Convolutional Neural Network

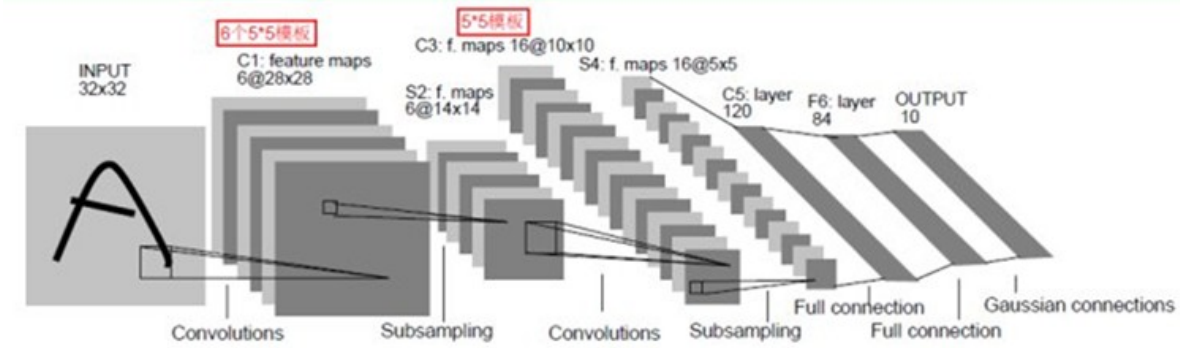
最近刚从PRML转向DL，感觉有了基本的ML知识，对于DL的理解也会更加深刻。

下文是参考了各个网站上的CNN介绍，以及MATLAB上关于DeepLearningToolBox的代码实现，若有不正确的地方请大家原谅并指出，本文是对这几天的实验总结，目的只是单纯的想让自己在忘记的时候，能够看到这篇文章快速想起来。**若有转载，请注明出处，谢谢！**

网上关于CNN的介绍文章数不胜数，毕竟在我看来算得上是DL标志性算法之一，具体的介绍请看http://blog.csdn.net/zouxy09

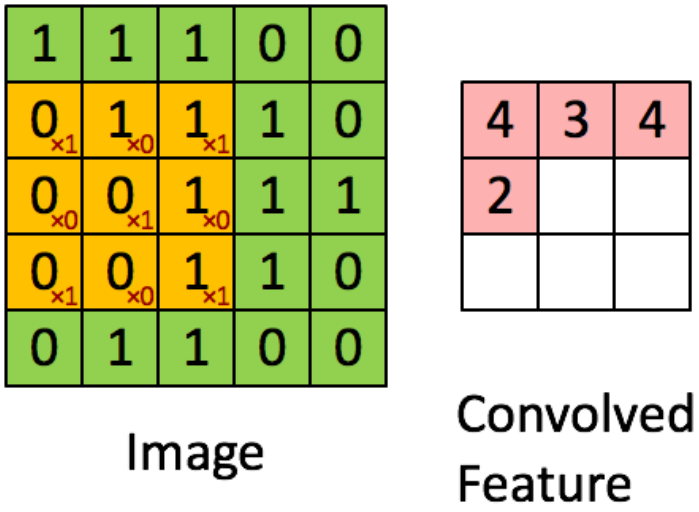
Structure of CNN

CNN的结构见仁见智，变化的多样性数不胜数，常见的是一层卷积层跟一层降采样层（pooling），最后补上一个全连接的感知器，通过1-of-K scheme导出各个Label的可能性，下图是经典的LeNet-5



Feed forward in CNN

不算最后的全连接感知器，CNN的前向传播的过程很简单，如果当前层是卷积层，那么该层的输出就是输入卷积矩阵卷积后，加上偏置（每一个层的每一个节点共享一个偏置和一个卷积矩阵，这就是CNN的最重要的特点之一，权值共享）。两个矩阵卷积的过程见下图



这样计算出来的输出通过激活函数（通常是sigmoid或者tanh）即可得到相应的卷积层的输出

如果该层是池化层，那么该层的输出矩阵的大小是输入矩阵的大小除于池化系数，输出的值是输入矩阵的局部平均。

$$\begin{pmatrix} 8 & 1 & 2 & 3 \\ 1 & 2 & 1 & 4 \\ 2 & 2 & 9 & 0 \\ 2 & 1 & 1 & 1 \end{pmatrix} \xrightarrow{pooling} \begin{pmatrix} 3 & 2.5 \\ 1.75 & 2.75 \end{pmatrix}$$

Back propagation in CNN

全连接感知器的最后一层的BP过程与ANN的BP过程没什么区别，具体的BP过程可以根据Softmax或者Logistic两种不同的多样本分类器进行不同的BP。我们假设误差已经传入CNN的最后一层（感知器的前一层）

Pool layer BP procedure

如果该层是池化层，我们首先需要对前一层的误差进行Desigmoid处理（我们假设前一层是卷积层，如果不是则不需要进行这一步），这样做的原因是卷积层的输出做过激活函数处理，对于误差首先需要对激活函数求导。此时，误差的传播过程实际上就是池化的逆过程，也就是将该层的误差复制池化系数的倍数， 并将误差平均化，具体做法为

$$Error: \begin{pmatrix} 4 & 8 \\ 6 & 9 \end{pmatrix} \xrightarrow{depooling} \begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1.5 & 1.5 & 2.25 & 2.25 \\ 1.5 & 1.5 & 2.25 & 2.25 \end{pmatrix}$$

然后没有参数需要修改

Convolutional layer BP procedure

如果该层为卷积层，误差的传播过程是同样也是一种卷积的过程，具体的做法是

$$Weight\ Matrix: \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$Error\ of\ current\ layer: \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

> *一般的卷积是有效矩阵的卷积，而下一种卷积可以看成是一种全卷积*

将本层的误差矩阵补成大小为(2 * conv.rows + err.rows - 2, 2 * conv.cols + err.cols - 2)，补上的区域都可以添上为0，本例中可以视为

$$Changed\ Error: \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A & B & 0 \\ 0 & C & D & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

接下来，将卷积矩阵（本身不需要）**旋转180度**

$$Rot\ Weight\ Matrix: \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$$

用旋转过的权值矩阵去卷积添0后的误差矩阵，即可以获得前一层的误差矩阵

$$Error\ of\ previouslayer: \begin{pmatrix} A & 2A+B & 2B \\ 3A+C & 4A+3B+2C+D & 4B+2D \\ 3C & 4C+3D & 4D \end{pmatrix}$$

相应的**权值矩阵的修改**（可用线性代数证明）：

前一层的输出为$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$，该层的误差矩阵为$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$

该层的权值矩阵的修改为该层的误差去卷积前一层的输出值

$$dWeightMatrix: \begin{pmatrix} Aa+Bb+Cc+Dd & Ab+Bc+Ce+Df \\ Ad+Be+Cg+Dh & Ae+Bf+Ch+Di \end{pmatrix}$$

Notation

其实CNN另一个重要特点是局部感知域，以上的前向传播和后向误差传递的过程都需要考虑局部连接的条件

About Me

如果有任何问题都可以发邮件给651636074@qq.com，欢迎指出问题并一起讨论

Author: Yang

Date: 2015-9-18