

**《自然语言处理》**

**课**

**程**

**大**

**作**

**业**

人工智能与计算机学院

数字媒体技术专业

班级：数字媒体技术2004

姓名：万嘉涌

学号：1191200426

授课教师：吴杰

二〇二三年六月

# 用于文本分类的卷积神经网络的超参数研究

1191200426万嘉涌

**0 摘要（Abstract）**

文本分类是自然语言处理任务中实际被应用最多的一类任务，具体方法包括传统方法（TF-IDF+SVM）, TextCNN, 预训练词向量+TextCNN, TextRNN。其中卷积神经网络(CNN)在文本分类的实际重要任务上取得了显著的强大性能。但不同的卷积神经网络结构也会使得性能之间有所差异。卷积神经网络结构差异包括卷积核大小、卷积层数等等。因此，我在前科学家[1][2]研究的基础上，针对MR(Movie Review)数据集，专注于单层卷积神经网络,使用华为开发的MindSpore对不同的卷积神经网络超参数进行了测试，从若干实验中获得性能更好的卷积神经网络超参数。

**1 背景与个人实验简介**

**（Background and Personal experiment profile）**

卷积神经网络前几年被证明在文本分类这一实际重要任务上取得了显著的成效，以及在文本分类任务上使用大数据集上的预训练词向量能够使准确率取得很大的提高[1]。Kim(2014)提出了CNN-static-Word2Vec，CNN-non-static-Word2Vec和CNN-multichannel三个模型。这三者的性能都明显优于无预训练词向量的基准模型CNN-rand。本文基于MindSpore实现了第一个模型（CNN-static-Word2Vec），并依据[2]的实验流程对不同的CNN结构在MR数据集上进行了测试。在基准实验（CNN-rand）的情况下添加了预训练词向量Word2Vec，选择了不同的卷积核大小(kernel\_size)、卷积核组合（kernel\_group）、特征提取个数(feature\_map)，并调整了迭代次数(epoch\_size)。

**2 用于文本分类的卷积神经网络结构概述**

**（Overview of The Structure of CNN）**

2.1 词嵌入层（Word-embedding Layer）

对于一个句子来说，词嵌入层将句子中的每个单词转化为一个维度(dimension)相同的向量，也叫做词向量。词嵌入层起到将文本输入量化成向量输出的作用。因此一个包含有n个单词的句子，经过词嵌入层后输出一个n行矩阵，矩阵的列数即为每个向量的维度d。

2.2 卷积层（Convolutional Layer）

本文研究仅限于单卷积层结构。

在卷积层中包含有若干卷积核（convolutional kernel / filter）。每个卷积核都有自己的高度和宽度。在文本分类任务中，卷积核的宽度等于词向量的向量维度d，是个定值。而高度则由使用者自己设定。在卷积层运算中，卷积核在词向量矩阵上按照一定的步长(stride)由上而下滑动，与被覆盖住的矩阵进行互相关运算（数学称呼）。具体运算细则可见[1]。

同种卷积核、不同卷积核的个数也由使用者自己设定。Figure1中使用了三种不同规模的卷积核（卷积核高度分别为2，3，4），每个卷积核两个。共计6个卷积核，得到输出6个，但此时输出还不是特征矩阵(feature\_map)，仍须经过激活函数（activation function）映射后才得到6个特征矩阵。激活函数会在接下来具体解释。

2.3 激活函数（Activation function）

上述卷积层的输出即类似于全连接神经网络中的z=w\*x+b，只不过在卷积神经网络中，参数w就是卷积核，x是卷积核覆盖的矩阵词向量矩阵。相对于全连接神经网络中每个输出都与当前层的所有输入相关联，卷积层中的每个输出只与当前层所有输入的局部区域存在连接权重（即产生交互），因此卷积层是稀疏连接的，或者称为稀疏权重。

