# 语言分析和机器翻译报告

## ——比较 RNN 和 LSTM 在语言模型中的效果

计硕 2007 黄灿安 2071758

摘要：语言模型是自然语言处理中一个非常重要的技术，它在机器翻译、语音识别等任务中有着广泛的应用。过去的语言模型通常是基于规则或是基于统计的模型，其中基于统计的 n-gram 语言模型由于较好的性能，被用于许多实际的应用中。随着神经网络的发展，不同的神经网络如 RNN、LSTM 也被用于语言模型的实现，并表现出比 n-gram 语言模型更出色的效果。本报告分别基于 RNN 和 LSTM 构建语言模型，用困惑度作为评价指标，对比两个模型在同一语言模型数据集 PTB 上的效果。

## 一、背景介绍

语言模型（Language Model）是自然语言处理中重要技术。自然语言处理中最常见的数据是文本数据。我们可以把一段自然语言文本看作一段离散的时间序列。假设一段长度为 T 的文本中的词依次为 $W_1$，$W_2$，…，$W_T$，那么在离散的时间序列中，$W_t$（$1 \le t \le T$）可看作在时间步（time step）t 的输出或标签。给定一个长度为 T 的词的序列 $W_1$，$W_2$，…，$W_T$，语言模型将计算该序列的概率：

$$\mathrm{P}(W_1, W_2, \ldots, W_T)$$

语言模型的计算可以定义为，假设单词序列 $W_1$，$W_2$，…，$W_T$ 中的每个词是依次生成的，给定前 T-1 个单词，可以计算出下一个单词的条件概率分布，计算公式如下：

$$\mathrm{P}(W_1, W_2, \ldots, W_T) = \prod_{t=1}^{T} \mathrm{P}(W_t \mid W_1, W_2, \ldots, W_{t-1})$$

语言模型经常使用在许多自然语言处理方面的应用，如语音识别，机器翻译，词性标注，句法分析和资讯检索等。过去的语言模型按照所使用的技术方法不同通常分为两类：基于规则的语言模型和基于统计的语言模型。基于规则的语言模型又称文法型语言模型，用人工来编制语言学文法，文法规则来源于语言学家掌

握的语言学知识和领域知识。后来以统计学为基础的统计语言模型广泛使用，包括 n-gram 语言模型、指数语言模型、基于神经网络的语言模型等。

当序列长度增加时，计算和存储多个词共同出现的概率的复杂度会呈指数级增加。n 元语法通过马尔可夫假设简化了语言模型的计算。这里的马尔可夫假设是指一个词的出现只与前面 n 个词相关，即 n 阶马尔可夫链。如果 n=1，那么有 $P(W3 \mid W_1, W_2) = P(W_3 \mid W_2)$。基于 n−1 阶马尔可夫链，我们可以将语言模型改写为：

$$P(W_1, W_2, \ldots, W_T) = \prod_{t=1}^{T} P(W_t \mid W_{t-(n-1)}, W_2, \ldots, W_{t-1})$$

以上也叫 n 元语法（n-grams）模型。当 n 分别为 1、2 和 3 时，我们将其分别称作一元语法（unigram）、二元语法（bigram）和三元语法（trigram）。n-gram 语言模型包含了前 n-1 个词的全部信息，可解释性强，直观易理解。但其缺点也非常明显：只能建模到前 n-1 个词，无法利用上下文的信息；随着 n 的增大，参数空间呈指数增长；数据稀疏，测试集中会出现训练集中未出现过的词，导致语言模型计算出的概率为零的问题。

