这本书主要包括自然语言处理，计算语言学，信息检索，口语理解，重点在于面向新技术，新应用，人类语言学的两个或多个子领域。

书的前半段涉及监督机器学习和前馈神经网络的基础，通过语言数据进行机器学习的基本知识，以及使用矢量的方式而不是符号对单词进行表示，同时涵盖了计算图抽象，这种方式允许我们简单地定义和训练任意神经网络，并且式现代神经网络软件库设计的基础。

书的第二部分包括一些特定的神经网络架构，包括一维卷积神经网络，循环卷积神经网络，条件生成模型以及注意力模型。这些架构和技术是机器翻译，句法分析和其他许多应用程序的推动力量。最后，我们还讨论了树形网络，结构化预测和多任务学习前景。

# 引语

自然语言处理指的是人类语言自动计算处理。包括采用人类产生的文本作为输入的算法，和产生类似自然的文本作为输出的算法。这其中的困难在于二义性，变化性以及没有明确的定义。

自然语言本质上是符号，在开始进行自然语言处理的尝试中也是符号型的，基于逻辑，规则和知识。然而自然语言具有高度的模糊性和可变性，这些特性需要统计的方法来解决。实际上当前主流的语言处理是基于统计机器学习。过去的十年自然语言理解基于线性模型的方法，主要包括感知机，线性SVM，和逻辑回归，通过非常高维但非常稀疏的特征向量进行学习。

在2014年左右，在密集输入的非线性神经网络取得了更大的成功。一些神经网络是线性模型的简单推广，并且可以近似看作是线性分类器的直接替代。一些新的创造性的改变，例如使用RNN减轻对马尔科夫性质的依赖在序列模型上表现良好。允许适应任意长的序列和产生高效的特征提取，这些导致了语言模型，自动翻译，和其他应用上的突破。

作者在这本书里传输了一些基本概念，术语以及工具。

这本书是从Goldberg，2016上发展而来的。目标那些希望采用现有技术解决实际问题的读者。如果要进行进一步的。讨论，Bengio的书被高度推荐。随后推荐了机器学习上的一些著作以及其他的相关著作。

这里主要涉及一些已经很成熟的技术和一些有希望的技术。

# 1章

## 1.1 NLP的挑战

NLP是设计（输入或输出非结构化的，自然语言数据的模型和算法）。自然语言是模糊的，随语境的改变而改变，也是变化和发展的。人类是语言很好的使用者，但在理解和解释语言规则方面很无力。

使用计算机理解和创造语言是十分困难的。事实上，已知处理语言数据的最好方法是使用监督学习的方法，这些方法从一些预先标注的输入和输出对中获得一些模式和规则。机器学习的算法适合在一些很难定义规则但可以通过标注给定输入得到输出很容易的领域。

除了语义模糊和规则难以定义外，自然语言往往是离散的，组合的和稀疏的。

语言是符号性的和离散的，书写语言的基础元素是字符。字符形成单词，反过来表示目标，概念，事物，动作和概念。字符和单词均是离散的符号，单词例如汉堡知识的是某种确定的心理感知的表示，汉堡包和披萨之间从符号本身和单独的字母之间没有内在的联系。这里作者也对图像与语言之间进行了比较，例如颜色来源于机器视觉，它是连续的，可以从彩色图像映射到灰度图像通过数学操作，或是通过内在的属性可以简单的比较粉红色和红色，而在语言上并不容易做到这一点。

语言也是组合性的，字母形成了单词，单词形成了短语和句子，短语的意义可能比单独的单词的意义更大，并且遵循一些复杂的规则。为了理解文本，除了要在字母和单词上工作之外，还要看词语的序列例如句子或者完整的文件。

上述特性的组合造成了数据的缺少。不同方式组合起来单词形成的意义是实际上是无限的。有效的句子量极为庞大，不能期望对其进行枚举，就像本文中出现的许多四字的意义就是很新的。十年前的一些人，品牌，公司，网络用语，技术名词和未来十年你会碰到的都是新颖的。因此没有明确的形式从一句话推断出另一句话或是定义出他们的相似性。这并不依赖于他们我们未知的含义。这对我们要从一些例子中进行学习进而观察到从未出现在样本集中的的实例和在数据集中非常不同的实例很有挑战性。