显然，经过卷积层的输出z还需经过激活函数映射后才可得到特征输出。一般卷积神经网络中使用的激活函数包括双曲正切函数（tanh）和线性整流函数（ReLu）。本文中不对不同激活函数的性能进行测试，采用ReLu作为激活函数。

卷积层输出经过激活函数映射后即得到特征矩阵。

Figure1中可见特征矩阵大小与卷积核相互对应，不同大小的卷积核得到的输出矩阵也是不同的。

2.4 池化层（Pooling Layer）

池化层对输入的操作其实可以看作和卷积是一样的，二者均使用一个filter在输入矩阵上滑动，只不过池化层的卷积核计算操作与卷积层不同。本文采用最大池化（1-MaxPooling）策略，即对池化层的卷积核覆盖的区域内，取出该区域的最大元素，然后把其他元素抛弃。

如Figure1所示，对每个feature\_map进行最大池化操作，池化卷积即为输入矩阵的行数，最后每个输入矩阵得到一个元素输出。

2.5 全连接层（Dense Layer）

容易知道经过池化层之后得到若干个最大池化输出，对它们追根溯源的话，可以发现它们分别来源不同的卷积核，不同的特征，可以说它们的产生过程是无关联的，这些池化层输出是松散的。但是它们是我们分别从之前若干个特征矩阵中寻找到的最大特征，这些特征的来源都是相同的——进入神经网络的文本。因此接下来我们将池化层的输出拼接在一起（容易知道它们是维度相同的向量），作为全连接层的输入。

直到全连接层之前，文本的特征提取算是结束了。在全连接层中，我们就可以参照传统全连接神经网络的经验，对提取到的特征向量进行处理，从而实现二分类了。

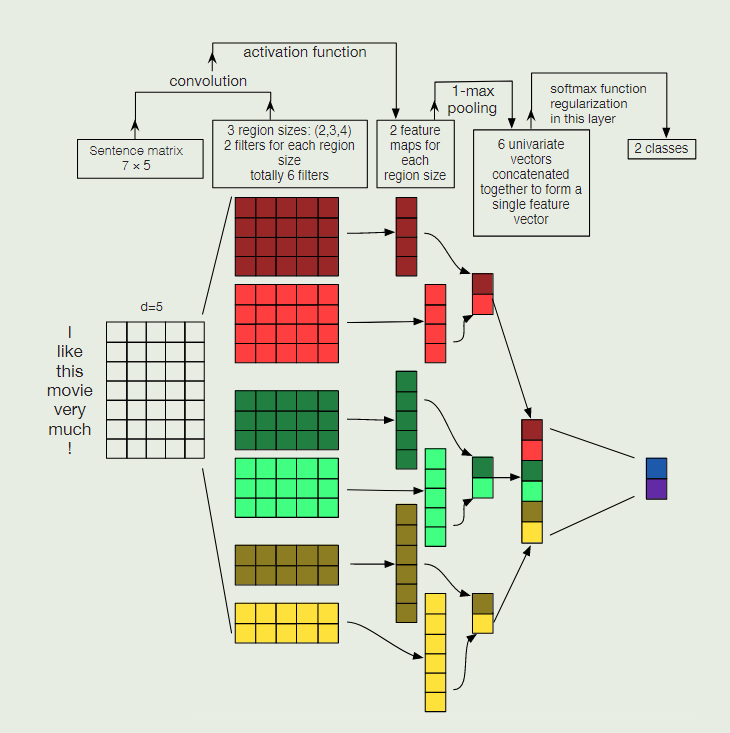


Figure1.卷积神经网络结构

**3 数据集和实验设置**

**（Dataset and Experimental Setup）**

3.1 数据集（Dataset）

我使用MR数据集测试不同卷积神经网络结构的性能。

MR(Movie Review)：影评数据集，每条电影评论只有一句话，分类结果包含正面评论和负面评论两种。

3.2 基准设置（Baseline Configuration）

预训练词向量（pre-trained\_word\_vectors） ： 无

卷积核区域大小（filter\_size） ： (3,4,5)

