6.深度学习模型

CONTENTS

一、深度学习

深度学习

深度学习是一种复杂的机器学习算法，在近年来得到了非常广泛的应用。深度学习模型的本质是神经网络。神经网络在1943年首次被McCulloch & Pitts提出，其旨在模仿神经元的工作方式来进行机器学习。

近年来科学技术的发展，深度学习显示出了其强劲的生命力，在很多领域取得了非凡的成果，效果远超先前的相关技术。在文本分析领域，深度学习也取得了很好的效果。

深度学习

FNN是一种较为基础的神经网络，而RNN和CNN都是FNN的进一步发展，同时RNN和CNN也是非常经典的深度学习模型。

- **CNN模型**主要是对空间维度上特征的提取，在图像识别等领域取得了非凡的成果。Kim在2014年使用CNN模型进行文本分类，在短文本领域分类效果较好，不过其在长距离建模方面能力受限，对于语序不够敏感。
- **RNN模型**主要是对时间维度上特征的提取，在时间序列和文本分析等领域有着非常重要的应用。

深度学习

RNN模型的核心思想在于不断保留和传递历史信息，并使得整个序列不断地向前发展，但是传统的RNN模型的缺点在于无法实现长期记忆性，因此Hochreiter & Schmidhuber (1997) 提出了**长短期记忆模型（LSTM，Long Short Term Memory）**。LSTM模型也属于RNN模型，其对传统的RNN模型进行了改进，增加了长期状态变量，能够兼顾近期的信息和长期的信息。

深度学习

2017年，Vaswani 等在论文Attention Is All You Need中提出了一种全新的网络结构——**Transformer**，它抛弃了传统的RNN和CNN模型，由一系列神经网络层堆叠组成。在每一个神经网络层中都包含一个自注意层（self-attention layers），自注意层使得当前的输入可以利用所有的历史信息，而不需要像RNN模型中那样通过从前至后依次连接，这一模型在机器翻译中取得了很好的应用成果。

2018年，**BERT模型**被提出，这是一个基于Transformer网络结构的模型，借用了其encoder的部分完成动态词嵌入，被认为是深度学习在文本分析领域的重大进展。BERT模型被视作是词嵌入的一种拓展。

深度学习

本章所要介绍的模型如下所示:

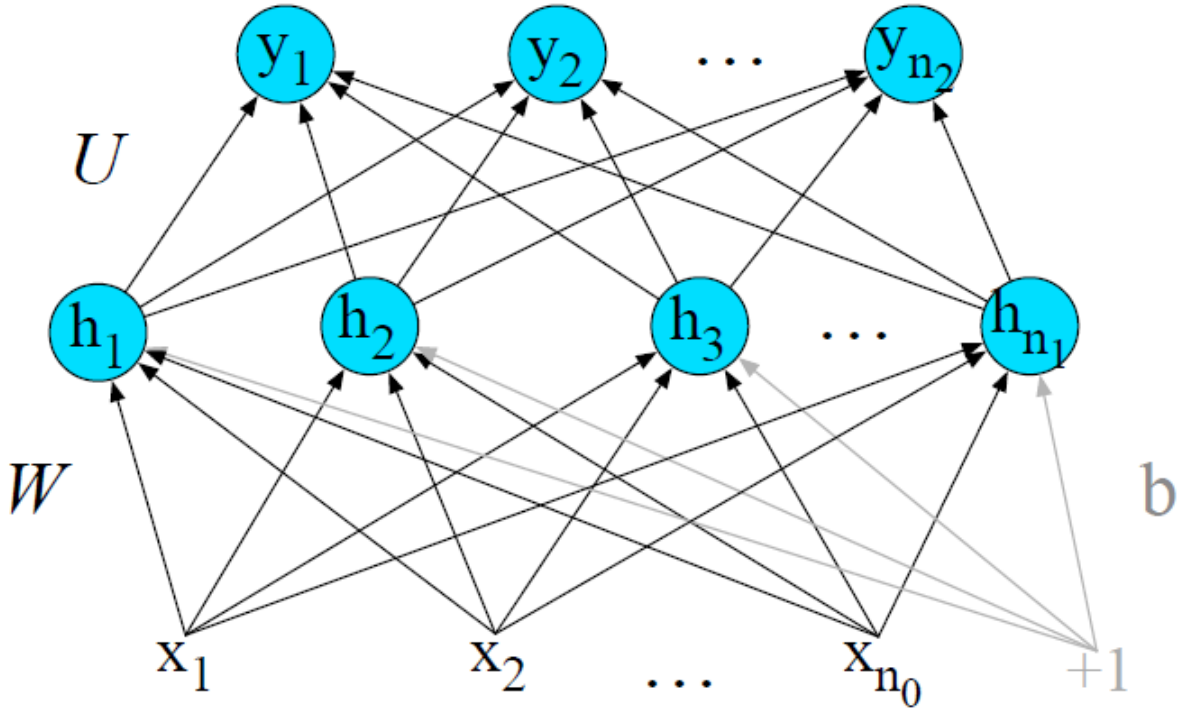
模型名称	中文名称	基本情况	主要应用领域
FNN模型	前馈神经网络	最基础的神经网络模型之一	
RNN模型	循环神经网络	多用于处理时间序列数据	时间序列预测、文本生成、文本分类等
LSTM模型	长短期记忆模型	RNN模型的拓展之一	文本生成、文本分类、机器翻译等
CNN模型	卷积神经网络	类型多变，以“卷积”、“池化”为核心	图像处理、文本分类等

