第1章
Neural Network Methods for Natural Language Processing

引　言

1.1　自然语言处理的挑战

自然语言处理(Natural Language Processing，NLP)是一个设计输入和输出为非结构化自然语言数据的方法和算法的研究领域。人类语言有很强的歧义性(如句子“I ate pizza with friends”(我和朋友一起吃披萨)和“I ate pizza with olives”(我吃了有橄榄的披萨))和多样性(如“I ate pizza with friends”也可以说成“Friends and I shared some pizza”)。语言也一直在进化中。人善于产生和理解语言，并具有表达、感知、理解复杂且微妙信息的能力。与此同时，虽然人类是语言的伟大使用者，但是我们并不善于形式化地理解和描述支配语言的规则。

使用计算机理解和产生语言因此极具挑战性。事实上，最为人所知的处理语言数据的方法是使用有监督机器学习(supervised machine learning)算法，其试图从事先标注好的输入/输出集合中推导出使用的模式和规则。例如，一个将文本分为4类的任务，类别为：体育、政治、八卦和经济。显然，文本中的单词提供了非常强的线索，但是到底哪些单词提供了什么线索呢？为该任务书写规则极具挑战性。然而，读者可以轻松地将一篇文档分到一个主题中，然后，基于每类几百篇人为分类的样例，可以让有监督机器学习产生用词的模式，从而帮助文本分类。机器学习方法擅长那些很难获得规则集，但是相对容易获得给定输入及相应输出样本的领域。

除了使用不明确规则集处理歧义和多样输入的挑战之外，自然语言展现了另外一些特性，其使得用包括机器学习在内的计算方法更具挑战性，即离散性(discrete)、组合性(compositional)和稀疏性(sparse)。

语言是符号化和离散的。书面语义的基本单位是字符，字符构成了单词，单词再表示对象、概念、事件、动作和思想。字符和单词都是离散符号：如“hamburger”(汉堡包)或“pizza”(披萨)会唤起我们头脑中的某种表示，但是它们也是不同的符号，其含义是不相关的，待我们的大脑去理解。从符号自身看，“hamburger”和“pizza”之间没有内在的关系，从构成它们的字母看也是一样。与机器视觉中普遍使用的如颜色的概念或声学信号

相对比，这些概念都是连续的，如可以使用简单的数学运算从一幅彩色图像变为灰度图像，或者从色调、光强等内在性质比较两幅图像。对于单词，这些都不容易做到，如果不使用一个大的查找表或者词典，没有什么简单的运算可以从单词“red”(红)变为单词“pink”(粉红)。

语言还具有组合性，即字母形成单词，单词形成短语和句子。短语的含义可以比包含的单词更大，并遵循复杂的规则集。为了理解一个文本，我们需要超越字母和单词，看到更长的单词序列，如句子甚至整篇文本。

以上性质的组合导致了**数据稀疏性**(data sparseness)。单词(离散符号)组合并形成意义的方式实际上是无限的。可能合法的句子数是巨大的，我们从没指望能全部枚举出来。随便翻开一本书，其中绝大部分句子是你之前从没看过和听过的。甚至，很有可能很多四个单词构成的序列对你都是新鲜的。如果你看一下过去 10 年的报纸或者想象一下未来 10 年的报纸，许多单词，特别是人名、品牌和公司以及俚语和术语都将是新的。我们也不清楚如何从一个句子生成另一个句子或者定义句子之间的相似性，这不依赖于它们的意思——对我们是不可观测的。当我们要从实例中学习时也是挑战重重，即使有非常大的实例集合，我们仍然很容易观测到实例集合中从没出现过的事件，其与曾出现过的所有实例都非常不同。

1.2 神经网络和深度学习

深度学习是机器学习的一个分支，是神经网络(neural network)的重命名。神经网络是一系列学习技术，历史上曾受模拟脑计算工作的启发，可被看作学习参数可微的数学函数[1]。深度学习的名字源于许多层被连在一起的可微函数。

虽然全部机器学习技术都可以被认为是基于过去的观测学习如何做出预测，但是深度学习方法不仅学习预测，而且学习正确地表示数据，以使其更有助于预测。给出一个巨大的输入-输出映射集合，深度学习方法将数据“喂”给一个网络，其产生输入的后继转换，直到用最终的转换来预测输出。网络产生的转换都学习自给定的输入-输出映射，以便每个转换都使得更易于将数据和期望的标签之间建立联系。