每个卷积核对应特征矩阵个数(feature\_map) ： 96

激活函数（activation Function） ： ReLU

池化策略（pooling） ： 最大池化

Dropout Rate ： 0.5

迭代次数（epoch\_size） ： 4

批大小（batch\_size） ： 64

测试集准确率（accuracy） ： 0.7548

3.3 预训练词向量（Pre-trained Word Vectors）

使用从无监督神经语言模型中获得的词向量初始化词向量是在缺乏大型监督训练集的情况下提高性能的一种流行方法。本实验使用公开的word2vec向量，这些向量是在谷歌新闻的1000亿个单词上训练的，向量的维数为300。对于未出现在预训练词集中的词，随机初始化其对应的词向量。

（使用的预训练词向量：<https://code.google.com/archive/p/word2vec/>）

3.4 模型种类（Model Variations）

CNN-rand ：基准模型的模型结构，使用3.2的基准设置，其中所有词向量均为随机初始化，且在训练过程中不更改。

CNN-static ：使用预训练词向量word2vec的模型，所有单词都保持静态——即在训练过程中不会更改，只学习模型的其他参数。

**4 实验数据分析**

**（Analysis of Experimental Data ）**

4.1 迭代次数对不同大小卷积核区域下模型准确率的影响

在基准模型中对原数据集迭代了四次，通过对迭代数据的观察，发现迭代进行到一定迭代次数时，损失函数已经非常小且在一个数值附近不断震荡，说明这之后的迭代已经属于无效迭代，因此对迭代次数进行一些调整。接下来的实验均从epoch\_size=1开始进行测试，观察不同卷积核区域在不同迭代次数下的变化情况。

相关实验数据如下：

定量设置：

预训练词向量（pre-trained\_word\_vectors） ： 有

每个卷积核对应特征矩阵个数(feature\_map) ： 96

激活函数（activation Function） ： ReLU

池化策略（pooling） ： 最大池化

Dropout Rate ： 0.5

迭代次数（epoch\_size） ： 4

批大小（batch\_size） ： 64

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accuracy | Filter\_size | (3,4,5) | (5,6,7) | (6,7,8) | (7,8,9) | (7,7,7,7) |
| Epoch\_size |  |  |  |  |  |  |
| 1 |  | 0.8085 | 0.7822 | 0.8066 | 0.8066 | **0.8164** |
| 2 |  | 0.7968 | 0.8085 | 0.8105 | **0.8154** | 0.8037 |
| 3 |  | **0.8134** | **0.8212** | **0.8115** | 0.8056 | 0.7988 |
| 4 |  | 0.8066 | 0.7968 | 0.8056 | 0.7968 | 0.8056 |

Figure2.在相同filter\_size的情况下不同epoch\_size对模型准确率的影响

对于特征量或卷积核区域较小的模型来说，适当增加迭代次数，降低损失函数，有利于提高模型在测试集上的准确性。

对于特征量较大的模型来说，通过高迭代次数获得很小的损失函数，容易导致模型过拟合。此时减少迭代次数反而有利于提高模型在测试集上的准确性。

总而言之，对于不同卷积核区域大小组合的模型来说，每个模型都有一个最适合于自身的迭代次数，在这个迭代次数下能使模型准确率相对于其他迭代次数较高。

4.2 预训练词向量使用与否对模型准确率的影响

文本分类模型以一个句子中单词的向量表示作为输入，这种体系结构使得在模型初始化期间可以构造或下载我们想要的预训练词向量，并使之作为输入。而基于无监督神经网络训练获得的预训练词向量word2vec在对单词的特征提取上效果显著高于独热编码（one-hot）和随机初始化词向量。

本文对比了基准设置模型和使用word2vec作为词嵌入层输入的模型，结果显示预训练词向量的使用使模型准确率获得了显著提升。

相关实验数据如下：

定量设置：

卷积核区域大小（filter\_size） ： (3,4,5)