## 神经网络和深度学习

深度学习是机器学习的一支，是神经网络重命名的，以可微分的参数化函数为特征。深度学习是来源于多层可微分函数的连接。

虽然所有的机器学习方法都是以基于过去的观察为特征来进行预测，但是深度学习不仅是要预测也是要正确的表述数据。给定要求的输入输出映射，深度学习可以通过将数据通过一个网络产生一系列转换，直到最终可以预测输出。这种转换通过给定输入输出的映射来实现，以便于每一个转换能够使数据与相应的标签相联系。

人们设计网络的架构，训练机制，提供合适的输入输出集合，编码输入数据以合适的方式，网络学习正确的表示，通过网络架构自动运行。

## 深度学习在NLP

神经网络在自然语言问题的处理上很有吸引力。神经网络对于语言的使用的一个中心部分使编码层，将离散的符号映射到连续的向量上在一个相对低维的空间。当对单词进行嵌入时将孤立的符号转化为数学可以操作的对象。尤其是，向量之间的距离可以等同于单词之间的距离，是将一些行为可以泛化到各个单词上。把单词映射为向量的表示是作为网络训练过程大的一部分。沿着网络的层次结构往上，网络学会了将词向量组合起来以有利于预测任务的方式。这种能力一定程度上缓解了离散性和数据稀疏性的问题。

有两种主要的神经网络架构可以以一些方式进行组合，前馈神经网络和循环神经网络。

前向网络，特别是多层感知机，允许以固定大小的输入输入，或是可变的长度输入但是不考虑元素的顺序。当给网络一组输入组件的时侯，它会学会将它们以有意义的形式组合起来，MLPs曾经被用作一种线性模型。非线性的网络架构带有可以轻易的结合预训练的词嵌入的形式，通常会导致优秀的分类准确率。

卷积神经网络是特殊设计的结构，它在提取数据局部特征上表现优秀，可以输入任意大小的输入，对于特定的单词顺序敏感，而不考虑这种模式出现在哪里，因此他们工作在表示固定长度的短语和成语的指示上表现良好。

循环网络是专门为顺序数据而设计的模型。它们是一些网络可以一些序列化的事物作为输入，产生一个固定大小的向量去总结这个序列.这种总结在不同任务中表现不同（例如回答关于句子情感的信息和关于语法的信息是不同的）。循环网络很少独立的进行任务，通常作为一个可训练组件将将数据输入到其他的网络结构中，与他们协同工作。举例可以将循环网络的输出传递给前馈神经网络中，以尝试预测一些值。循环网络通常被用作输入转换器被训练去产生有信息的表示然后传递给前馈网络在顶层对其进行操作。循环网络对于序列是十分有意义的，可能是神经网络对于语言处理给出的最令人激动的贡献。他们允许放弃过去几十年一直流行的马尔科夫假设，并且设计完全基于整个句子的模型，在需要的时候把单词顺序考虑进去，而且不必为数据的稀疏性而苦恼。他的可以进行语言建模的能力令人印象深刻，而且可以承担预测下一个词的任务，这时许多NLP应用的基石。递归网络将循环扩展为树。

自然语言处理的很多问题是结构化的，要求产生复杂的输出结构例如序列或是树。神经网络适合这种需要，既可以通过对已知的线性模型的结构化预测算法进行调整或是通过一些新奇的结构例如S2S(encoder-decoder)模型（我们在这本书中提到了条件生成模型）来达到这一目标。这些模型是现今的机器翻译方法的核心。

最后许多语言任务是彼此相关的，在某种意义上可以通过学习一种任务来进行另一种任务。当我们缺少有标记的训练数据的时候，我们可以简单地使用原始的数据作为支持。我们能够学习未标记的样本吗？神经网络可以通过学习相关领域的问题和半监督学习来解决这一问题。