人类设计者负责设计网络结构和训练方式，提供给网络合适的输入-输出实例集合，将输入数据恰当地编码，大量学习正确表示的工作则由网络自动执行，同时受到网络结构的支持。

1.3 自然语言处理中的深度学习

神经网络提供了强大的学习机制，对自然语言处理问题极具吸引力。将神经网络用于语言的一个主要组件是使用嵌入层(embedding layer)，即将离散的符号映射为相对低维的连续向量。当嵌入单词的时候，从不同的独立符号转换为可以运算的数学对象。特别地，向量之间的距离可以等价于单词之间的距离，这使得更容易从一个单词泛化到另一个单词。学习单词的向量表示成为训练过程的一部分。再往上层，网络学习单词向量的组合方式以更有利于预测。该能力减轻了离散和数据稀疏问题。

有两种主要的神经网络结构，即前馈网络(feed-forward network)和循环/递归网络(recurrent/recursive network)，它们可以以各种方式组合。

前馈网络，也叫多层感知器(Multi-Layer Perceptron，MLP)，其输入大小固定，对于变化的输入长度，我们可以忽略元素的顺序。当将输入集合喂给网络时，网络学习用有意义的方式组合它们。之前线性模型所能应用的地方，多层感知器都能使用。网络的非线性以及易于整合预训练词嵌入的能力经常导致更高的分类精度。

卷积(convolutional)前馈网络是一类特殊的结构，其善于抽取数据中有意义的局部模式：将任意长度的输入“喂”给网络，网络能抽取有意义的局部模式，这些模式对单词顺序敏感，而忽略它们在输入中出现的位置。这些工作适合于识别长句子或者文本中有指示性的短语和惯用语。

循环神经网络(RNN)是适于序列数据的特殊模型，网络接收输入序列作为输入，产生固定大小的向量作为序列的摘要。对于不同的任务，“一个序列的摘要”意味着不同的东西(也就是说，用于回答一个句子情感所需的信息与回答其语法的信息并不相同)。循环网络很少被当作独立组件应用，其能力在于可被当作可训练的组件“喂”给其他网络组件，然后串联地训练它们。例如，循环网络的输出可以“喂”给前馈网络，用于预测一些值。循环网络被用作一个输入转换器，其被训练用于产生富含信息的表示，前馈网络将在其上进行运算。对于序列，循环网络是非常引人注目的模型，可能也是神经网络用于自然语言最令人激动的成果。它们允许：打破自然语言处理中存在几十年的马尔可夫假设，设计能依赖整个句子的模型，并在需要的情况下考虑词的顺序，同时不太受由于数据稀疏造成的统计估计问题之苦。该能力使语言模型(language-modeling)产生了令人印象深刻的收益，其中语言模型指的是预测序列中下一个单词的概率(等价于预测一个序列的概率)，是许多自然语言处理应用的核心。递归网络将循环网络从序列扩展到树。

自然语言的许多问题是结构化的(structured)，需要产生复杂的输出结构，如序列和树。神经网络模型能适应该需求，一方面可以改进已知的面向线性模型的结构化预测算法，另一方面可以使用新的结构，如序列到序列(编码器-解码器)模型，本书中我们指的是条件生成模型。此类模型是目前公认最好的机器翻译模型的核心。

最后，许多自然语言预测任务互相关联，在某种意义上知道一种任务是如何执行的将对另一些任务有所帮助。另外，我们可能没有足够的有监督(带标签)训练数据，而只有足够的原始文本(无标签数据)。我们能从相关的任务或者未标注数据中学习吗？对于多任务学习(Multi-Task Learning，MTL，即从相关问题中学习)和半监督(semi-supervised)学习(从额外的、未标注的数据中学习)，神经网络方法提供了令人激动的机会。

成功案例

大部分情况下，全连接前馈神经网络(MLP)能被用来替代线性学习器。这包括二分类或多分类问题，以及更复杂的结构化预测问题。网络的非线性以及易于整合预训练词嵌入的能力经常带来更高的分类精度。一系列工作[2]通过简单地将句法分析器中的线性模型替换为全连接前馈神经网络便获得了更好的句法分析结果。直接将前馈网络用作分类器(通常同时使用预训练词向量)为许多语义任务带来了好处，包括：非常基本的语言模型任务[3]，CCG标注(supertagging)[4]，对话状态跟踪[5]，统计机器翻译中的预排序[6]。Iyyer等人[2015]证明多层前馈网络能对情感分类和事实型问答带来富有竞争力的结果。Zhou等人[2015]和Andor等人[2016]将它们与基于柱搜索(beam-search)的结构化预测系统相整合，在句法分析、序列标注以及其他任务中获得了很高的准确率。

