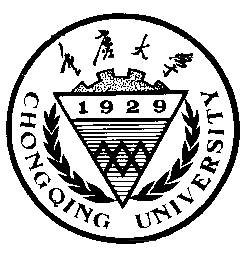
重庆大学本科学生毕业设计（论文）

基于RNN深度神经网络的预测模型研究



学 生：黄宇霄

学 号：20144487

指导教师：冯亮

专 业：计算机科学与技术

重庆大学计算机学院

二O一八年六月

**Graduation Design (Thesis) of Chongqing University**

**Research of the Prediction Model Based on Deep RNNs**



**Undergraduate: Huang Yu Xiao**

**Supervisor: Feng Liang**

**Major: Computer Science and Technology**

**College of Computer Science**

**Chongqing University**

**June 2018**

# 摘 要

近些年来，随着计算机硬件性能的提高（主要是Graphic Processing Unit, GPU性能的提升），人工神经网络这一机器学习的分支，再次火热了起来。有大量的专家学者，基于最基本的人工神经网络（Artificial Neural Networks, ANNs）,根据学习任务的多样性，研究并提出了大量的多种多样神经网络，例如卷积神经网络（Convolutional Neural Networks, CNNs）、循环神经网络（Recurrent Neural Networks, RNNs）、深度信念网络（Deep Belief Networks, DBNs）、生成对抗网络（Generative Adversarial Networks, GANs）、深度残差网络（Deep Residual Networks, DRNs）以及它们的变种。神经网络如高速列车般快速发展，然而，随着网络的复杂度越来越高，参数越来越多，通过调整参数来使某个网络模型适应于某一特定问题变得越来越困难。把各个能使得神经网络表现的最好的参数组合起来，不一定能取得全局最好的结果。对于RNNs来说，暂时没有一个能在各种问题上都表现良好的固定的RNN结构，一旦当处理的问题改变，则需要很多经验丰富的研究者调整RNN结构及其参数。然而人为调整具有人力成本高、时间成本高的局限性。针对该问题，很多文献已经提出了针对不同问题，自动学习与调整RNN结构的智能算法。但是，他们大多关注于调整局部的网络结构或参数，少有从整体上调整网络结构且取得较大成果的文献。本文基于这个问题，通过三个长短期记忆网络（Long Short-Term Memory, LSTMs，RNNs的一种变种）模型，探讨模型的结构对其在各问题的效果的影响，为RNN的参数和整体结构配置的智能优化设计提供了更加深入的见解。

**关键词**：神经网络、长短期记忆网络（LSTMs）、结构

# ABSTRACT

Recently, with the development of computer hardware devices (e.g., graphic processing unit, GPU), artificial neural network (ANNs) which is a branch of the Machine Learning, becomes popular again. There are many different neural network architectures for solving various real world learning problems have been proposed in the literature, such as convolution neural networks (CNNs), circulation neural networks (RNNs), deep belief networks (DBNs), the generated against network (GANs), the depth of the residual network (DRNs) and their variants, etc. The development of new neural network techniques is extremely fast to date. However, due to the huge number of parameters and high complexity of the deep neural network, it is very hard to obtain the optimal parameter and structure settings of a deep neural network when a particular problem is encountered. Simply combining the parameter configurations which possess good performance separately, may not lead to an expected performance. For the RNNs, there is no single structure of RNN that performs well on all the problems of interests. Once the problem changed, the trained RNN model requires extra efforts for adjusting the corresponding structure and parameter settings. Nevertheless, adjusting the configurations of RNN by users is time consuming and heavily relies on expert knowledge. To solve this problem, much research work have been conducted to investigate intelligent algorithms that learn to adjust structure of RNN for different problems automatically. However, most of them focus on adjusting the local network structure or parameters, no or only a little work can adjust the global network structure and parameter settings of the RNN. In this paper, we thus discuss and investigate how the structure and parameter settings affect the performance of RNN by three long short-term memory network (LSTMs, a variant of RNNs) models. This study could provide deeper insight of the design of intelligent optimization for adapting the parameter and structure configuration of RNN globally.

**Keywords**: neural networks, long short-term memory networks (LSTMs), structure

目 录

[摘 要 I](#_Toc513574287)

[ABSTRACT II](#_Toc513574288)

[1 绪论 1](#_Toc513574289)

[1.1 研究背景和现状分析 1](#_Toc513574290)

[1.2 研究目的和意义 2](#_Toc513574291)

[2 循环神经网络 4](#_Toc513574292)

[2.1 循环神经网络（RNNs） 4](#_Toc513574293)

[2.2 长短期记忆网络（LSTMs） 4](#_Toc513574294)

[2.3 门控制单元（GRU） 6](#_Toc513574295)

[3 不同应用的循环神经网络（RNNs）模型 7](#_Toc513574296)

[3.1 文本生成 7](#_Toc513574297)

[3.2 seq2seq模型 7](#_Toc513574298)

[3.3 图像识别模型 8](#_Toc513574299)

[4 研究方案设计 9](#_Toc513574300)

[4.1 文本生成 9](#_Toc513574301)

[4.2 seq2seq模型 9](#_Toc513574302)

[4.3 图像识别模型 10](#_Toc513574303)

[5 实验 11](#_Toc513574304)

[5.1 实验数据 11](#_Toc513574305)

[5.2 实验方法 11](#_Toc513574306)

[5.2.1 文本生成 11](#_Toc513574307)

[5.2.2 seq2seq模型 11](#_Toc513574308)

[5.2.3 图像识别模型 12](#_Toc513574309)

[5.3 实验环境 12](#_Toc513574310)

[5.4 实验结果及分析 12](#_Toc513574311)

[5.4.1 文本生成模型 12](#_Toc513574312)

[5.4.2 序列转化模型 14](#_Toc513574313)

[5.4.3 图像识别模型 16](#_Toc513574314)

[6 总结与展望 17](#_Toc513574315)

[6.1 总结 17](#_Toc513574316)

[6.2 展望 18](#_Toc513574317)

[致 谢 19](#_Toc513574318)

[参考文献 20](#_Toc513574319)

[附 录 22](#_Toc513574320)

# 1 绪论

## 1.1 研究背景和现状分析

人工智能（Artificial Intelligence，AI）是一门源远流长的学科，它也是计算机领域最前沿并最神秘的学科之一。从20世纪中期发展至今。它致力于用机械符号处理的观点来解释人类自身思考并做出反应的过程。纵观其发展历程，人工智能的探索之路充满了坎坷。在上世纪60年代的启蒙阶段，各界人士对这个新的领域报以极其乐观的状态，但是在经过了短暂的十多年发展后，因为该领域没有什么产出，因此人工智能首次进入寒冬。这主要是因为它需要大量的计算资源，然而在当时，硬件满足不了它的计算需求。直到上世纪80年代人们意识到软件以及算法层面对它的发展有着极其重大的影响。到上世界90年代后期，因为计算机硬件性能的提升，AI的计算需求得到满足，终于，AI卷土重来。到目前为止，在学术界，有关AI的文章占了每年发表的文献的很大部分，人工智能的概念几乎是家喻户晓了。

近年来，国内外专家学者都希望能制造出能代替人类思考的智能机器。以AI为主题的小说和电影也层出不穷，这是现在人门对未来AI的期望。最近几年，由于计算机硬件，特别是GPU的快速发展，让深度学习（Deep Learning）这一机器学习（Machine Learning）的分支有了突破性进展。而深度学习的火爆，又让全世界的目光聚焦到人工智能中。深度学习不仅具有很大学术意义，而且也有很大的实用意义。最近，工业界也对这一领域投入了大量财力物力，有大量的基于深度学习的AI产品已经给人们带来了很多便利。