二、前馈神经网络

前馈神经网络

前馈神经网络是一种最为简单的神经网络模型，简单的前馈神经网络有三个构件，即**输入层、隐藏层、输出层。**其中隐藏层可以由多层组成。

所谓前馈神经网络是指输入信息会沿着每层神经网络传递到下一层，而不会出现反向传递的情况



前馈神经网络

首先，在输入层输入各个词语的向量$x = (x_1, \ldots, x_{n_0})^{\mathrm{T}} \in \mathbb{R}^{n_0}$，其中$n_0$表示每次输入的向量维度，如果事先将词语转换为词向量，则n_0代表词向量的维度。输入向量会在隐藏层进行处理。具体来说，首先采用加权求和的形式得到$\mathrm{z} = \mathrm{W}x + b$，其中$\mathrm{W} \in \mathbb{R}^{\mathrm{n}_1 \times \mathrm{n}_0}$代表权重，$n_1$表示隐含层的向量维度；$b \in \mathbb{R}^{n_1}$表示偏差。处理后的结果z再经过**激活函数**$f(\cdot)$。常用的激活函数有sigmoid函数$\sigma(x) = (1 + e^{-x})^{-1}$，Tanh函数$\tanh(x) = (1 - e^{-2x})/(1 + e^{-2x})$，ReLu函数$\mathrm{relu}(x) = \max(x, 0)$等。在不同的问题中可以采用不同的激活函数。通过激活函数的非线性转换最终得到隐藏层的输出结果为：$h = f(z) = \mathrm{f}(Wx + b) \in \mathbb{R}^{\mathrm{n}_1}$

从隐含层到输出层需要再次经过权重矩阵$U \in \mathbb{R}^{n_2 \times n_1}$的转换，从而获得向量$z' = Uh \in \mathbb{R}^{n_2}$。

前馈神经网络

神经网络的模型通常用于分类问题，因此模型的因变量通常是一个分类变量，因此在输出层需要给出每一个样本取各个类别的概率，所以在输出层需要对向量z'进行**softmax变换**，即

$$\text{softmax}(z_i') = e^{z_i} / \sum_{j=1}^{n_2} e^{z_j}$$，其中i表示第i个类别。

通过softmax变换后即可得到输出向量$y = \text{softmax}(z') \in \mathbb{R}^{n_2}$。在自然语言处理中，输出结果y可能是文本所属的类别，也可能是与输入词语临近的词（在词嵌入模型中）。

