 0.1], max==> [1,0]**

虽然它们的分类结果都正确，但是我们明显发现做出第一组结果的网络是不够优秀的，是需要继续学习的，而第二组已经很优秀了。但是单纯使用max函数，我们从[1,0]这个结果上看，它们的误差竟然是相同的，那么这个评分标准就是有问题的。

上面的问题在所有分类NN中都有，而人家选择了softmax和“交叉熵”来解决这个问题。

## 2.1 softmax

还是看上面的例子，我们希望最终结果能体现这样一种需求：它能通过**数值**表达出**[0.98, 0.97]**这个分类结果的表现不好，要加大学习量；而**[0.9, 0.1]**这个分类结果，很优秀，相关网络只需微调即可。有人想出了softmax算法。

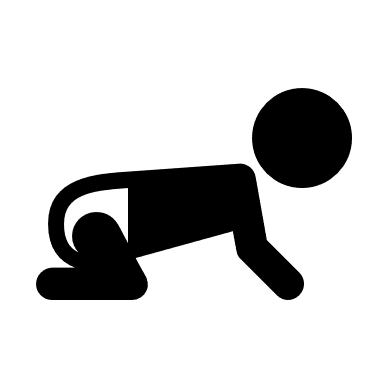
**算法定义**：

上式中：为softmax的输入数组的第i个元素， 就是它对应的softmax输出。上式所表达的就是：所有输入的自然指数和与它的输入的自然指数之比，就是这个输入的输出值。我们可以看出来，这其实是一种归一化方法。

光凭感觉，初步算一算，我们能看出它基本能达到我们的目的，但是有那么多算法，为什么非要用softmax呢？再说它看起来还是有点复杂啊！

关于为什么要用它，以及它的优点。要评价这些，就不得不说到交叉熵。

## 2.2 交叉熵

交叉熵在CNN（以及普通DNN）中用作误差求解算法。说到交叉熵，就不得不说到信息量，香农熵。让我们回到一切的起点，从头说起。

### 2.2.1 信息熵

首先，说一个能计算世间万物复杂度的公式，香农公式（信息熵公式）：

S代表整个集合，是第i类在S中再现的概率。

当然，对于类别i的熵：

之所以一开始讲集合的熵，而非个体的熵，是因为：统计必然是在多样本下做的才能统计出概率麻。

例：计算以下三个句子的信息熵。

代入信息熵公式，得：

**这里解释一下，复杂度函数为什么是对数函数？**

复杂度函数需要满足两个条件：

1. 复杂度函数f是概率P(x)的单调递减函数，一个类型出现概率越大，当然复杂度越低，全都是它了，那就没有什么复杂度了，反之亦然。
2. 两个独立变量的复杂度，应为各自复杂度之各，即f()=f()+f()

要同时满足以上两个要求的，就只有对数函数：。

一般来说，log的底数取2，也可以取10，无所谓(必须>1)，反正有万能的**换底公式**。

个人总结：

**熵 = 复杂度**

**= 维持复杂度所需的代价（储存空间，暴力支持）**

**= 消除复杂度所需的代价（猜球次数，暴力破坏）**

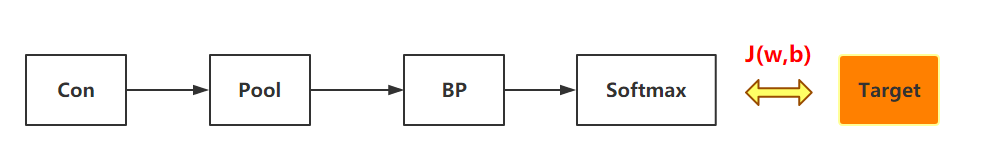
### 2.2.2 交叉熵&相对熵

令p为我们的期望分步概率，而q为我们的实际分步概率结果。那么用期望来表示实际的熵，公式为：

上式就叫交叉熵公式。它与期望熵相比，就能得到相对熵：

# 三、CNN的BP算法推导

我们以最简单的4层CNN网络为例，推导CNN的BP算法，网络结构如下图：



最终的输出误差为：