每个卷积核对应特征矩阵个数(feature\_map) ： 96

激活函数（activation Function） ： ReLU

池化策略（pooling） ： 最大池化

Dropout Rate ： 0.5

迭代次数（epoch\_size） ： 4

批大小（batch\_size） ： 64

|  |  |  |
| --- | --- | --- |
| Pre-trained word vectors | no | yes |
| Accuracy | 0.7548 | **0.8066** |

Figure3.在其他参数相同的情况下使用预训练词向量与否对模型准确率的影响

4.3 卷积核区域大小对不同迭代次数下模型准确率的影响

本文探索了组合不同卷积核区域大小的效果。同时保证每个区域大小的特征图矩阵数量固定为96。基于[2]的研究，在单个最优卷积核区域大小附近进行组合。

相关实验数据如下：

定量设置：

预训练词向量 ： 有

每个卷积核对应特征矩阵个数(feature\_map) ： 96

激活函数（activation Function） ： ReLU

池化策略（pooling） ： 最大池化

Dropout Rate ： 0.5

迭代次数（epoch\_size） ： 1

批大小（batch\_size） ： 64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Accuracy | Epoch\_size | 1 | 2 | 3 | 4 |
| Multiple region\_size |  |  |  |  |  |
| (3,4,5) |  | 0.8085 | 0.7968 | 0.8134 | **0.8066** |
| (5,6,7) |  | 0.7822 | 0.8085 | **0.8212** | 0.7968 |
| (6,7,8) |  | 0.8066 | 0.8105 | 0.8115 | 0.8056 |
| (7,8,9) |  | 0.8066 | **0.8154** | 0.8056 | 0.7968 |
| (7,7,7,7) |  | **0.8164** | 0.8037 | 0.7988 | 0.8056 |

Figure4.相同迭代次数的情况下不同的卷积核区域组合对模型准确率的影响

容易发现，当迭代次数越高，卷积核区域越小的组合准确率高；当迭代次数越低，卷积核区域越大的组合准确率高。

因此当选用较小的卷积核区域组合时，使用多次迭代能得到较好的效果，而使用较大的卷积核区域组合时，很少次数的迭代就可以得到较好的效果。

**5 结论（Conclusion）**

本文主要针对CNN的迭代次数和卷积核区域组合这两类超参数进行了文本分类实验分析。接下来我将总结从上述实验的主要发现，供其他文本分类CNN的学习者和使用者参考借鉴。

显然预训练词向量对于数据量较小的文本分类任务来说对模型的准确率能有很大的提升。

在CNN使用过程中，数据集的迭代次数（epoch\_size）会对模型的最终性能产生一定的影响。在确定其他参数不变的情况下，可以对数据集进行不同次数的迭代并测试最后的性能，每个神经网络常常都会有一个最合适的迭代次数来使得

模型的性能达到最佳。

同时对于不同大小的卷积核区域组合（kernel\_size），越大的卷积核区域往往经历较小的迭代次数就能使模型达到最好的效果；越小的卷积核区域往往经历较大的迭代次数后使得模型效果最好。故而当卷积核区域大小较小时，可以增加迭代次数以提高准确率；卷积核区域大小较大时，可以从迭代次数为1开始寻找最佳的迭代次数。

**参考文献（Reference）**

1. KIM Y. Convolutional Neural Networks for Sentence Classification[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. 2014. http://dx.doi.org/10.3115/v1/d14-1181. DOI:10.3115/v1/d14-1181.
2. A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification[J].
3. <https://zhuanlan.zhihu.com/p/77634533>
4. <https://blog.csdn.net/huwenxing0801/article/details/85197722>
5. <https://juejin.cn/post/6844903557410324494>
6. <https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526>
7. <https://zhuanlan.zhihu.com/p/39784226>
8. <https://www.mindspore.cn/tutorials/application/zh-CN/r2.0/nlp/sentiment_analysis.html?&highlight=RNN#dense>（MindSpore的nn.Embedding层导入预训练词向量实现细节）