前馈神经网络

在训练模型时，首先应该初始化参数（权重等），然后使用**交叉熵损失**来作为损失函数。每次训练时，对损失函数求偏导然后使用**梯度下降法**来进行优化。需要注意的是，当神经网络层数增加时，神经网络的参数也不断增加，由于参数维度太大，梯度下降法难以操作。

为了解决这一问题，Rumelhart等在1986年提出了**误差反向传播**算法。简单来说，信息随着神经网络的结构从前向后传播，得到某个估计值后，可以计算估计值和真实值之间的误差，然后进一步令该误差从后向前传播，也就是将该误差从后向前分解到每一层上，从数学公式来看，就是使用链式法则进行逐层求导，从而获得每层需要更新的梯度值。误差反向传播算法也是目前深度学习模型的常用估计方法。

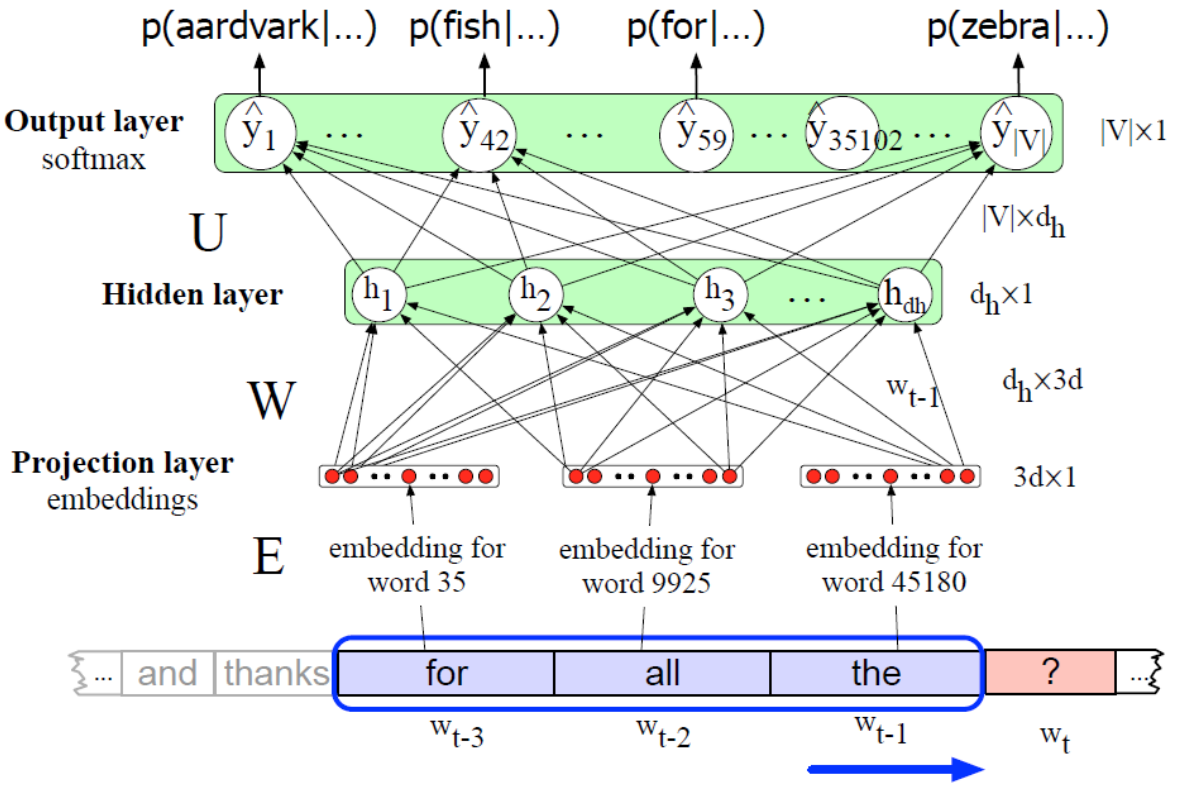
前馈神经网络

对于如何将前馈神经网络应用于自然语言模型，Bengio等在其2003年发表的论文A Neural Probabilistic Language Model上进行了讨论。假设$P(w_n|w_{1:n-1})$表示给定前n-1个词语以后，第n个词语出现的概率。假设w_n出现的条件概率只和它前面出现的最后N个词有关（$N < n-1$），而无需使用所有前n-1个词，因此有如下近似关系：

$$P(w_n|w_{1:n-1}) \approx P\left(w_n\middle|w_{(n-N+1):(n-1)}\right)$$