### 成功的案例

全连接网络在某种意义上可以是大多数情况下线性学习器使用场景下代替线性学习器。这包括二分类和多分类问题和更加复杂的结构化预测问题。网络的非线性以及能够很容易地整合预先训练的单词嵌入的能力通常导致卓越的分类准确率。一系列的工作成功地获得了改进的语法解析结果，通过将线性模型代替为全连接的前馈前向神经网络。将前馈网络作为分类器替换的简单应用（通常是使用预先训练的单词向量）为许多语言任务提供了好处，包括语言建模的非常基本的任务，CCG超级标签，对话状态跟踪，以及统计机器翻译的预序。Iyyer等人2015年演示了多层结构前馈网络可以在情绪分类和问答方面提供有竞争力的结果回答问题。周等人2015年和Andor等人将他们整合到一个定向搜索中的结构预测系统，在句法分析、序列标注上达到卓越的精度和等等。具有卷积和池化层的网络对于分类任务非常有用，我们希望找到针对分类对象有关的强有力的局部线索，但这些线索可以出现在输入的不同位置。例如，在一个文档分类任务中，一个关键短语（或ngram）可以帮助确定文档的主题。卷积和池化架构在许多任务上显示了有希望的结果，包括文档分类、短文本分类，情绪分类，实体之间的相对类型分类，事件检测，解释识别，语义角色标签，问题回答，预测电影票房收入（基于评论家的评论），对文本趣味度的建模，以及对字符序列和词性标注之间的关系进行建模。

在自然语言中，我们经常使用任意大小的结构化数据，比如序列和树。我们希望能够在这种结构中捕捉规律，或者模拟这种结构之间相似性。循环网络是被设计模拟序列的，递归网络是被设计模拟树的.循环网络在语言建模上表现强大，以及序列标注，机器翻译、解析、处理和许多其他任务，包括噪声的文本规范化、对话状态跟踪，响应生成，并建模字符序列之间的关系和词性标记。

递归模型被证明可以产生最先进的或接近最先进的结果对于选区的依赖性解析重新排序，话语解析，语义关系分类，基于解析树的政治意识形态检测，情绪分类，目标独立的情感分类，问答。

## 书籍组织结构

e书由四个部分组成。第一部分介绍了我们将在书中使用的基本学习机制：监督学习、mlp、基于梯度的训练，以及用于实现和训练神经网络的计算图抽象。第二部分将第一部分中介绍的机器与语言数据连接起来。它介绍了在处理语言数据时可用的主要信息来源，并解释了如何将它们与神经网络机制集成在一起。它还讨论了文字嵌入算法和分布假设，以及语言建模的前馈方法。第三部分处理专门的体系结构及其对语言数据的应用：用于处理ng的一维卷积网络，以及用于建模序列和堆栈的RNNs。RNNs是将神经网络应用于语言数据的主要创新，而第三部分的大部分内容都是针对它们的，包括它们所促进的强大的条件生成框架和基于注意力的模型。第四部分是各种高级主题的集合：用于建模树的递归网络、结构化预测模型和多任务学习。

第一部分，涵盖了神经网络的基础，包括四章。第2章介绍了监督机器学习的基本概念，参数化函数，线性和对数线性模型，正则化和损失函数，训练为最优化，和基于梯度的培训方法。它从底层开始，为以下内容提供所需的材料章。熟悉基础学习理论和基于梯度的学习的读者可以考虑跳过这一章。第三章阐明了线性模型的主要局限性，激发了对非线性模型的需要，为多层神经网络奠定了基础和动力。第四章介绍了前馈神经网络和mlp。它讨论了多层网络，它们的理论力量，以及普通的子组件，比如非线性和损失函数。第五章处理神经网络训练。介绍了计算图允许对任意网络进行自动梯度计算的抽象（反向传播算法），并为有效的网络提供了几个重的技巧和技巧培训。