2006年，机器学习大师Hinton[1]提出逐层贪心训练法，该训练深度网络的方法是深度学习领域的专家和学者的曙光。2012年，机器学习的登记学者Andrew Ng和分布式系统顶级专家Jeff Dean联合打造Google Brain项目。同年，微软首席研究官Rick Rashid在21世纪的计算大会演示了一套自动同声传译系统。2013年，Google收购了一家研究深度神经网络(Deep Neural Network, DNN)的初创公司，这家公司由Hinton和他的两个学生组成。Google收购这家公司是希望Hinton可以将深度学习利用在Google未来的核心技术里。同年，深度学习专家Yann LeCun加盟Facebook，负责深度学习的研发工作。也是这一年，百度成立深度学习研究所（Institute of Deep Learning, IDL）。

自从Hinton [1]后，全世界掀起了对深度学习的研究热潮。从之前提到的信息中可以看出，近年来各大巨头公司都在争相跟进，将顶级的深度学习人才从学术界抢到工业界。而这一切都标志着，深度学习不再是一个很深奥难懂的概念，而是成为了一种实用的技术。目前，深度学习在几个主要领域都取得了突破性的进展。许多人在自然语言处理（Nature Language Process, NLP）这种很难的问题上，逐渐引入了深度学习的方法，并取得了很大成功，达到了很棒的结果。当然，者也归功于近年来各大专家学者的不懈努力。在附录中，表[1]列举了近年RNN发展的重大突破，以及做出贡献的专家学者。

随着近年来深度学习的突破式进展，可以预见，曾经只出现在电影和小说中的那些未来生活中的人工智能不再是南柯一梦，在不久的将来，人们可能不在需要司机，道路上所有由AI控制的车辆来来往往有秩序的行驶，出行在国外，可能某个设备能直接将语言翻译成当地用语，让语言不再是外出交流的障碍，在医院里，人们可能只需要扫描一次身体，就能检查出身体各种情况，包括正常和异常的部分，在家，忘了钥匙也不怕，可能只需要面部扫描，房门就能打开，同时，家里各种自动化的电器开始运作，出门后，家里的电器自动关闭。这将是未来人工智能带给人类生活品质上的提升。

## 1.2 研究目的和意义

深度学习的基础是人工神经网络（ANN），ANN由多层相互连接的神经元（Cell）单元构成。浅层的神经网络指的是不包含循环连接，且只有一个输入层、一个输出层、最多一个隐藏层的网络。为了让网络拟合数据，一般来说可以增加隐藏层的神经元数。但是，这样会导致过拟合(Overfitting)的问题。并且，大量的实验发现，这种单纯增加神经元个数的方法，不适合比较复杂的任务。目前已经有许多的论文发现，随着网络层数的增加，网络复杂度也增加，从而使得网络能提取更高维度、更多层次的数据特征。例如，对于人像识别的任务，层数较低的网络可能只提取到每个线条特征。通过大量的学习后，这个低层模型能通过这些固定的线条的排列组合判定人像。但是，当另一张图片中的人的线条很不一样时，要判定成人像就非常的难。而对于高层的网络，在某一层可能会提取到两只手、两只腿、躯干、头的信息，而基于这些高层次的信息，再判定这是人像，就非常的容易了。这也就是深度学习的真正目的。

一般来说，神经网络由非线性的、简单的单元构成，更深的隐藏层提供更抽象的数据特征，并丢掉一些不重要的特征。由于每个层是非线性的结构，训练变得十分困难。直到出现重大进展以前，关于深度网络架构的研究工作都比较少。

结合了循环连接的ANNs被称作RNNs，其独特的循环结构特别适用于序列化的数据。多数RNNs由高维的隐藏层状态和非线性的激活函数组成。也就是因为这样的结构，RNNs才得以存储、记忆、处理序列化的复杂信息。循环神经网络（RNNs）已经在实践中被证明它对于NLP问题是很有效果的，Socher[2]提出了一种把递归神经网络（RNN）用于自然场景和语言解析任务的方法，Mikolov[3]提出了一种基于循环神经网络的语言模型。

RNNs可以很好的应用在语言模型和文本生成上。给定一个单词序列，我们可以根据之前的单词来预测下次出现的单词。这些模型能预测一个句子正确的概率，这是机器翻译的一部分。一般来说正确预测的概率越大，预测出来的句子就越正确。在机器翻译问题中，RNNs需要先将原语言的语句序列输入，再输出翻译后的句子信息。也就是说，必须先从完整的输入序列获取了所有信息，再开始输出。

RNNs能从序列和时序数据中学习特征和长期依赖关系。训练好的RNN可以对任何动态系统进行建模。但是，训练RNN主要受长期依赖的问题的影响。对于这个问题，Sepp Hochreiter 和 Jürgen Schmidhuber等[4]提出了基于RNN的一种网络，那就是大名鼎鼎的（长短期记忆）LSTM网络。

LSTM能解决梯度消失的问题，这是之前影响RNNs效果的核心因素之一。LSTM还可以保留误差，用于沿时间和层进行反向传递，这使得RNN学习远距离词组的因果关系变得可能。LSTM把控制信息存放在网络正常信息流之外的门控单元中。这些单元可以控制存储、写入或读取信息，以及控制何时允许读取、写入或清除信息。这些门依据接收到的信号而开关。

总的来说，LSTM的提出是RNN研究史上的一个重大突破。但是，LSTM的结构比较复杂，收敛速度比较慢。为了更好的解决这个问题， GRU（Gated Recurrent Unit）应运而生。GRU和LSTM很像，都通过门将重要特征保留。GRU简化了LSTM的结构，参数更少，训练速度会快一些，是一种使用很频繁的变体。

从近年来发表的关于神经网络的文献来看，大多数的文献中神经网络的结构和参数，都是为了让网络模型在一定的数据集上表现出色而特别设定的，这些文献的相同点都是把参数或结构固定，并且对于每个参数或结构并没有做太多的解释，但从结果上来说是比较好的模型，总的来说就像是一个黑盒。当然，因为网络变得越来越复杂，也有许多专家学者开始考虑用自动优化的方法来确定神经网络的结构或者参数。Becker等[7]利用基因算法搭建系统来帮助编程，Liu等[13]用多目标优化的方法来优化深度网络的隐藏层中连接的结构，Zhang等[14]为了解决剩余使用寿命预测问题（Remaining Useful Life, RUL），将各深度信念网络的预测准确率和其整个网络集合的多样性作为两个目标，进行优化。但是这些文献中，并不是专注于优化神经网络的整体结构，都是在局部上让神经网络自我调整结构去适应问题。

LSTM的网络结构比较简单，网络的核心部分都在LSTMs中的隐藏层神经元（Cell）中，LSTM各种网络的变种也是在基于Cell做的变化。因此，LSTM的整体网络结构的变化就比较简单，都是基于层数，每层Cell数，Cell种类做的改变。本文中，设置的三个模型都是基于层数、每层Cell数以及核心种类做出改变，探究LSTM网络模型在各个问题上的表现。当然，因为问题规模以及数据集规模、问题难度等因素不同，各个模型结构的差异还是比较大的。本文着重于找出共同的一种规律，为使用基因自动优化LSTM网络结构做前言的学习以及探索。

接下来，本文第二节将详细的介绍现在网上常用的循环神经网络，从最基本的Simple RNN开始，到改进RNN提出的LSTM以及LSTM的衍生网络GRU。文中第三节介绍了本文实验中所用到的三个基于LSTM的网络模型，包括文本生成、seq2seq、图像识别模型。文中第四节介绍三种模型的搭建过程以及如何对每个模型做改变，第五节基于上述模型进行实验，并且对实验的结果进行了分析，第六节则是本文的结论和展望部分。