前馈神经网络

模型假定当前词w_t的预测与前$N=3$个词语（$w_{t-3}, w_{t-2}, w_{t-1}$）有关。在输入这三个词的词向量后，经过加权求和、激活函数的变换，获得隐含层的h向量，之后再经过从隐含层到输出层的权重矩阵U，再进行softmax变化得到最后的输出y。在该例子中，预测结果实际表示的是词库里所有词出现w_t之前三个位置上的概率。这一前馈神经网络模型与讲述的词向量模型有一些相似之处，事实上CBOW和Skip-gram模型正是基于这一模型的拓展。



三、循环神经网络

循环神经网络

循环神经网络是一种对于前馈神经网络（FNN）的改进。前馈神经网络中假定 $P(w_n|w_{1:n-1}) \approx P(w_n|w_{(n-N+1):(n-1)})$，即某一个词语的预测只与其前N-1个词语有关，这使得较远的文本无法对中心词语起到预测作用。但是实际中，一些较远的语句事实上也可能与当前语句有着较好的对应关系，能够起到较好的预测效果，因此有了循环神经网络模型。

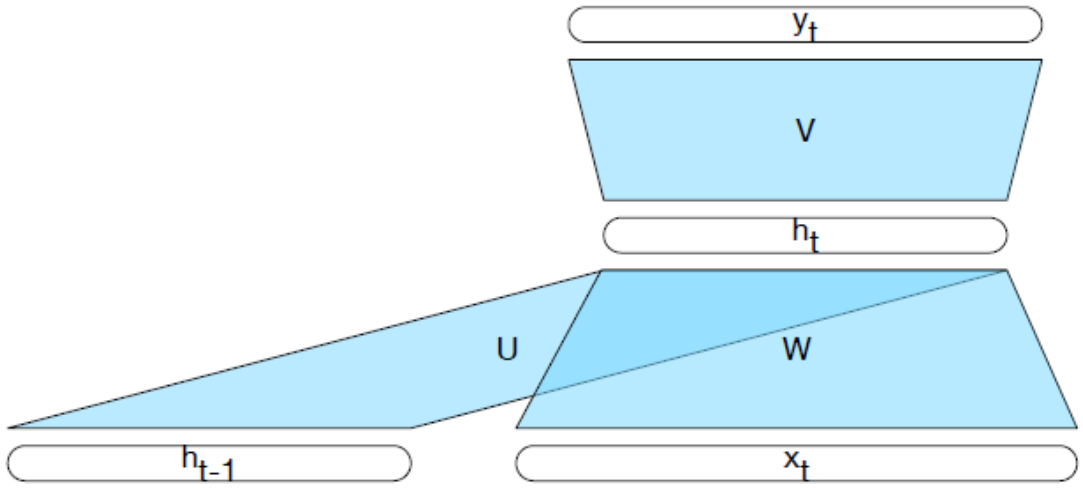
循环神经网络（RNN）的核心思想在于不断的保留和传递历史信息，并对历史信息进行压缩，与此同时，循环神经网络模型没有对历史的文本长度有所限制。由于可以不断的传递历史信息，循环神经网络模型可以形成“短期记忆”，因此相较于前馈神经网络而言，可以更好的实现对文本逻辑的判断。