第二部分介绍语言数据，由七章组成。第6章介绍了常见语言处理问题的类型，并讨论了使用语言数据时可用的信息来源（功能）。第7章提供了具体的案例研究，展示了前一章描述的特征如何用于各种自然语言任务。熟悉语言处理的读者可以跳过这两章。第8章连系第6章和第7章的材料与神经网络，并讨论了各种编码方式基于语言的特征作为神经网络的输入。第9章介绍语言建模任务和前馈神经语言模型体系结构。这也为后面章节中讨论预训练的词嵌入。第10章讨论分布式和词义表征的分布方法。它引入了单词上下文矩阵进行分布式语义表示，以及神经语言建模通过词嵌入算法，比如GLOVE和Word2Vec，并讨论它们和分布式方法之间的联系。第11章介绍了如何使用外部的词嵌入神经网络的背景。最后，第12章介绍了一个案例研究

第三部分介绍了专门的卷积和重复架构，由五章组成。第13章介绍了卷积神经网络，它专门学习信息性的ngram模式。还讨论了另一种hash-kernel技术。这部分的其余部分，第14章17节，是关于rnn的。第14章描述了用于建模序列和堆栈的RNN抽象。第15章描述了RNNs的具体实例化，包括简单的RNN（也称为艾尔曼RNNs）和封闭式架构，如长短期内存（LSTM）和门控的重复单元（GRU）。第16章提供了使用RNN抽象建模的示例，展示了它们在具体应用程序中的使用。最后，第17章介绍了条件生成框架，它是最先进的机器翻译背后的主要建模技术，以及无监督的句子建模和许多其他的创新应用。第四部分是高级和非核心主题的混合，由三章组成。第18章介绍了树木结构的递归网络。虽然这一模式非常吸引人，但它仍处于研究阶段，而且还没有展现出令人信服的成功故事。尽管如此，对于那些致力于将建模技术推广到最先进水平的研究人员来说，这是一个重要的模型家族。那些对成熟和健壮的技术感兴趣的读者可以安全地跳过这一章。第19章涉及结构化预测。这是一个相当技术性的章节。那些对结构化预测特别感兴趣的读者，或者已经熟悉线性模型或语言处理的结构化预测技术的读者，很可能会欣赏这种材料。其他人可能会相当安全地跳过它。最后，第20章介绍了多任务和半监督学习。神经网络为多任务和半监督学习提供了充足的机会。是重要的技术，仍处于研究阶段。然而，现有的技术相对容易实现，并且确实提供了真正的收益。章在技术上并不具有挑战性，并被推荐给所有的读者。

## 什么没有介绍

没有关于语音方面的介绍，对于对话系统，文件总结或是知识问答这类open的问题没有涉及。这里描述的技术可以在这些领域使用，但没有具体讨论。语义分析也超出了范围。关于语言和其他模态（视觉）。数据的关系只是简单的提及，最后是以英语为中心的。

要求基础：合适的评价机制和数据标注。这些并没有被讨论。合适的评价机制要求对给定任务可以给出评估标准，和其他人的工作做对比，表现为误差的分析以及统计意义上的评估。

数据标注是NLP重要组成部分，要对数据的来源有了解，考虑到数据形成过程中的影响。数据标注是很大的话题，包括对标注任务选取合适的形式，决定标注的指导方针。决定标注数据的源，它覆盖的范围，类属性，好的训练测试集分割。和使用注释器，合并决策，验证注释器和注释的质量，以及各种各样的类似的主题。

## 术语的提示

“特征”是指一个具体的、语言的输入，例如一个单词，一个后缀，或者一个部分。语义标签。例如，在一阶的语义标记中，特征可能是“当前的单词，前一个词，下一个词，前一段话”。“输入向量”是用来指对于输入到神经网络分类器的实际输入。相对于其他的文献，这里讲输入向量拿了出来，而不是与特征混在一起。

## 数学符号

用大写字符表示矩阵，小写表示向量。一系列相关的矩阵和向量用上标索引。加粗表示代指，不加粗是具体化。向量被认为是行向量，矩阵乘向量是xW+b。Wi，j i是列，j是行