# 2 循环神经网络

## 2.1 循环神经网络（RNNs）

循环神经网络(RNNs)已经在众多的实际问题例如自然语言处理、图像处理、金融预测等中取得了巨大的成功及广泛应用。由近年来发表的文献可以看到，RNNs有着许多各种不同的结构设计。其中最简单的Simple RNN包含输入层、输出层和隐藏层。为了解决多种多样的序列问题，RNN的隐藏层的结构显得尤为重要。隐藏层单元的输入包括上一隐藏层的状态，隐藏层内的节点也相互连接或者自连，其大致的网络结构如图1：

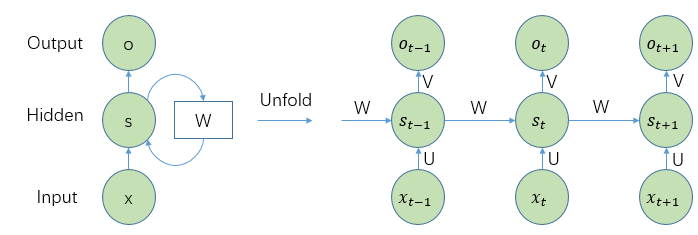


图1：RNN网络模型沿时间的展开图

依据图1，我们使用简洁的公式来表示Simple RNN正向传播：

公式(1)到(2)中，是输入层到隐藏层的参数矩阵，是隐藏层到隐藏层的参数矩阵，是隐藏层到输出层的参数矩阵。是激活函数。每个时间步，参数共享。对于NLP这类问题，参数共享有着很大的意义。其不仅减少了极大地计算开销，也更适合挖掘一些共同的规律。

RNNs有很多不同的架构，一般的，有1对N、N对1、N对N、N对M模型。种类繁多的架构也使得RNNs应用灵活广泛。

训练RNN和训练传统的ANN类似，使用BP误差反向传播算法。不过有一点区别。若将RNN按照时间步展开，那参数是共享的。使用梯度下降时，每一步的输出不仅依赖当前时间步的网络的状态，并且还需要用到前面若干时间步网络的状态。例如，在时，我们还需要向后传递两步，该算法被称（Backpropagation Through Time, BPTT）。普通的RNNs结构的BPTT会带来梯度消失的问题，当然也有许多方法解决该问题，例如将RNN隐藏层核心换成LSTM核心。

## 2.2 长短期记忆网络（LSTMs）

RNNs特有的结构设计使得网络的记忆具有持久性，但是随着时间步增长，梯度逐渐降低，甚至接近0，因此导致RNNs难以学到长期依赖。为了解决这个问题，LSTM在RNN的基础上增加了几个门控制单元，用来处理控制信息流动。Jozefowicz等 [6] 在实验中证明LSTMs在许多任务中都比Simple RNN更加出色。几乎所有的基于RNN的极优的结果都是LSTMs取得的。

LSTM也有链状的结构，但是重复的模块构造不同。LSTM重复模块中有4个神经网络层，且它们之间的交互很特别。

LSTM的核心是细胞状态(Cell State)，LSTM第一步就是决定要从细胞状态中扔掉哪些信息。这个操作由一个叫做“遗忘门(Forget Gate)”的函数层控制。下一步是决定把哪些新信息存储到细胞状态中。这由一个叫做“输入门(Input Gate)”的函数层控制。接下来一个函数创造了一个新的候选值，该值可能被加到细胞状态中。最后由一个叫做“输出门(Output Gate)”的函数层控制输出部分，LSTM的Cell图解如下：

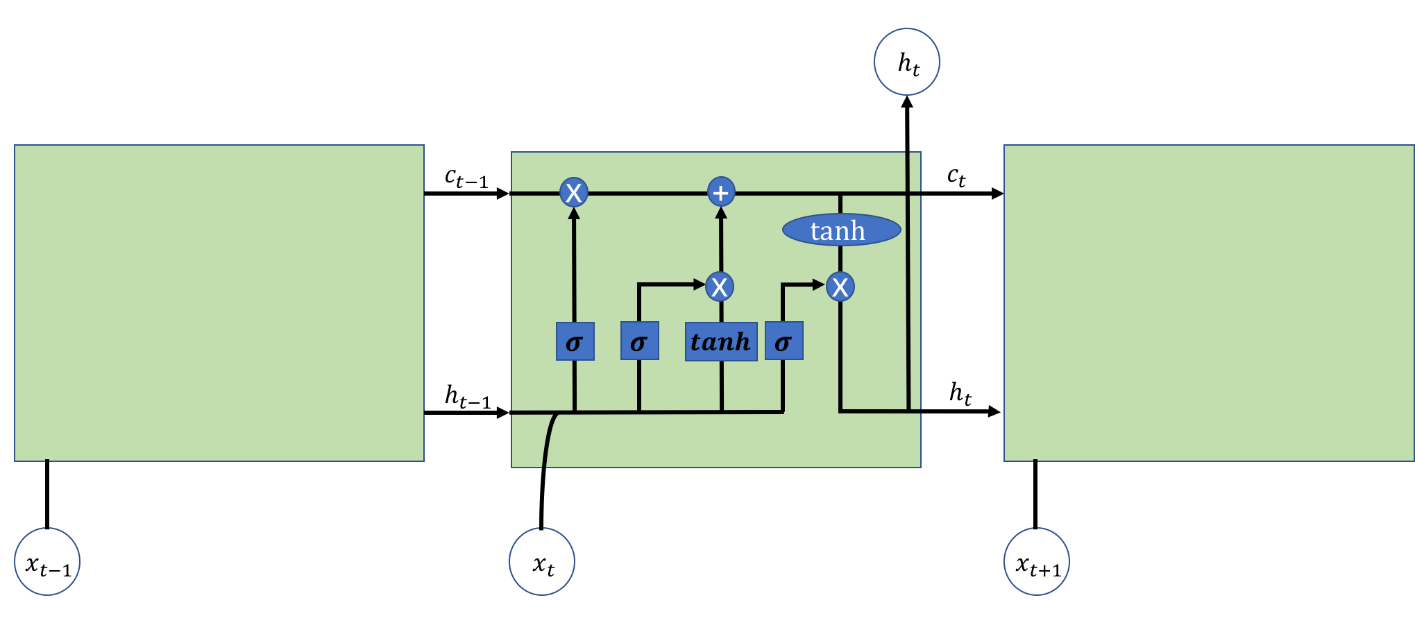


图2：LSTM的核心单元示意图

由图2，可以用简洁的公式表达了LSTM正向传播的门控逻辑:

遗忘控制门：

输入控制门：

输出控制门：

细胞状态更新：

输出：

(3)到(7)中，是参数矩阵,是偏置，每个Cell的输出包括和，表示函数，表示内积

## 2.3 门控制单元（GRU）

由2.2节可以看出，基本的LSTM网络结构比较复杂，参数比较多。这也是LSTM在训练较大规模的数据时很慢的主要原因之一。Cho等[5]基于LSTM在提出了一种现在非常流行的变体Gated Recurrent Unit。他们利用该变体，在学习短语表示中取得了巨大成功。

GRU背后的原理和LSTM类似，即用门控制机制来控制输入、记忆等信息，并在当前时间步做预测。GRU有两个门，分别是“重置门(Reset Gate)”和“更新门(Update Gate)”。重置门决定如何把新的输入信息和之前的记忆结合，更新门决定保存之前记忆的量。若将重置门设置为1，更新门为0，那么该网络就是Simple RNN。虽然使用门控制学习长期依赖关系的思想和LSTM一致，但是也有一些区别。首先GRU控制门比LSTM少一个，并且GRU不会控制并保留内部记忆()，且没有LSTM的输出门。再者，LSTM中的输入与遗忘门对应于GRU的更新门，重置门直接作用于前面的隐藏状态。GRU的神经元模型如下：