具有卷积和池化层的网络对于分类任务非常有用，我们期望从中发现很强的关于类别的局部线索，这些线索能出现在输入的不同位置。例如，在文本分类任务中，单一的关键短语(或者连续的n个词)能帮助确定文本的主题[Johnson and Zhang，2015]。我们期望学习某些有利于指明主题的单词序列，不需要关注它们出现在文档中的位置。卷积和池化层允许模型学习到这些局部指示，忽略它们的位置。卷积和池化结构在许多任务中展现了鼓舞人心的结果，包括文本分类[7]、短文本分类[8]、情感分类[9]、实体之间关系类型分类[10]、事件检测[11]、复述识别[12]、语义角色标注[13]、问答系统[14]、基于影评的电影票房预测[15]、文本趣味性建模[16]以及字符序列和词性标记之间关系的建模[17]。

在自然语言中，我们经常与任意长度的结构化数据打交道，例如序列和树。我们期望能够获取这些结构的泛化性，或者建模它们之间的相似性。循环和递归结构对序列和树结构很有效，能保留许多结构化信息。循环网络[Elman，1990]被设计用于对序列进行建

模，而递归网络[Goller and Küchler，1996]是对循环网络的泛化，能处理树。循环模型已经在很多任务上展示了非常强的效果，包括语言模型[18]、序列标注[19]、机器翻译[20]、句法分析[21]、噪声文本规范化[22]、对话状态跟踪[23]、反馈生成[24]以及字符序列和词性标记之间关系的建模[25]。

对于短语结构[26]和依存[27]句法分析的重排序、语篇关系分析[28]、语义关系分类[29]、基于句法分析树的政治意识形态检测[30]、情感分类[31]、目标依赖的情感分类[32]以及问答系统[33]，递归模型显示出能获得目前最好或近似最好的结果。

1.4 本书的覆盖面和组织结构

本书由四部分构成。第一部分介绍本书中将使用的基本学习机制，包括有监督学习、多层感知器、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介绍的机制与语言进行关联，介绍处理语言时所能用到的主要信息源，并解释如何将它们与神经网络机制进行整合。同时讨论词嵌入算法和分布式假设，以及将前馈方法用于语言模型。第三部分处理特殊的结构以及它们在语言数据中的应用，包括用于处理 ngram 的一维卷积网络、用于建模序列和栈的循环神经网络(RNN)。第三部分描述最多的 RNN 是针对语言数据的神经网络应用的主要创新，包括强大的条件生成框架以及基于注意力的模型。第四部分是各种新进展的集合，包括用于建模树的递归网络、结构化预测模型以及多任务学习。

第一部分涵盖了神经网络的基础，包括四章。第 2 章介绍有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、作为优化问题的训练以及基于梯度的训练方法。它从头开始，提供了后续章节所必需的材料。已经熟悉基本的学习理论和基于梯度的学习方法的读者可以跳过该章。第 3 章指出线性模型的主要缺陷、非线性模型的动机以及多层神经网络的基础和动机。第 4 章介绍前馈神经网络和多层感知器，讨论多层网络的定义、它们理论上的能力以及常用的组件，如非线性函数和损失函数。第 5 章涉及神经网络的训练，介绍能对任意网络进行自动梯度计算的计算图抽象(反向传播算法)以及提出几个用于有效训练网络的重要技巧。

第二部分介绍语言模型，包括七章。第 6 章提出一个通用语言处理问题的原型并讨论在使用语言数据时可用的信息(特征)资源。第 7 章提供了具体的案例，展示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 章将第 6、7 两章的内容与神经网络进行结合，讨论各种对基于语言的特征作为神经网络输入进行编码

的方式。第9章介绍语言模型任务，以及前馈神经语言模型结构。这也为后续章节中讨论预训练词嵌入铺就了一条道路。第10章讨论用于词义表示的分布式(distributed)和分布(distributional)方法，介绍用于分布语义的单词-上下文矩阵方法，以及受神经语言模型所启发的词嵌入算法，如GloVe和Word2Vec，并讨论了分布式和分布方法之间的联系。第11章在神经网络上下文之外讨论词嵌入。最后，第12章给出了一个任务相关的前馈网络案例，其专门用于自然语言推理(Natural Language Inference)任务。