近年来，神经网络在各个任务中都取得了不错的效果，利用神经网络来构建的语言模型也取得了很大的进步。一个语言模型中比较常用的神经网络是循环神经网络 RNN。最开始将 RNN 引用语言模型是为了解决定长信息的问题，即使得模型能够尽可能地保存前文序列的所有信息，它的原理是通过不断的应用同一个矩阵 W 可实现参数的有效共享，从而可以存储之前时间步的信息。RNN 模型的优势是，可以处理任意长度的输入；理论上 t 时刻可以利用之前很早的历史信息；对于长的输入序列，模型大小并不增加；每一时刻都采用相同的权重矩阵，有效的进行了参数共享。其缺点在于需要顺序计算而不是并行计算，计算较慢；而且在实际上由于梯度消失等问题距离较远的早期历史信息难以捕捉。长短期记忆模型（LSTM）是 RNN 模型的一种特殊结构类型，其增加了输入门、输出门、忘记门三个控制单元（“cell”），随着信息的进入该模型，LSTM 中的 cell 会对该信息进行判断，符合规则的信息会被留下，不符合的信息会被遗忘，以此原理，可以解决神经网络中长序列依赖问题。

衡量一个语言模型的好坏，主要是使用困惑度（Perplexity，简写为 PPL）这

个指标。困惑度用于度量概率分布或概率模型的预测结果与样本的契合程度，困惑度越低则契合越准确。报告的后面部分将给出困惑度的计算方法。

本报告基于以上背景，利用 RNN 和 LSTM 两种神经网络构建语言模型，在语言模型数据集 PTB 上进行训练和测试，并计算出相应的困惑度大小，比较两个神经网络语言模型的效果。

## 二、准备工作

为了分别用 RNN 和 LSTM 实现语言模型，本报告在 Tae Hwan Jung (Jeff Jung) @graykode 等人工作的基础上完成。Tae Hwan Jung 等总结了自然语言处理领域经典的神经网络模型，包括词嵌入模型（Basic Embedding Model）、卷积神经网络（CNN）、循环神经网络（RNN）、注意力机制（Attention Mechanism）和 Transformer 模型，并基于 pytorch（1.0.0+）、Python（3.5+）实现，为自然语言处理的初学者提供了简单易懂的代码用于入门学习。该项目已经在 GitHub 上进行开源，可通过以下地址获得：https://github.com/graykode/nlp-tutorial。

本报告引用了该项目中基于 RNN 和 LSTM 实现的语言模型，在此基础上添加了句子预处理部分和计算困惑度部分。

## 三、实验介绍

为了比较 RNN 和 LSTM 对于语言模型的效果好坏，本报告在基本的 RNN 模型和 LSTM 模型基础上，修改了 batch 的生成部分，增加了计算困惑度的函数，修改了训练的次数，而神经网络模型部分并未作修改。

### 1、数据集

实验所使用的语言模型数据集是 PTB 数据集。PTB(Penn Treebank Dataset)文本数据集是目前语言模型学习中使用最为广泛的数据集。数据集中的数据已经过一定的预处理，未登录词都用<unk>进行替换，没有数字文本，相邻的单词之间用空格隔开。数据集中共包含了 9998 种不同的单词词汇。本实验将使用 PTB 数据集中的训练集和验证集。

2、数据预处理部分

借鉴 n-gram 语言模型的思想，为了减少计算量，实验将模型简化为一个“四元组”的句子结构，即给定前三个单词序列，通过模型预测出第四个单词。将 PTB 数据集中训练集和验证集的句子，通过预处理后，变为由一个数组的存储该训练集或验证集所包含的所有四元组句子，也就是句子中所有的四个前后相连的单词序列。该数组将用于模型的训练以及验证。

3、batch 生成部分

由于训练集中的四元组句子数量比较大，故每次选取数量大小为 1000 的四元组句子生成 one-hot 张量进行训练或验证。

4、计算困惑度（PPL）部分

困惑度是用来评价语言模型好坏的指标，本质上是计算句子的概率。

对于一段词语序列 W，其计算公式如下：

$$\mathrm{PPL(W)} = \mathrm{P}\left(W_1,\ W_2,\ \ldots,\ W_T\right)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{\mathrm{P}\left(W_1,\ W_2,\ \ldots,\ W_T\right)}}$$