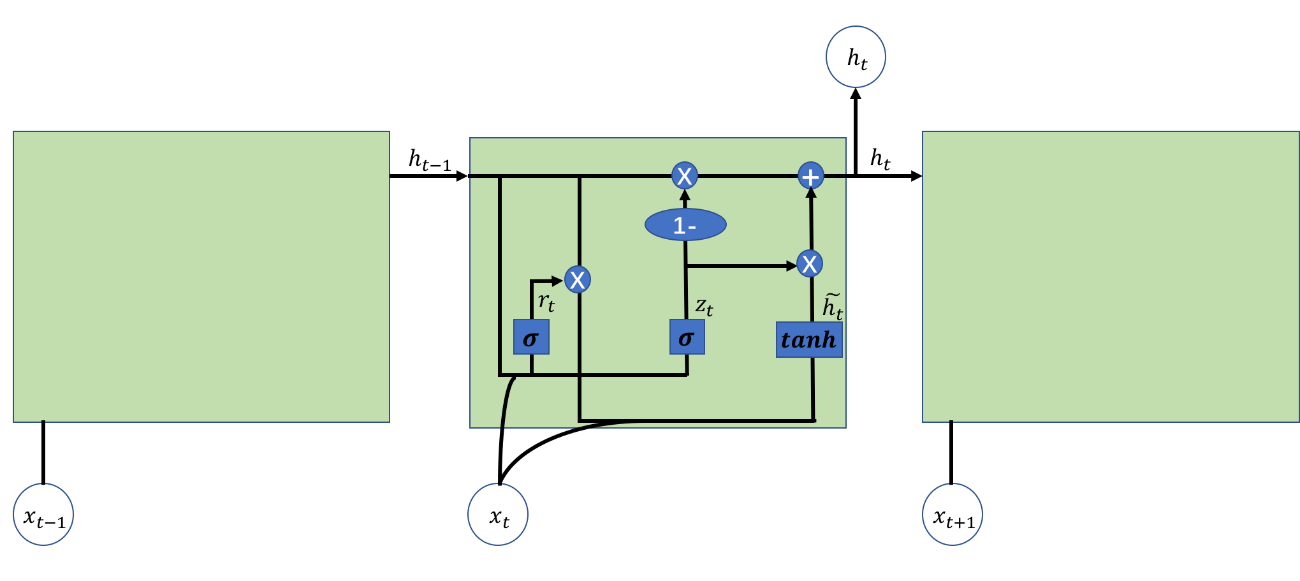


图3：GRU网络核心单元示意图

由图3，可以以下列出GRU正向传播公式，简洁高效的表达GRU核心逻辑：

更新门：

重置门：

更新状态：

输出：

从公式(8)到(11)中就能看出，GRU比LSTM简单了许多，同时Jozefowicz等人[6]通过大量的实验也证明，GRU有着极其优越的性能，它能在加快训练速度的同时，不失LSTM的准确度。

# 3 不同应用的循环神经网络（RNNs）模型

## 3.1 文本生成

文本生成是NLP问题里的一个很有趣应用。Becker等[7]利用了遗传算法来自动生成程序代码，Balog等[8]利用了神经网络来让程序学习写程序，Google的自动化机器学习（AutoML）项目利用神经网络构建神经网络模型，且其在图像识别中的效果比由人设计的网络更好。除了让神经网络写代码，还有许多有趣的事情。比如让神经网络写小说，让神经网络看图写诗，甚至是让神经网络写论文等。

文本生成实际上属于语言模型（Language Model），语言模型可以定义成一个概率分布模型，对于语言里每一个字符串给出一个概率:

(12)到(13)中，是中的第个字符。文本生成即给了初始的文本，通过上面的概率，预测。如下图所示：

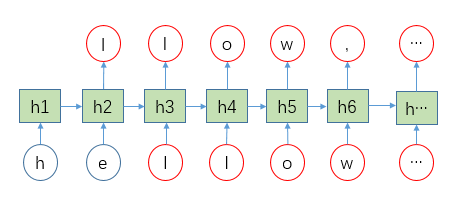


图4：文本生产过程图

图4是通过初始字符串“he”，预测出“hellow,…”的整个模型，预测时，前一个预测的结果会作为后一个预测的输入，直到生成的文本达到预定长度，或预测到句末符，预测结束。

## 3.2 seq2seq模型

seq2seq也是NLP的一个应用。近年来，人们尝试用循环神经网络应用到seq2seq模型中，并且取得了巨大的成功。Seq2seq模型有很多的应用，例如语言翻译。Sutskever等[9]基于LSTM的seq2seq翻译模型在实验中取得了巨大成果，达到了传统方法的最佳效果。Seq2seq模型也可以应用在自动问答系统，聊天机器人中。总的来说，seq2seq目前已经广泛地应用到人们的生活的方方面面之中。

基于LSTM的seq2seq模型包括两个核心部分，其一是编码层（Encoder），其二是译码层（Decoder）。这两层都由LSTM构成，每个Cell可以是LSTM，也可以是GRU等LSTM的变种。如下图：

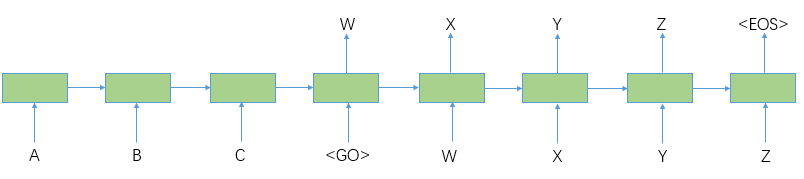


图5：seq2seq模型时间过程图

图5中，左边编码端（不含输出的就是编码端）的主要作用是把输入的句子（如ABC）映射到一个维度固定的向量（Vector）中。这样，输入的全部信息都会保存在这个向量（LSTM的隐藏层状态）中。译码端（右端含输出的部分）通过这个向量，获得输入句子的信息，然后再输出目标句子（<GO>WXYZ<EOS>，其中<GO>和<EOS>分别代表句子的开始和结尾）。

当然，需要有几个特殊符号来表示句子开始、结束等标志，如上图中的<GO>和<EOS>等。

## 3.3 图像识别模型

虽然近些年来大部分有关计算机视觉的文章，都是基于CNNs做图像处理的，但是在一些比较简单的图像识别任务中，RNNs也能胜任，甚至比CNNs表现的更好。RNNs图像处理模型如下所示：

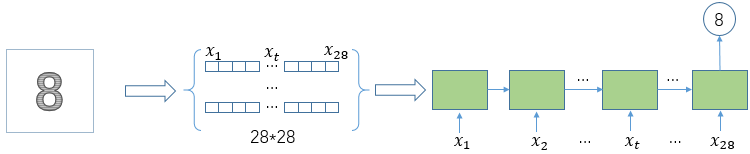


图6：图像识别模型处理过程图

一张图是由的像素矩阵表示。假设这张图是灰度图，那么就是。抛开图的意义，我们可以认为这张图的输入就是个长度为的序列，那么RNNs就能像处理序列任务一样来处理图像识别任务。如上图，手写数字集数据的图片大小为，那么输入RNN序列的时间步则为28，这个模型也是典型的N对1模型。

# 4 研究方案设计

LSTMs整体结构中，最重要的部分也就是隐藏层层数，每层神经元数，以及具体的神经元的种类（例如LSTM、GRU等）。这三个部分确定之后，整个模型也能大致确定，大多数很好的模型都是确定了这三部分，然后对其他参数进行微调的结果。

因此，本文中在每个模型里，设定好超参数，通过枚举的方式，着重改变这三个部分，探究这三个部分对不同模型的影响，并且从中找到一些规律和共同点，为优化LSTM网络结构提供一个深入的见解。

