自然语言处理系列（6）：TextCNN

这篇文章翻译自卷积神经网络用于文本分类的开山之作Convolutional Neural Networks for Sentence Classification，并相应的对其实现代码进行讲解。

我们进行了一系列关于卷积神经网络(CNN)的实验，这些实验是基于预先训练的词向量训练的，用于句子级别的分类任务。我们得到了一个简单的参数微调的以及静态矢量的CNN在多个基准上取得了优异的结果。通过微调学习任务特定的向量可以进一步提高性能。我们还建议对体系结构进行简单的修改，以允许使用特定于任务的和静态的向量。CNN在这里讨论了7项任务中4项的改进，包括情绪分析和问题分类。

一、导语

近年来，深度学习模式在计算机视觉(Krizhevsky et al.， 2012)和语音识别(Graves et al.， 2013)中取得了显著的成绩。在自然语言处理中，许多深度学习方法都涉及通过神经语言模型学习词向量表示(Bengio et al.， 2003;Yih et al .,2011;Mikolov et al.， 2013)，并通过学习的词向量进行分类(bert et al.， 2011)。词向量，在单词中从一个稀疏的1- V编码(这里V是词汇量)到一个较低维度的向量空间，通过一个隐藏层，本质上是特征提取器，在它们的维度中编码词汇的语义特征。在这样稠密的表示法中，语义上相近的词在欧几里得或余弦距离上也同样接近（低维向量空间）。

卷积神经网络(CNN)利用具有卷积滤波器的层来应用于局部特征(LeCun等人，1998)。开端于计算机视觉的CNN模型后来被证明在自然语言处理中也很有效，在语义解析(Yih et al .,2014)、搜索查询检索(沈et al .,2014)、句子建模(Kalch -布伦纳et al .,2014)和其他传统NLP任务(Collobert et al .,2011)都取得了非常好的效果。

在目前的工作中，我们训练了一个简单的CNN，它只有一层卷积层，前面连接由无监督的神经语言模型训练的的词向量。这些向量是由Mikolov等人(2013)在谷歌新闻的1000亿字上进行训练得到，并已经公开了。我们首先保持向量静态，只学习模型的其他参数。即使对超参数进行了微调，这个简单的模型在多个基准上也都取得了出色的结果，这表明预先训练的词向量是“通用”的特征构造器，它们可以用于各种分类任务。通过微调，学习特定任务的向量可以得到进一步的改进。我们最后描述了对体系结构的一个简单的修改，允许使用多个通道来使用预先训练的和任务特定的向量。

我们的工作与Razavian et al.(2014)类似，它表明，对于图像分类，从预先训练的深度学习模型获得的特征提取器在各种任务中都表现良好，包括与最初的任务不同的任务，这些任务的特征提取器是经过训练的。