第三部分介绍特殊的卷积和循环结构，包括五章。第13章是卷积网络，其专用于学习富含信息的n元语法模式。其替代品哈希核(hash-kernel)技术也将被讨论。第14～17章主要介绍循环神经网络。第14章描述用于序列和栈建模的循环神经网络抽象。第15章描述具体的循环神经网络实例，包括简单循环神经网络(也被称作Elman循环神经网络)以及带门的结构，如长短期记忆(Long Short-Term Memory，LSTM)和门限循环单元(Gated Recurrent Unit，GRU)。第16章给出使用循环神经网络抽象进行建模的一些实例，展示它们的具体应用。最后，第17章介绍条件生成框架(其为目前最好的机器翻译系统)，以及无监督句子建模和很多其他创新应用背后的建模技术。

第四部分是最新的非核心主题的混合，包括三章。第18章介绍用于建模树的树结构递归网络。虽然非常吸引人，但是这类模型仍处于研究阶段，还没有展示令人信服的成功案例。尽管如此，对于想进一步提高目前最好性能(state-of-the-art)的研究人员来说，这仍然是一类很重要的模型。对成熟和鲁棒技术更感兴趣的读者可以跳过该章。第19章处理结构化预测。这是很有技术含量的一章。对结构化预测有特别兴趣或者已经熟悉基于线性模型的结构化预测技术的读者会喜欢这部分材料。其他人可以跳过该章。最后，第20章介绍多任务和半监督学习。对于多任务和半监督学习，神经网络带来了丰富的机会。这些是很重要的技术，它们仍处于研究阶段。然而，已有的技术相对容易被应用并且确实提供了真正的收益。本章没有什么技术挑战，推荐所有读者阅读。

依赖性 对于大部分内容，一个章节依赖于其前面的章节。第二部分最前面的两章是个例外，它们不依赖前面任何章节的内容，可以以任意顺序阅读。一些章节可跳过，而不会影响对其他概念和材料的理解。如10.4节和第11章可以跳过，其介绍词嵌入算法的细节和在神经网络之前应用词嵌入。第12章描述了一个用于斯坦福自然语言推理(Stanford Natural Language Inference，SNLI)数据集的特殊结构。第13章描述卷积网络。在递归网络的序列中，第15章描述特殊网络的细节，也能相对安全地被跳过。第四部分的各章都是相互独立的，既能跳过也能以任意顺序阅读。

1.5 本书未覆盖的内容

本书专注于将神经网络应用于语言处理任务。然而，一些基于神经网络的语言处理的子领域被本书排除在外了。特别地，我专注于处理书面语言，不包括语音数据和信号。在书面语中，我保留了相对底层的、定义明确的任务，没有包括如对话系统、文本摘要、问答系统等领域，我认为这些是更开放的问题。虽然本书描述的技术也能用于这些任务，但是我没有提供实例或显式地直接讨论这些任务。类似地，语义分析也超出了范围。多模态应用只是稍微提及，它们将语言数据与视觉、数据库等模态的数据进行连接。最后，讨论主要以英语为主，形态学更丰富以及更少计算资源的语言只会简短地讨论。

一些重要基础也没讨论。特别地，语言处理中的两个重要方面是*恰当的评价*和*数据标注*。这两个主题都超出了本书的范围，但是读者应该意识到它们的存在。

*恰当的评价*包括对于给定的任务选择正确的评价性能的指标、目前最好的方法、与其他工作公平的比较，进行错误分析以及评估统计显著性。

*数据标注*是自然语言处理系统的基础。没有数据，我们不能训练有监督模型。作为研究者，我们经常仅使用其他人产生的“标准”标注数据。知道数据源以及考虑标注过程仍然很重要。数据标注是一个非常巨大的主题，包括：恰当的标注任务定义；开发标注规范；决定标注数据来源，其覆盖性和类别分布，好的训练-测试划分；与标注者一起工作，合并决策，验证标注者和标注的质量以及各种类似的主题。

1.6 术语

“特征”(feature)一词用于表示一个具体的、语言上的输入，如单词、后缀或者词性。例如，在一阶词性标注器中，特征可能是“当前词、前一个词、下一个词、前一个词性”。术语“输入向量”(input vector)用于表示被“喂”给神经网络分类器的真正输入。类似地，“输入向量条目”(input vector entry)表示输入的具体值。这不同于大部分神经网络资料，其中“特征”一词过度承担了这两种用法，并且主要用于表示一个输入向量条目。

1.7 数学符号

我们使用粗体大写字母表示矩阵($\boldsymbol{X}$，$\boldsymbol{Y}$，$\boldsymbol{Z}$)，粗体小写字母表示向量($\boldsymbol{b}$)。当有一系