## 4.1 文本生成

该模型旨在进行训练之后，能读入一小段字符，并生成其余的字符。为了达到目标，本文在基于Tensorflow的谷歌开源框架，模块化的构建该模型。首先对输入数据进行加载和编码转化，得到文章中所有的字符的集合、字符到数字的映射、数字到字符的映射以及对原文进行编码转换后的列表。

接下来划分数据集。本文中，把数据集进行mini-batch划分，批(batch)的大小和序列长度（或者叫时间步）在第5节中有详细定义。假设每个批中有个字符，那么数据集就能产生个批，data\_size是数据集的字符总数。

然后构建网络模块，该模块包括输入层、LSTM层、输出层、损失函数(loss)以及优化器(optimizer)等部分。因为之前数据预处理得到每个批的大小为，因此输入层的大小也是，且定义dropout概率，详细值在第5节中阐明。LSTM层定义了Tensorflow中最基本的LSTM核心，且构建多层的时候，每层的核数保持一样（方便整体做变化），每层的LSTM核心也用初始化函数进行初始化。输出层采用函数，它和LSTM层进行全连接，并计算logits和经过规范之后的概率分布。训练误差在对目标进行one-hot编码后用交叉熵来计算。定义Adam[10]算法优化器，通过梯度消减（gradient clippling）防止梯度爆炸，具体的值见第5节。构建完网络的模块后，在通过一个类将这些部分组合起来，并把输入进行one-hot编码，最后使用Tensorflow的动态RNN训练接口对网络进行训练。因为该模型数据量较大，因此每层的LSTM核数变化区间在[128,768]之间，且变化大致均匀，层数由1层增长到13层。

## 4.2 seq2seq模型

为了简单探究层数以及每层神经元数对模型的影响，该模型的输入来自上个模型输入中提取的单词，然后需要输出的是单词对字符排序的结果。例如输入单词“selection”，则需要输出“ceeilnost”。

该模型同样基于Tensorflow的框架搭建，首先对数据进行预处理，和上个模型类似，不过需要在文本字符集中添加补全符<PAD>、句首标志符<GO>、句末标志符<EOS>、未出现字符<UNK>四个特殊符号。

接下来构建网络模型，主要包括编码（Encoder）层和译码（Decoder）层，首先对输入数据利用Tensorflow的接口进行单词嵌入（Word Embedding）操作，然后堆叠编码层，并把处理过得数据传入，通过动态RNN训练接口训练，得到其最后的隐藏层输出以及状态。搭建译码层与编码层类似，本文中设定两个层的层数以及神经元数都一样。但是在训练译码层时，本文将目标序列作为输入传递给译码层的每个时间步，而不是使用前一时间步的输出作为下一时间步的输入，这样能使模型更准确，因此，在训练阶段以及测试阶段需要搭建不同的译码层，并共享其参数。构建完成编码和译码层后，将它们连接起来构建成seq2seq的模型，并用Tensorflow的sequence\_loss接口定义损失函数，再自定义Adam[10]算法优化器，在每次求出梯度后对梯度消减。该模型处理的任务比较简单，因此将每层LSTM核数定义在[32,192]范围内，层数由1增加到12。

## 4.3 图像识别模型

模型加载手写数字识别（MNIST）标准的数据集的训练、验证、测试数据，每张图片的规格都固定。把每张图每列作为输入，因为每列都是28维，不需要在做one-hot变换，把时序长度也定义成28，刚好能读入一张图片。

构建网络模型的方法与文本生成模型类似，同样基于Tensorflow构建输入层、LSTM层、输出层、损失函数以及优化器。输入层大小为，LSTM层初始化方式与之前的模型相同，输出层与LSTM层全连接，输出采用函数，并计算logits和经过处理的概率分布作为输出。训练误差用实际概率和预测值的交叉熵来计算。定义Adam[10]优化器时，通过梯度消减（gradient clippling）防止梯度爆炸，具体的值见第5节。构建完网络的模块后，在通过一个类将这些部分组合起来，最后使用Tensorflow的动态RNN训练接口对网络进行训练。该模型识别的类别比较少（只有10类），训练数据较少，任务也很简单，训练速度也很快。因此每层LSTM核数定义在区间[32,600]之间，层数由1增加到20，并且在小于5时步长为1，之后步长改为2。这样设计是因为在层数较少时，随着层数增加，可能结果变化会比较敏感。

# 5 实验

## 5.1 实验数据

文本生成的实验数据取自英文长篇小说。把该长篇小说中，出现的单词提取出来去掉重复，得到的是seq2seq模型的实验数据。通过把单词复制相应的倍数，可以得到seq2seq模型更加复杂的数据。图像识别的数据来自MNIST手写数字识别数据集。MNIST有完整的正确的数据集，分为训练集，验证集和测试集。其中训练集包含60000个正确的样本和标签，测试集包含10000个正确的样本和标签。为了保证数据的正确性，来自英文长篇小说的数据经过数据处理程序分析输出，且经过了校准。

## 5.2 实验方法

### 5.2.1 文本生成

首先对训练数据进行预处理，生成词汇列表，并生成词汇到整数和整数到词汇的字典，然后定义超参数。本模型中，批的大小定义为100，时间步定义为100，学习率定义为0.001，神经元dropout的概率定义为0.5，隐藏层的范围规定为[1,13]，每层的神经元个数范围规定为[128,768]。并且把第一个批的数据作为验证集数据，其余定义为输入数据。为了降低网络过拟合的成都，模型中每个隐藏层都添加dropout方法，把输入的数据循环向右移动一位，得到我们需要学习的内容。为了解决梯度爆炸的问题，该模型在优化时，对梯度进行消减，把梯度控制在5以下。

为了和GRU对比，本文中只把标准的LSTM核换成GRU，在隐藏层层数作为和每层的神经元个数做变量的情况下，固定其余所有的参数进行实验。

### 5.2.2 seq2seq模型

该模型较其他两个模型会复杂一些。首先对输入数据进行预处理，定义4个辅助符号，它们分别是补全符<PAD>、译码器端的句子结束标识符<EOS>、词汇表未出现过的字符<UNK>、译码器端句子起始标志符<GO>，将这四个符号和文中出现的符号一起放入词汇表集和，并构建相应的映射表，接着定义超参数。其中，训练轮数定义为50，批的大小定义为128，字符嵌入操作的维度大小定义为15，学习率定义为0.001，dropout的概率为0.2。编码和译码模块的层数和每层核数都定义为变量，其中层数区间是[1,12]，核数区间是[32,192]。把输入的句子开头都添加<GO>符号，并且在句末都添加<EOS>符号，将每个批的序列都用<PAD>补充到等长，然后去掉最后一列。同时，把第一个批的数据作为验证集，其余作为训练集。输入的数据全部做字符嵌入操作。每个编码层LSTM核心的参数初始化值均匀分布在[-0.1,0.1]之间。

译码层的LSTM核心的参数初始化值均匀分布在[-0.1,0.1]之间。为了加快训练速度，在进行训练的时候，本文中也用实际的输出作为译码层的输入，而不是将编码层的输出输入到译码层。这样做能让RNN在正确的指导下训练。因为该模型的任务比较简单，因此把输入单词长度经过复制，增加了三倍，作为对比实验数据。

### 5.2.3 图像识别模型

模型的输入取自规范化的MNIST数据集，省却了其他两个模型的数据预处理操作。每个手写数字的图的大小都是。训练的批的大小为128，测试的批的大小为1000，训练轮数为100，学习率为0.001。隐藏层层数变化范围是[1, 20]，每层的神经元个数变化范围是[32, 600]，隐藏层每层LSTM核心的参数初始化值均匀分布在 [-0.1, 0.1]之间。训练时候的dropout概率设为0.5，验证和测试的时候不进行dropout（其概率为0）。