循环神经网络

所有具有环形结构的神经网络模型均可以被称为循环神经网络。

对于输入x_t，将该输入乘以一个权重矩阵W；与此同时，之前传递来的历史信息h_{t-1}将乘以权重矩阵U；最后两部分信息通过激活函数$g(\cdot)$结合起来作为当前的信息，即：$h_t = g(Uh_{t-1} + Wx_t)$。

对当前信息再乘以权重矩阵V获得当前输出，即$y_t = f(Vh_t)$。按照这种方式传递下去，就做到了不断保留历史信息对当前结果提供参考。



循环神经网络的示意图

循环神经网络

RNN模型的优点在于模型形式比较简单，只有若干个权重矩阵需要训练，而且其不断的利用了历史的信息，能够比FNN模型获得更好的预测效果。

但是RNN模型也有着一些问题。首先，由于其需要不断的向下传递，所以必须要顺次训练，如果文本较长，则需要花费很长的时间；此外，在信息不断的传递过程中，需要估计的权重矩阵是固定的，因此最后的预测效果不一定很好。

循环神经网络的训练和前馈神经网络类似，也是使用反向传播的方法进行训练。

循环神经网络

循环神经网络在自然语言中有着广泛的应用，可以用于文本分类、文本生成等任务。

- RNN模型最常见的应用就是通过学习大量的文本数据，获取文本生成的规律，并通过该规律**自动生成文本**。例如：通过学习大量的古诗词来模拟作诗；通过学习小说《哈利波特》来模拟生成小说等。
- RNN也可以完成**文本分类**等工作。使用RNN进行文本分类时，首先将文本按顺序输入RNN模型中，最后一个输入完成后，其输出即代表了整个序列的一个压缩表示，将其进一步输入到前馈神经网络的分类器中，即可完成分类。

不过简单的RNN模型一般训练效果不佳，在实际应用中，我们一般会使用RNN模型的拓展.

四、长短记忆模型

长短记忆模型

RNN模型在训练时，与当前距离较远的信息将在不断传递过程中被弱化，因而只具有短期记忆性。但是在真实情况中，较远的文本可能对于当前文本有良好的对应、解释关系。例如有些文章在开头的概述部分就会对整个文章进行介绍；又例如在有从句的英文语句中，某一些词语可能与靠前的主语存在关系而与中间插入的从句关系不大。因此长短期记忆模型（Long Short Term Memory, LSTM）应运而生。

LSTM的核心思想在于同时保留长期和短期的记忆，并通过学习保留重要的历史信息对当前语句进行建模。它主要需要处理两个任务：（1）去掉不再有用的信息，（2）保留对当前训练仍然有意义的信息。

