# CNN文本分类程序说明书

## 数据集介绍

该程序使用的是新闻文本数据，训练数据50000个，测试数据10000个，验证集5000。整个数据集分为娱乐、财经、科技等10个类别

## CNN在文本分类中的应用

如图2.1，输入有两个通道（可看成一个），两个通道分别是 static 和 non-static； non-static 就是词向量随模型训练变化，好处是词向量可以根据数据集做适当调整，数据集较少时不推荐此操作，否则容易产生过拟合现象；static是直接使用 word2vec 训练好的词向量。输入层是将一个句子所有单词的词向量拼接成一个矩阵，每行代表一个单词；每个句子固定20(或其他个数)个词，不够的padding。

**因此行的长度是词向量的长度，列的长度是句子中单词的个数**

卷积层，每个卷积核大小为 filter\_size\*embedding\*size。filter\_size 代表卷积核纵向上包含单词数；embedding\_siz代表词向量的维度；每个卷积核计算完成 后就得到 1 个列向量，代表卷积核从句子中提取出的特征；上图中使用了 4 个卷积核。

**注意：卷积核只能在纵向上移动**

池化层pooling 操作就是将卷积得到的列向量最大值提取出来，pooling 操作后会得到一个 num\_filter(卷积核数目)维行向量；若之前未对句子进行 padding 操作，句子长度不同，卷积后列向量维度也不同，通过 pooling 可消除句子间长度不同差异

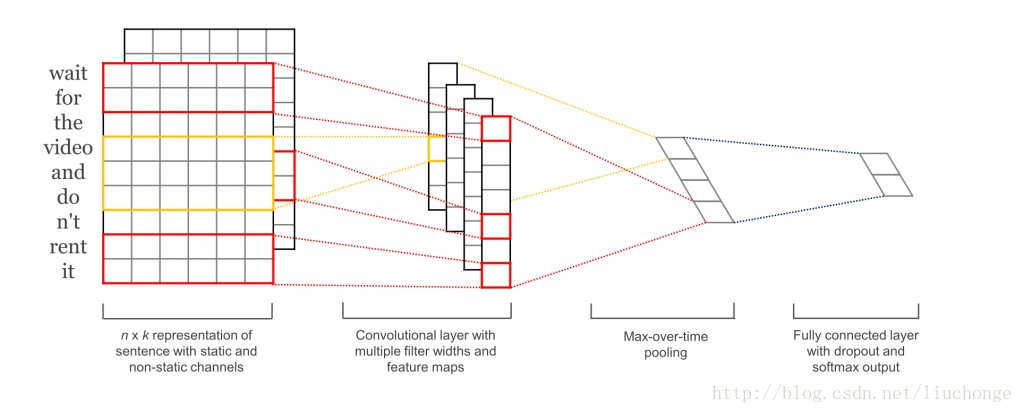


图2.1

## 程序实现过程

读取数据，将标签和文本分开，并将labels用one\_hot编码，content里的文字用词向量表示，在本程序中各个字的词向量都是随机初始化，并未使用word2vdc训练词向量。读取一个batchsize数据放入CNN网络中，注意每个batchsize中样本都要pad成相同的长度

CNN网络框架：1个卷积层+2个全链结层，loss函数为softmax\_cross\_entropy\_with\_logits, 优化函数是AdamOptimizer

本项目包括三个子程序分别是model.py(CNN模型)，main.py(主函数)，predict.py(模型训练好后实际部署用于预测)

## 4程序效果