二、模型

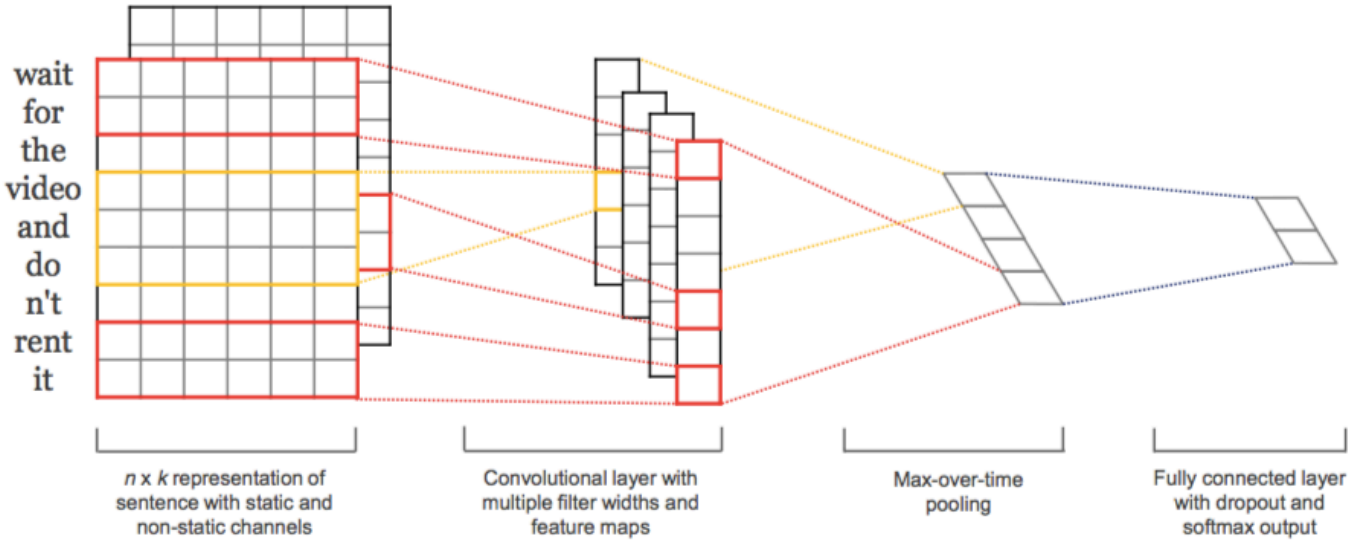

Figure 1: Model architecture with two channels for an example sentence.

图1所示的模型架构是CNN架构的一个小变体。$x_i \in R^k$表示一个句子中第i个词的k维词向量。一个长度为n的句子（可以padding）表示为

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \quad (1)$$

其中$\oplus$是连接操作符。一般来讲，$x_{i:i+j}$表示词$x_i, x_{i+1} \cdots , x_{i+j}$的连接。卷积运算涉及到一个滤波器，它被应用到一些单词的窗口，以产生一个新特征。比如，特征c_i产生于一个词汇窗口$x_{i:i+h-1}$，计算公式为：

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad (2)$$

$b \in R$是偏置项，f是一个非线性函数,例如双曲正切函数。该滤波器应用在句子中$x_{1:h}, x_{2:h+1}, \ldots , x_{n+h-1:n}$每一个可能的单词窗口从而产生一个特征映射：

$$c = [c_1, c_2, \ldots , c_{n+h-1}]$$

这里$c \in R^{n-h+1}$。然后我们对特征映射采用最大池化策略（Collobert et al.， 2011）即取最大的值$\hat{c} = max c$作为对应此滤波器的特征。此思路是去捕获最重要的特征——每个特征映射中最大的值。最大池化可以处理不同的句子长度。

我们已经说明了通过一个滤波器抽取一个特征的过程。当然使用多个滤波器（不同的窗口大小）可以获取多个特征。这些特征组成了倒数第二层并且传给全连接的softmax层，输出标签的概率分布。

在其中一个模型变种中，我们做了将词向量分两个“通道”的实验，一个通道中的词向量在模型训练的过程中保持不变，另一个通过BP算法（3.2节）进行细粒度的调节。在多通道架构中，如图1所示，每个滤波器应用在两个通道，再把结果加起来，然后用等式（2）计算c_i。除此之外模型等价于单通道的架构。

2.1 正则化

为了解决过拟合的问题，我们采用正则化的方法，在倒数第二层我们使用dropout机制，并且使用L-2范数来约束权重向量。D ropout机制可以防止隐藏层的过拟合，通过随机的dropout：例如在反向传播的过程中将p部分的隐藏单元设置为 0 。即，已知，倒数第二层的特征向量z=[C1，……，Cm] (表示这里我们有m个滤波器)。经典的情况神经元的输出应该是：

$$y = w \cdot z + b(4)$$

在前向传播中对输出单元y，dropout使用的是：

$$y = w \cdot (z \circ r) + b \ (5)$$

这里◦表示按元素逐个相乘操作，$r \in R^m$是一个“掩盖”向量，向量中的元素都是一个伯努利随机变量，有p的概率变为1。梯度仅仅可以通过非掩盖的单元反向传播。在测试阶段，权重向量通过因子p缩减例如$\hat{w} = pw$,并且$\hat{w}$被用来（没有使用dropout）给掩盖的句子打分。我们另外限制权重向量的二范式，在每一步梯度下降之后，如果$\|w\|_2 > s$，重新将w的二范式设置为$\|w\|_2 = s$。