输出模块接受隐藏模块最后一层的状态，并增加一个softmax层预测每个数字的概率，损失函数是softmax层输出和实际标签的交叉熵，准确度是预测正确的数据占总预测数的比例。同样，为了解决梯度爆炸的问题，在Adam[10]优化的时候定义梯度裁剪，保证其最大梯度为4。

## 5.3 实验环境

本次实验基于tensorflow1.3-gpu框架，操作系统版本是ubuntu16.04，python版本是3.6。文中所有的模型都使用含一个Titan X (Pascal)的主机。因为是采用枚举层数和每层神经元数的方法进行实验，因此耗时较长。所有模型跑完结果（包括修改的模型）都记录在文件中，实验共计耗时约3周。

## 5.4 实验结果及分析

### 5.4.1 文本生成模型

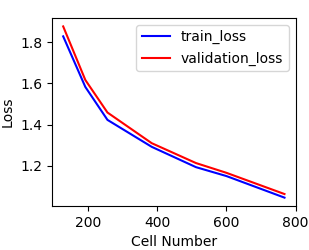
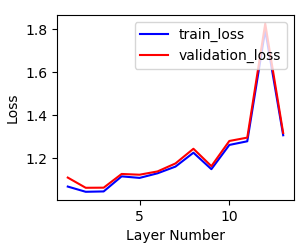
 

图7：左图表示把层数固定为3，随着每层LSTM核心数的增加，训练误差和验证误差的变化，右图表示的是把每层LSTM核心数固定为768时，随着层数增加，训练误差和验证误差的变化。

由图7中的左图可以看出，随着每层核心(LSTM)的增加，训练误差和验证误差逐渐降低，可以预见，当核心数再增多时，误差将趋于平缓。因为数据集是一篇英文小说，作者的写作风格相对来说比较固定，因此训练误差和验证误差大致在同一水平线上。从图7的右图看出，当固定了每层核心数，随着层数增加，误差出现了预期之外的变化。预期中，它的曲线应该和图一的左图类似，但是实际上，误差先降低，接着波动性上升。

为了避免实验的偶然性，图8画出了其他核心数随着层数增加误差变化图。可以看出，随着层数增加，误差改变的趋势大致相同——先降低，再增加。

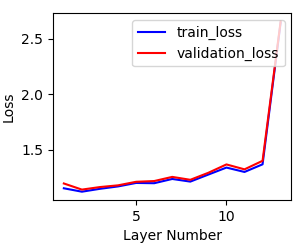
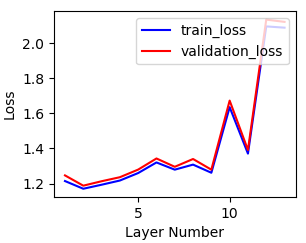
 

图8：左图表示把每层LSTM核心数固定为600时，随着层数增加，训练误差和验证误差的变化，右图表示把每层LSTM核心数固定为512，随着层数增加训练误差和验证误差的变化。

按照一般的规律，层数增加不应该让结果变得更坏。为了解释这个原因，本文借鉴了残差网络(ResNet)[12]的说法——让深层网络变得难以训练的真正原因不是梯度消失，而是网络退化。网络退化是指随着网络层数的增加，模型的误差或者准确度都不如更低层数的网络。Orhan等[11]对网络退化做了详细的描述，并提出导致网络退化的原因是层数增加而引起了参数矩阵奇异值的降低。

图9不同的曲线代表着不同的核数，由颜色分明的曲线我们能看出，通过把每层的LSTM核数增加，似乎能减缓网络退化的程度（咖啡色、粉色以及紫色的曲线上升趋势有明显的延迟）。

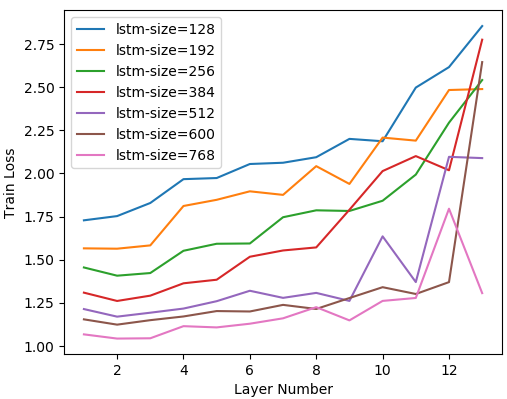


图9：图中不同的核数由不同颜色的曲线标注，随着层数增加，训练误差的变化。

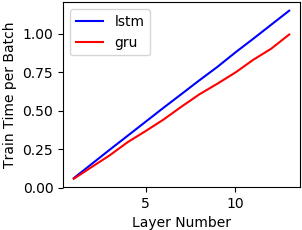
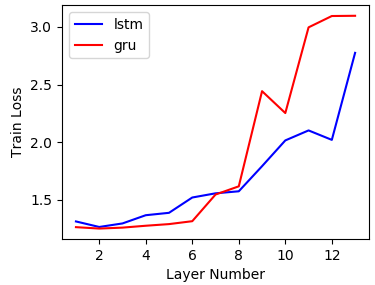


图10：左图表示把每层的核数都固定为384，随着层数增加，LSTM核心与GRU核心的训练误差的变化，右图表示把每层核数固定为768，随着层数增加，不同的RNN核心训练每个批的数据的平均用时。

图10中，由左图看出GRU的核心在网络退化问题严重影响到网络之前，表现的比LSTM的核心更优越，但是随着层数增加，GRU核心的网络退化的更加严重。同时，右图可以看出，训练GRU核心的网络确实会比训练LSTM核心的网络更快一些。

### 5.4.2 序列转化模型

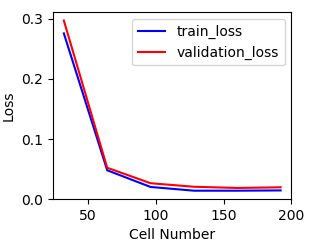
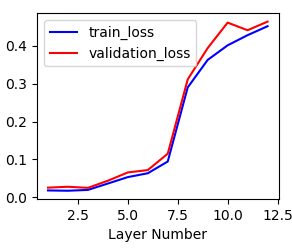
 

图11：左图是把隐藏层固定为3（编码和译码层都是3）的情况下，随着LSTM核心数增长，训练误差和验证误差的变化。右图是固定LSTM核心数为96，误差随着层数增加的改变。左图和右图都基于简单的数据（序列长度为原始长度）。

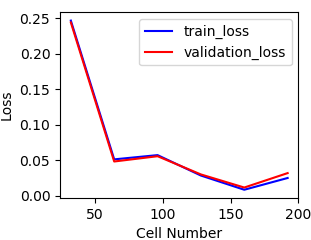
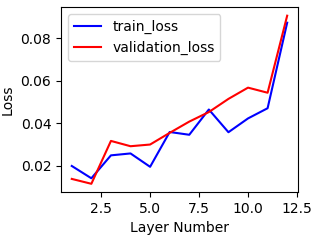
 

图12：左图是把隐藏层固定为3（编码和译码层都是3）的情况下，随着LSTM核心数增长，训练误差和验证误差的变化。右图是固定LSTM核心数为192，误差随着层数增加的改变。左图和右图都基于更困难的数据（序列长度为原始长度三倍）。

由图11、图12可以清晰地看到，在数据集相差不大的情况下，随着模型内部结构的改变，误差的改变趋势都相差不大，例如，在核心数增加到64之前误差都比较大，模型不能拟合数据，在增加到足够的核心数后，误差趋于平缓。而随着层数增加，无论数据是否改变，网络都产生了退化的问题，误差波动的上升，这说明网络的退化与数据集的关系不大。