长短记忆模型

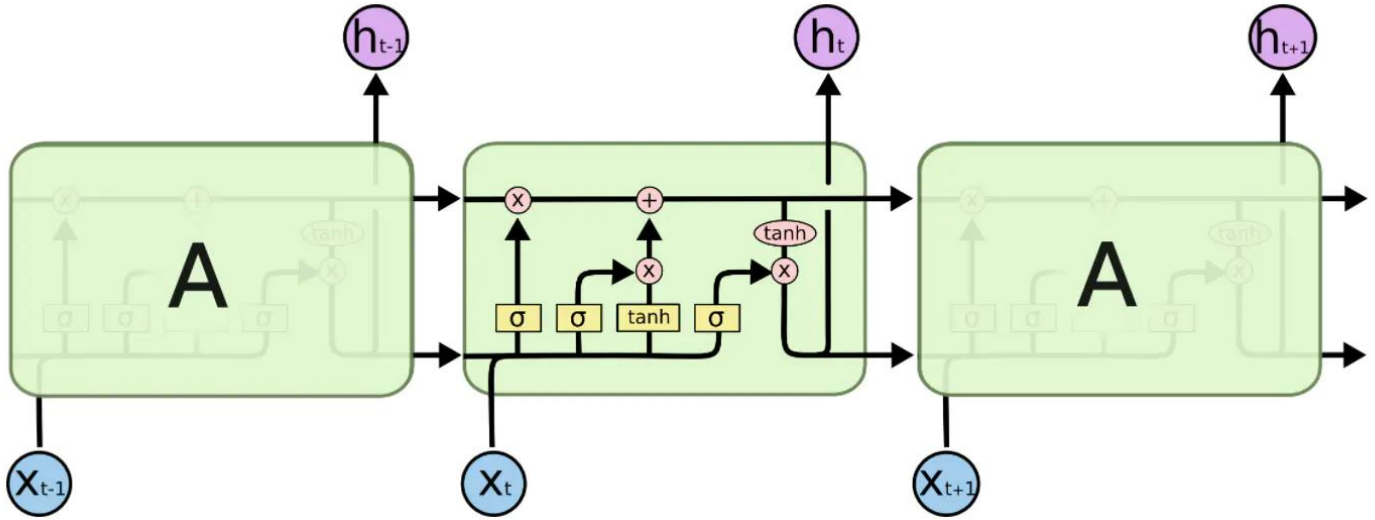
LSTM在原有的RNN模型中另外开辟了一个内存（cell state），用于存储历史信息，并判断哪些历史信息是应该保存的。是否保存历史信息通过“门”（gate）这一结构实现。形象的来说，如果这个门打开，那么其对应的历史信息就会被输入。

门一共有三种，即遗忘门、输入门、输出门，门的结构决定了门内储存的历史信息是否应该被利用。这些门是否打开根据当前的信息进行判断。

长短记忆模型

假设第t个状态的输入为x_t，输出为h_t，该状态的状态量为C_t。

- 在第一个σ的地方对应的是遗忘门，它的输入是h_{t-1}和x_t，分别对应的是上个状态的输出和该状态的输入，它的输出是一个概率，表示“需要遗忘的比例”；
- 第二个σ为输入门，决定信息应该“被记住的比例”；