三、数据集和实验步骤

Data	c	l	N	$\|V\|$	$\|V_{pre}\|$	*Test*
MR	2	20	10662	18765	16448	CV
SST-1	5	18	11855	17836	16262	2210
SST-2	2	19	9613	16185	14838	1821
Subj	2	23	10000	21323	17913	CV
TREC	6	10	5952	9592	9125	500
CR	2	19	3775	5340	5046	CV
MPQA	2	3	10606	6246	6083	CV

表1：分词之后的数据集的简要统计。C：目标类的个数。l：平均句子长度。N：数据集大小。|V|：单词总数。|V_pre |：出现在预训练词向量中词的个数。Test：测试集的大小（CV意味没有标准的训练/测试集并且采用十折交叉验证的方法）

我们在不同的基准上测试我们的模型。数据集的简要统计如表1所示。

- MR: 一句话的电影评论。分类标签分为积极/消极的评论（Pang and Lee, 2005）。
- SST-1：斯坦福情感树库——MR数据的扩展，但是包含train/dev/test数据集的划分及细粒度的标签（非常积极、积极、中立、消极、非常消极），被Socher et al.（2013）重新标记。
- SST-2: 和SST-1一样，但是没有中立的评论，只有积极和消极两种标签。
- Subj：主观数据集，任务是去划分一个句子是主观性的还是客观性的（Pang and Lee. 2004）。
- TREC：TREC问题数据集——任务涉及到将一个问题划分为六种问题类型（人，位置，数值信息等）（Li and Roth, 2002）。
- CR：不同产品（照相机、MP3s等等）的客户评论。任务是预测积极/消极的评论（Hu and Liu, 2004）。
- MPQA：MPQA数据集（Wiebe et al., 2005）的观点极性检测子任务。

3.1 超参数和训练过程

我们在所有的数据集中都采用下列参数：

- 对于所有的数据集我们使用修正线性单元(Rectified linear units)；
- 滤波器窗口大小h为3、4、5， 每种滤波器个数为100。；
- dropout的比例为0.5；
- l2正则化限制权值的大小为 3；
- mini-batch大小为50；
- 这些值都是在SST-2 验证数据集上通过网格搜索选择得到的。

我们除了在验证集上进行early stpping外没有另外进行任何特定数据集的调节，对于没有标准验证集的数据集，我们从训练数据集中随机选择10%的数据作为验证集。通过采用Adadelta更新参数（Zeiler, 2012）及随机mini-batches策略的随机梯度下降算法进行训练。

3.2 预训练的词向量

在没有大量监督训练集（Colobert et al., 2011; Socher et al., 2011; Iyyer et al., 2014）的情况下，使用从非监督神经语言模型训练得到的词向量进行初始化是用来提升结果的流行方法。我们使用公用的、由10亿Google 新闻数据中训练出来的Word2vec词向量。此向量的维度是300并且是采用连续的词袋架构（Mikolov et al., 2013）训练出来的。而将没有出现在预训练词向量中的单词随机初始化。

3.3 模型变种

我们使用以下模型变种进行实验。

- CNN-rand：我们的基准模型，所有的词被随机初始化，并在训练的工程中进行调节。
- CNN-static：使用预训练的词向量——Word2vec。所有的词——包括随机初始化的未出现在预训练词向量中的词——保持不变而仅仅调节模型其它的参数。
- CNN-non-static：和CNN-static相似，但是预训练的词向量在每个任务中会有细粒度的调节。
- CNN-multichannel：有两个词向量集合的模型。将每个向量集合看作一个“通道”并且每个滤波器应用在所有的通道，但是梯度只能通过其中一个通道进行反向传播。因此，模型能够细粒度的调节其中一个向量集合，而保持另外一个不变。

为了探究上述变种对其它随机因子的影响，我们消除了其它随机化的影响——交叉验证次数赋值，未知词向量的初始化，CNN模型参数的初始化——在每个数据集上保持一致。

四、结果和讨论

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

- RAE：由维基百科(Socher et al.， 2011)中得到预先训练的词向量的递归自编码器。
- MV-RNN：句法分析树的矩阵-向量递归神经网络(Socher et al.， 2012)。
- RNTN：递归神经张量网络，具有基于时态的特征函数和句法分析树(Socher et al.， 2013)
- DCNN: k-max汇聚的动态卷积神经网络(Kalchbrenner et al.， 2014)