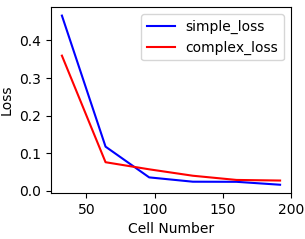


图13：上图是固定隐藏层为3(编码和译码层都为3)时，随着每层LSTM核心数增加，误差的变化。其中蓝线是基于简单数据，红线是基于复杂数据。

分析图13，可以知道数据的复杂度对网络模型的表现还是有着影响的。在LSTM核心数足够的情况下，模型在复杂数据上得到的结果会差一点点。

在图14中，我们更能看出网络退化的速度和每层的核心数有着极大地关系，通过增加每层网络的LSTM核心数，能明显减缓网络退化的速度。

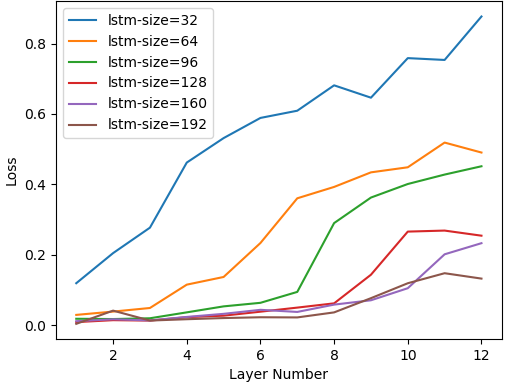


图14：上图表示随着层数增加，不同的LSTM核心数的网络的训练误差的变化。

### 5.4.3 图像识别模型

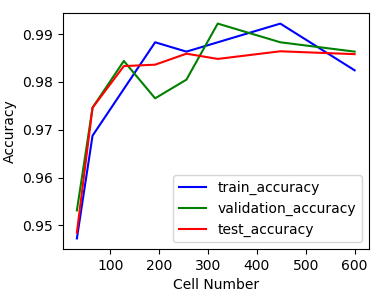
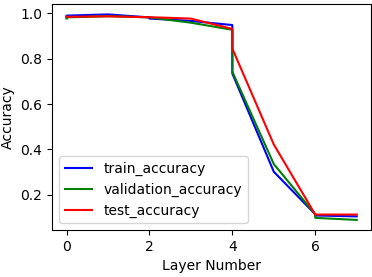
 

图15：左图是固定隐藏层为3，训练、验证、测试准确率随着LSTM核心数增加的变化情况，右图是在LSTM核心数为448时，随着层数增加，三种准确率的变化情况。

图15中左图的验证集在随LSTM核心数增加的时候，波动比较大。这是因为手写数字识别（MINIST）任务比较简单，其预测准确率很接近完美预测产生的正常波动。由此可见，每层LSTM的核数并不是越多越好。右图也表现出了网络退化的问题，这意味着网络退化问题，和模型需要处理的任务的关系也不大，无论哪种模型，单纯的增加网络深度，都会出现这样的问题。这也是导致深度网络很难训练的原因。

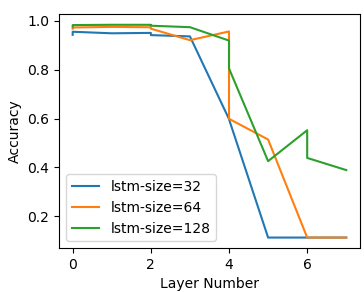


图16：上图是不同LSTM的核数，其测试准确率随着隐藏层层数增加的变化情况。

图16中虽然不太明显，但是也有着随LSTM核数增加，网络退化减缓的趋势，因此，网络退化能用增加每层核心数的方法来缓解。

# 6 总结与展望

## 6.1 总结

21世纪是深度学习蓬勃发展的时代。随着GPU性能的提升，人们构建的深度神经网络越来越复杂，模块越来越多，参数越来越庞大。深度网络的复杂度逐渐偏离了人们的掌控。无论是调整模型的结构，还是调整模型的超参数，都不一定能取得特别好的结果。从近些年发表的文献上看，还没有一个能使用于各种任务的很优秀的RNNs模型。并且各个文献，针对不同的问题，参数也几乎不一样。在文献中，也很少有提到为什么要那个数量的核心数，或者是层数，几乎所有的模型和参数都是按照经验主义设置的。目前为止，尚未有文献提到其文献中提出的模型就是最合适的模型。因此，通过某种算法，自动化的优化神经网络模型显得尤为重要。

本文中，通过三个较为简单的模型，探讨了RNN模型的结构对结果的影响。通过分析三个RNN模型的多个实验结果，本文发现，RNN模型的层数、每层的核心数、每层核心的种类、数据等对结果都有着或多或少的影响。并且，每种问题，其适应的模型的结构都不大相同，但也有共性。

从三个模型的实验结果分析可以知道，随着RNN模型每层核心数的增加，误差先降低，随之趋于平缓。当然这个结果也是公共所接受的，但是因为单纯的增加每层核心数会出现过拟合的问题，同时低层的网络结构提取到的特征不比高层的更好，因此实际上，各种应用中的神经网络都是增加合适的层数而不是单纯的增加每层神经元的个数。

从每个模型随着隐藏层层数增加的误差变化图中能看出，误差随着隐藏层层数增加先降低，接着升高。在此处，数据来自训练误差，因此可以排除掉因为过拟合导致误差增加这种原因。同时，因为使用的是LSTM核心，并且层数也限制到了20层内，因此也可以排除掉误差爆炸和消失这两种原因。Orhan等[11]提出了导致深度网络难以训练的真正原因是网络退化的观点，在单纯的增加网络深度的时候，误差会变大，甚至明显上升。因此，RNN深度网络模型并不能通过简单的堆叠隐藏层实现，需要解决因为深度带来的梯度消失、爆炸以及网络退化等问题。

在模型一的实验中能看出，GRU的有着不输LSTM的性能，甚至在有时候会得到更好的结果，但是其在网络退化问题上似乎表现的差了一点。本文因为时间条件和硬件条件，并没有通过大量实验对GRU和LSTM做详细的数据对比。Jozefowicz等[6]通过大量的实验，证明了GRU在各个问题上表现都让人满意。

通过分析三个实验不同的核心数在随隐藏层增加的表现上来看，本文发现，网络隐藏层的核心数越多，退化也就越慢。这个结论与Orhan[11]对退化原因的合理解释相符合。退化的原因通常是因为单纯堆叠神经网络的层数，导致其参数矩阵奇异值降低，而当增加每层核心数也等同于增加了参数矩阵的奇异值，因此对该降低作用有所减缓。He, Kaiming等人[12]在卷积神经网络VGG-19的基础上，提出了一种短连接(shortcut connection)模块，解决了网络退化的问题，并将网络的层数堆叠到了百层以上。

## 6.2 展望

随着深度学习进一步发展，现在的网络模型越来越复杂，层数越来越深，参数越来越多。网络的复杂性爆炸式增长，比如残差网络模型[12]可以堆叠层数到几百，甚至上千层的有效模型。但是实际上，并非所有的问题都需要极其复杂的网络模型，在几百上千层的网络模型中，各个层对整个问题的影响是极其小的。有人也试过从巨大的模型中拿掉一些层，发现其结果并没有什么变化。因此，为了让网络的模型更加简洁、有效，为了能在短时间训练出一个满意的模型，我们希望能找到一种方法，在保证模型精确度的前提下，尽可能的优化模型的结构，使得其结构简单，并且训练的计算复杂度也不高。然而，这不是一件简单的事情。

近年来，有大量的专家学者投入到优化算法以及迁移学习的研究工作中，其中也有优化神经网络结构的方法。Liu[13]提出了基于多目标任务优化Deep Neural Network（DNN）的隐藏层连接结构的方法，Zhang[14]提出基于多目标任务优化DBN的思想，Balog等[8]使用深度学习方法写程序，Becker等[7]基于基因算法设计的系统只需要极少的人为指导就能生成整个的软件程序，A. Gupta等[15]、L. Feng等[16]、F. Liang等[17]也提出了大量的优秀的优化算法。