不同模型的困惑度计算，由于对句子概率的计算方式的不同，具体公式也会不同。在基于神经网络实现的语言模型中，困惑度常常不是直接使用句子概率来计算的，而是使用了交叉熵（cross entropy）。假设每个句子算出的交叉熵为 $J_i$，那么困惑度的具体计算公式如下：

$$e^{\frac{1}{N}\sum_{i=1}^{N} J_i}$$

5、训练与验证

实验将分别利用基于 RNN 和 LSTM 的语言模型，在 PTB 数据集的训练集上进行训练，一共训练六轮，每轮将句子打乱后重新训练。然后在验证集上计算困惑度，将每一轮训练后得到的困惑度值进行比较，评估两个语言模型的效果好坏。

## 四、结果

本实验的代码基于 pytorch（1.4.0）、Python（3.5）的环境下实现。

通过计算两个模型在不同训练轮次的困惑度，得到如下表 1 和图 1 数据：

表 1 RNN 语言模型和 LSTM 语言模型在不同训练轮次下的 PPL 值

| Epoch | | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| PPL | RNN | 256.00 | 217.50 | 212.33 | 209.13 | 207.91 | 206.56 |
| | LSTM | 227.83 | 211.65 | 202.32 | 195.90 | 191.29 | 188.24 |

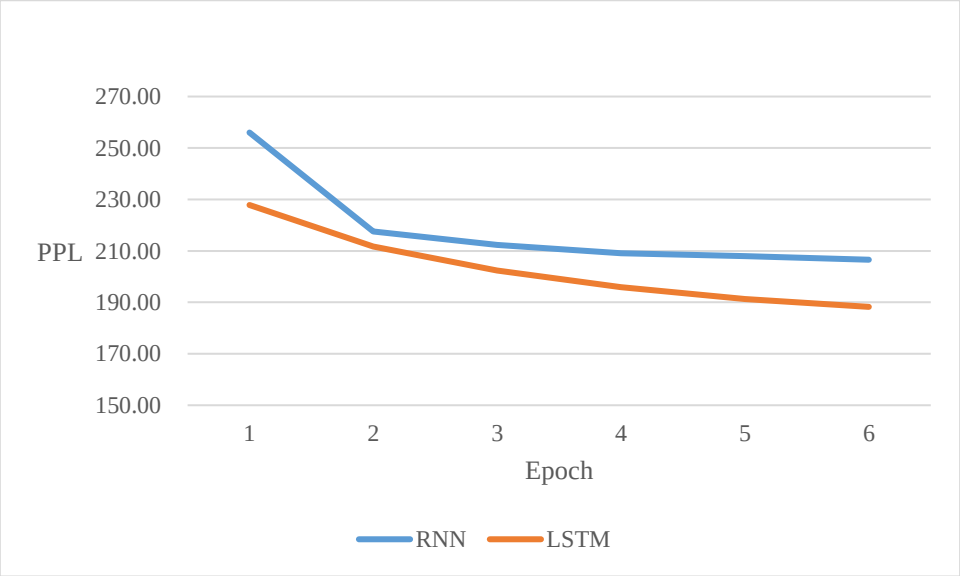


图 1 RNN 语言模型和 LSTM 语言模型在不同训练轮次下的 PPL 变化及对比

通过实验数据可以看到，随着训练次数的增加，连个模型的困惑度都有显著降低，说明模型预测的单词与正确的单词更加契合；随着训练轮次的增加，困惑度下降幅度减少，后面的训练对模型的优化效果不明显。对比两个模型，LSTM 语言模型的表现则要优于 RNN 语言模型。

## 五、结论

本报告分别实现了基于 RNN 和 LSTM 的神经网络语言模型，通过在同一语言模型数据集 PTB 上进行训练，并计算了不同训练次数的困惑度进行比较。实

验结果表明，LSTM 语言模型要优于 RNN 语言模型；模型的困惑度一开始显著下降，随着训练次数的增加，下降变得不明显。

## 六、不足与展望

本实验所使用的的数据集 PTB 由于数据量比较小，只能通过不断重复训练来优化模型，训练效果有限。利用数据量更大的数据集，能对模型的训练有更好的效果。另外，本次实验简化的模型的构建过程，对单词的预测只利用了前三个单词的信息，可能不能很好的体现出 LSTM 语言模型长序列依赖的优势，增加每个句子的长度能提高其预测的准确度。