- Paragraph-Vec：在段落向量之上的逻辑回归(Le and Mikolov, 2014)
- CCAE: 组合类自动编码器与组合类语法运算符(Hermann and Blunsom, 2013)
- Sent-Parser::情绪分析特定解析器(Dong et al.， 2014)
- NBSVM, MNB: 朴素贝叶斯SVM和多项朴素贝叶斯。
- G-Dropout, F-Dropout: 高斯Dropout和快速Dropout
- Tree-CRF: 条件随机场依赖树(中川等，2010)
- CRF-PR：后正则化条件随机场(Yang and Cardie, 2014)
- SVMS: SVM使用uni-bi-trigrams、wh word、head word、POS、parser、hypernyms和60个手工编码的规则作为特性。

表2为我们模型和其它模型的结果对比。我们随机化所有词向量的基准模型（CNN-rand）就其本身而言表现的不是很好。然而我们期望通过预训练的词向量来提升效果的模型，我们对效果提升的幅度感到很吃惊。即使使用静态向量的简单模型（CNN-static）表现的相当好，产生了与利用复杂池化模式的复杂深度学习模型（Kalchbrenner et al., 2014）和需要提前计算解析树的模型（Socher et al., 2013）可抗衡的结果。这些结果说明预训练的词向量是好的，它可以作为“通用”的特征抽取器，并且可以跨数据集使用。对每个任务细粒度的调节词向量可以进一步提升结果（CNN-non-static）。

4.1 多通道与单通道模型比较

我们一开始期望多通道的架构可以避免过拟合（通过确保学到词向量不会偏离初始值太远），并且比单通道的模型效果好，尤其在更小的数据集上。然而，结果是含混的，进一步的微调工作是有必要的。例如，与其使用非静态部分的额外通道，还可以维护单个通道，但使用额外的维度，在训练期间允许修改。

4.2 静态与非静态表示比较

与单通道非静态模型一样，多通道模型能够对非静态通道进行微调，以使其更特定于任务。例如，good在word2vec中最类似于bad，大概是因为它们(几乎)在语法上是等价的。但是对于在SST-2数据集上的非静态信道中的向量来说，情况并不是如此(表3)。同样地，可以说在表达情感上，nice和great相比，与good更相近，这在训练的词向量中被真实的反映出来。

对于（随机初始化）不在预训练向量集合中的词，细粒度的调节允许它们学到更有意义的表示：网络训练得到感叹号经常与热情洋溢的表达联系在一起，并且逗号经常和连接副词联系在一起。

4.3 进一步观察

我们做了进一步的实验和观察。

- Kalchbrenner et al.（2014）使用和我们单通道架构相同的CNN架构，但是却得到了不好的

结果。例如，他们采用随机初始化词向量的Max-TDNN（时间延迟神经网络）模型在SST-1数据集上获得的结果是37.4%，和我们模型获得的45%相比。我们将此差异归因于我们的CNN有更大的容量（多个滤波器和特征映射）。

- Dropout是如此好的正则化方法以致于用在一个比必要网络更大的网络上并且仅仅使用dropout去正则化效果是好的。
- Dropout一般可以将结果提升2%-4%。
- 当随机初始化不在Word2vec集合中的单词时，向量的每一维从U[-a,a]采样，结果获得了微小的提升，这里的a的选择要使随机初始化的词向量的方差和预训练的词向量有相同的方差。在初始化的过程中，尝试应用更复杂的方法反映预训练词向量的分布，看能否使结果得到提升是很有意思的事情。
- 我们在Collobert et al.（2011）通过维基百科训练的、另一个公用的词向量上进行试验，并且发现使用Word2vec的结果要远远优于它。目前不清楚是因为Mikovlov et al.（2013）的架构还是因为10亿的Google新闻数据集。
- Adadelta（Zeiler，2012）和Adagrad（Duchi et al., 2011）的结果相似，但是需要更少的训练次数。

五、结论

在这篇论文中，我们使用卷积神经网络和Word2vec进行了一系列的实验。尽管微小的超参数调节，具有一个卷积层的简单CNN就表现的非常好。我们的实验再次证明：将深度学习应用在NLP领域，非监督预训练的词向量是一个非常重要的因素。

六、代码解释

待完成