本文基于三个简单的模型，将RNN模型的核心层数、每层神经元数、LSTM核心类别、数据集作为变量，横向对比了影响各个模型的综合因素。其余的参数，例如学习率等都是固定不变的。同时，上述模型在增加层数时，也遇到了网络退化的问题，导致结果变得较差。并且，对于一个问题，在没有经验的情况下，冒然调整其网络结构或者参数往往会得到较差的结果。因此，借鉴上述优化算法以及迁移学习方法，并且设计一种合理可行的方法，对于特定问题去自动的优化LSTM网络的结构或参数就显得极其重要。

# 致 谢

大学四年的时间如白驹过隙般划过，不知不觉我在重庆大学求学即将满四年，我也由最初懵懂稚嫩的少年通过自己的努力蜕变成了现在成熟有担当的模样。日历翻过了立夏，又即将翻向六月，都说六月是凤凰花开的路口，是告别的季节。离别不说悲伤，几有所得，应谢于八方。

首先我要感谢的是黄桷树下的母校，感谢母校为我提供了良好的学习及生活环境，让我能在这优美的校园中每天保持一颗向上的愉悦的心，专注学业，刻苦拼搏。在水一方的不只是伊人，更有陪伴我度过了无数个刷题啃专业书夜晚的图书馆；忙碌了一天，回到寝室洗个热水澡洗掉所有疲倦，舒适的床让我第二天能有更好的状态去学习。

我要感谢这四年来认真给我授课的老师们以及身边一起学习吃饭的朋友们。师者，所以传道、授业、解惑也。感谢老师们将己所学，不遗余力教予我；将我之惑，悉数解答。桃礼不言，下自成蹊。感谢朋友让我生活上不再孤单，学习上有同路者，挫败时有倾心鼓舞、放松时有适当提醒。

最后，我要感谢我的毕业设计指导老师冯亮研究员。冯老师不仅在学业上言传身教，而且以其高尚的品格给以我情操上的熏陶。本次毕业设计能够完成，也是得益于冯老师提供的GPU服务器，以及良好的实验环境。而论文从选题到具体的安排，冯老师经过了仔细推敲，论文完成后，冯老师在审阅时一丝不苟，句句斟酌，让我在本次毕业设计中得到了极大锻炼，同时为我在以后的研究学习中作了重要铺垫。

# 参考文献

1. HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.
2. SOCHER R, LIN C C Y, NG A Y, et al. Parsing natural scenes and natural language with recursive neural networks[C]. Proceedings of the 28th international conference on machine learning (ICML-11), Washington, USA, 2011.
3. MIKOLOV T, KARAFIAT M, BURGET L, et al. Recurrent neural network based language model[C]. Eleventh Annual Conference of the International Speech Communication Association, San Jose, USA, 2010.
4. HOCHREITER SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
5. CHO K, MERRIENBOER B V, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv, 2014, 1406.1078.
6. JOZEFOWICZ R, ZAREMBA W, SUTSKEVER I. An empirical exploration of recurrent network architectures[C]. International Conference on Machine Learning, Lille, France, 2015.
7. BECKER K, GOTTSCHLICH J. AI Programmer: Autonomously Creating Software Programs Using Genetic Algorithms[J]. arXiv, 2017, 1709.05703.
8. BALOG M, GAUNT A L, BROCKSCHMIDT M, et al. Deepcoder: Learning to write programs[J]. arXiv, 2016, 1611.01989.
9. SUTSKEVER I, VINYALS O, and LE Q V. Sequence to sequence learning with neural networks[C]. Advances in neural information processing systems, Montréal, Canada, 2014.
10. KINGMA D P, and BA J L. Adam: A method for stochastic optimization[C]. arXiv preprint arXiv:1412.6980, San Diego, USA, 2015.
11. ORHAN A E, and PITKOW X. Skip connections eliminate singularities[C]. arXiv preprint arXiv:1701.09175, Vancouver, Canada, 2018.
12. HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 2016.
13. LIU J, GONG M, WANG X, et al. Structure learning for deep neural networks based on multiobjective optimization[J]. IEEE transactions on neural networks and learning systems, 2017: 2450-2463
14. ZHANG C, LIM P, QIN A K, et al. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics[J]. IEEE transactions on neural networks and learning systems 2017, 28(10): 2306-2318.
15. GUPTA A, ONG Y S, and FENG L, Multifactorial Evolution: Towards Evolutionary Multitasking[J], IEEE Transactions on Evolutionary Computation, Accepted 2015: 343 - 357
16. FENG L, ONG Y S, LIM M H, et al. Memetic Search with Inter-Domain Learning: A Realization between CVRP and CARP[J], IEEE Transactions on Evolutionary Computation, Accepted 2014, 19(5): 644 - 658.
17. FENG L, ONG Y S, TSANG I W H, et al, An Evolutionary Search Paradigm that Learns with Past Experiences[J], IEEE Congress on Evolutionary Computation, June 2012: 1-8
18. FENG L, ONG Y S, NGUYEN Q H, et al, Towards Probabilistic Memetic Algorithm: An Initial Study on Capacitated Arc Routing Problem[J], IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation 2010: 1-7.

# 附 录

1. 表1:

|  |  |  |
| --- | --- | --- |
| Year | Author | Contribution |
| 1990 | Elman | 普及了最简单的RNNs |
| 1993 | Doya | 在梯度下降过程用正确标记指导(techer forcing)，预训练RNNs网络 |
| 1994 | Bengio | 发现RNNs无法处理长句依赖的问题 |
| 1997 | Hochreiter | 提出长短期记忆网络（LSTM）用于解决梯度消失问题 |
| 1997 | Schuster | 提出了双向循环网络 |
| 1998 | LeCun | 海森矩阵在解决梯度消失问题的应用 |
| 2000 | Gers | 扩展了LSTM的遗忘门 |
| 2001 | Goodman | 利用单词分类快速训练极大熵 |
| 2005 | Morin | Hierarchical softmax优化输出层 |
| 2005 | Graves | 双向LSTM |
| 2007 | Jaeger | Leaky-integrator神经元的应用 |
| 2007 | Graves | 多维度的RNNs |
| 2009 | Graves | LSTM在手写识别的应用 |
| 2010 | Mikolov | 基于语言模型的RNN |
| 2010 | Neir | ReLU激活函数在解决梯度消失问题的应用 |
| 2011 | Martens | Hessian-free优化RNN学习过程 |
| 2011 | Mikolov | BPTT算法在RNN统计语言模型的应用 |
| 2011 | Duchi | 每个权重的自适应学习率 |
| 2012 | Gutmann | 噪声对比估计（NCE） |
| 2012 | Mnih | NCE在训练概率语言模型神经元的应用 |
| 2012 | Pascanu | 利用梯度消减避免梯度爆炸问题 |
| 2013 | Mikolov | 用负采样代替Hierarchical softmax |
| 2013 | Sutskever | 随机梯度下降法（SGD） |
| 2013 | Graves | 深度LSTM网络 |
| 2014 | Cho | 门循环单元结构的应用 |
| 2015 | Zaremba | 利用Dropout解决过拟合问题 |
| 2015 | Mikolov | 约束结构的递归神经网络（SCRN）在加强长记忆和解决梯度消失问题的应用 |
| 2015 | Visin | ReNet的使用（把CNN中卷积+池化层替换为循环神经网络） |
| 2015 | Gregor | DRAW（Deep recurrent attentive writer）在图片生成的应用 |
| 2015 | Kalchbrenner | Grid LSTM |
| 2015 | Srivastava | Highway networks减轻梯度下降所消耗的时间 |
| 2017 | Jing | 正交的循环单元门 |