列相关的矩阵和向量时(如每个矩阵代表网络中不同的层)，使用上标($\boldsymbol{W}^1$，$\boldsymbol{W}^2$)。对于一些很少出现的情况，比如想表示矩阵和向量的幂，在我们想求幂的项两边加上一对括号：$(\boldsymbol{W})^2$，$(\boldsymbol{W}^3)^2$。我们使用[]作为向量和矩阵的索引运算符：$\boldsymbol{b}_{[i]}$是向量 $\boldsymbol{b}$ 的第 i 个元素，$\boldsymbol{W}_{[i,j]}$是矩阵 $\boldsymbol{W}$ 的第 i 列第 j 行。当没有歧义的时候，我们有时使用更标准的数学符号 b_i表示向量 $\boldsymbol{b}$ 的第 i 个元素，类似地 $w_{i,j}$表示矩阵 $\boldsymbol{W}$ 的元素。我们使用 $\cdot$ 表示点乘运算：$\boldsymbol{w}\cdot\boldsymbol{v}=\sum_i w_i v_i = \sum_i \boldsymbol{w}_{[i]}\boldsymbol{v}_{[i]}$。我们使用 $\boldsymbol{x}_{1:n}$表示向量 $\boldsymbol{x}_1$，…，$\boldsymbol{x}_n$的序列。类似地，$x_{1:n}$是元素 x_1，…，x_n的序列。我们使用 $\boldsymbol{x}_{n:1}$表示序列的逆序。$\boldsymbol{x}_{1:n}[i]=\boldsymbol{x}_i$，$\boldsymbol{x}_{n:1}[i]=\boldsymbol{x}_{n-i+1}$。我们使用[$\boldsymbol{v}_1$；$\boldsymbol{v}_2$]表示向量串联。

如无其他说明，向量被假设为行向量。选择使用行向量多少有点不标准，其被矩阵右乘($\boldsymbol{x}\boldsymbol{W}+\boldsymbol{b}$)，而许多神经网络材料使用列向量，它们是矩阵左乘($\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}$)。我们相信读者在阅读这些材料时能适应[34]。

注释

1. 本书中我们采用数学的观点，而非脑启发的观点。
2. [Chen and Manning，2014，Durrett and Klein，2015，Pei et al.，2015，Weiss et al.，2015]。
3. 见第 9 章以及 Bengio 等人[2003]、Vaswani 等人[2013]。
4. [Lewis and Steedman，2014]。
5. [Henderson et al.，2013]。
6. [de Gispert et al.，2015]。
7. [Johnson and Zhang，2015]。
8. [Wang et al.，2015a]。
9. [Kalchbrenner et al.，2014，Kim，2014]。
10. [dos Santos et al.，2015，Zeng et al.，2014]。
11. [Chen et al.，2015，Nguyen and Grishman，2015]。
12. [Yin and Schütze，2015]。
13. [Collobert et al.，2011]。
14. [Dong et al.，2015]。
15. [Bitvai and Cohn，2015]。
16. [Gao et al.，2014]。
17. [dos Santos and Zadrozny，2014]。
18. 一些值得注意的工作是 Adel 等人[2013]，Auli 和 Gao[2014]，Auli 等人[2013]，Duh 等人

[2013]，Jozefowicz 等人[2016]，Mikolov[2012]，Mikolov 等人[2010，2011]。

19. [Irsoy and Cardie，2014，Ling et al.，2015b，Xu et al.，2015]。
20. [Cho et al.，2014b，Sundermeyer et al.，2014，Sutskever et al.，2014，Tamura et al.，2014]。
21. [Dyer et al.，2015，Kiperwasser and Goldberg，2016b，Watanabe and Sumita，2015]。
22. [Chrupala，2014]。
23. [Mrkšić et al.，2015]。
24. [Kannan et al.，2016，Sordoni et al.，2015]。
25. [Ling et al.，2015b]。
26. [Socher et al.，2013a]。
27. [Le and Zuidema，2014，Zhu et al.，2015a]。
28. [Li et al.，2014]。
29. [Hashimoto et al.，2013，Liu et al.，2015]。
30. [Iyyer et al.，2014b]。
31. [Hermann and Blunsom，2013，Socher et al.，2013b]。
32. [Dong et al.，2014]。
33. [Iyyer et al.，2014a]。
34. 使用行向量表示具有以下优势：输入向量的方式与网络图所画的方式相匹配；网络的层次结构更显而易见，将输入置于最左侧而不是被嵌套起来；全连接层的维度是 $d_{\text{in}} \times d_{\text{out}}$，而不是 $d_{\text{out}} \times d_{\text{in}}$；更好地与神经网络代码实现相对应，如使用 numpy 等矩阵库实现这些代码。