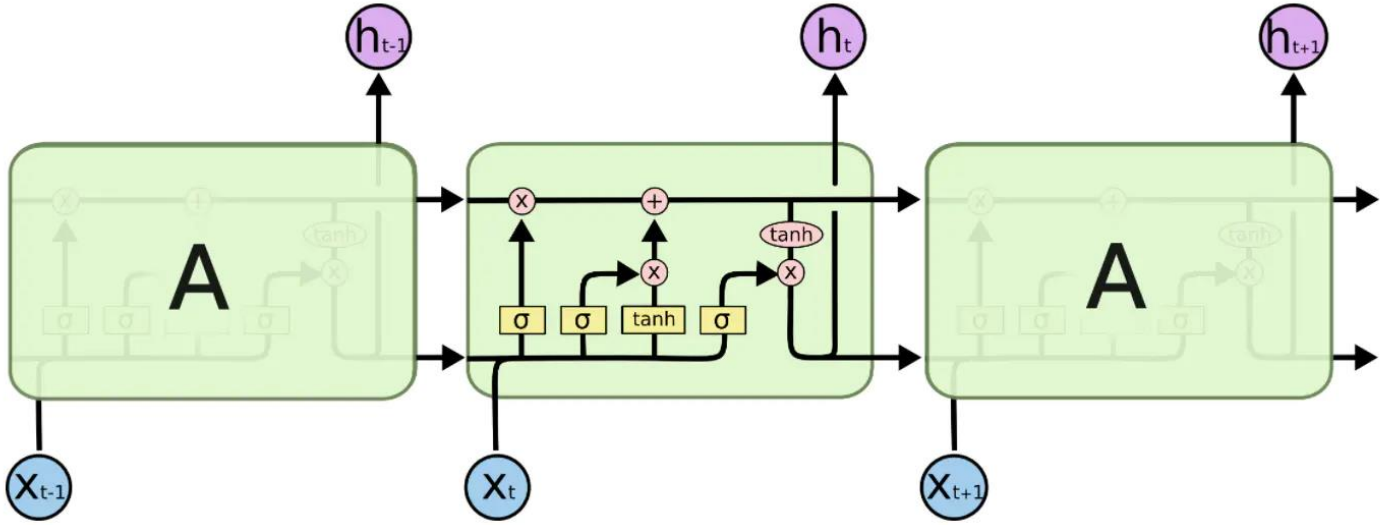


长短记忆模型

- 第二个σ右侧tanh激活函数，目的是形成新的信息，它的计算思想是：

  上一部分的信息 * 上一部分应该保存的信息比例 + 需要被记住的新信息比例；

- 第三个σ为输出门，即将需要利用的信息通过tanh函数写入h_t作为输出，而C_t作为该状态的状态量被传入下一个状态中。



长短记忆模型

LSTM的应用领域主要包括：文本生成、机器翻译、文本分类等。总体来说，LSTM能完成的机器学习任务与RNN基本类似，不过由于其能够选择性的保留长期特征的性质，在实际的模型拟合中，效果往往优于RNN模型。

除了LSTM以外，RNN还有很多其他的变形，例如Bidirectional RNN模型尝试从前向后和从后向前双向训练模型，通过上下文的共同作用，以期获得更好的训练效果。

五、卷积神经网络

卷积神经网络

卷积神经网络与循环神经网络是神经网络中最重要的两类基础模型。RNN模型由于其“循环”的特性，一般用于处理时间序列型的数据；而CNN模型在文本分析的应用中一般用于处理文本分类的问题（一维卷积）。

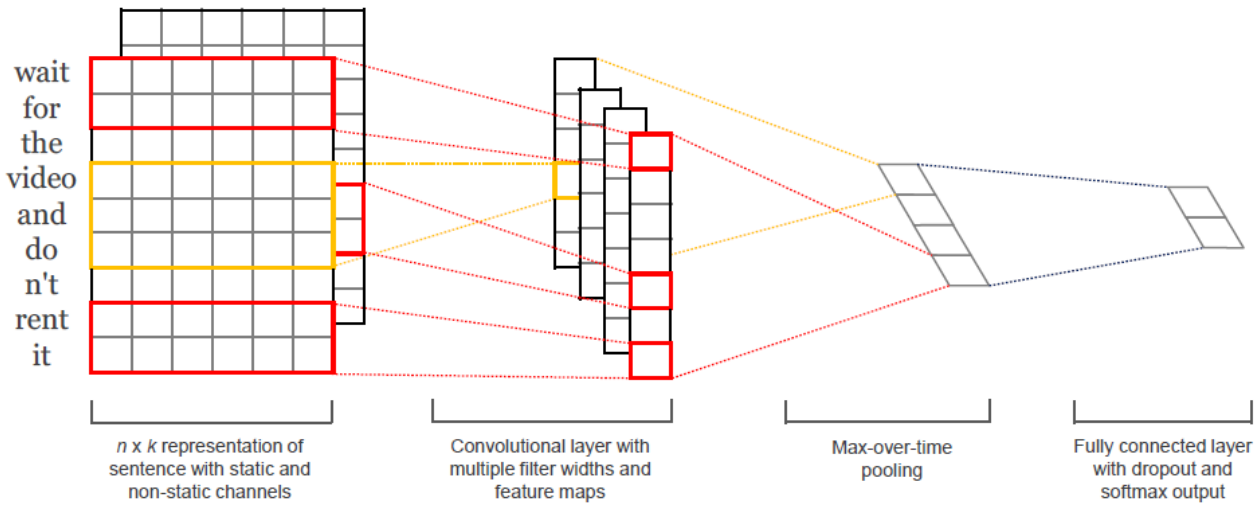
卷积神经网络（CNN）在图像处理上取得了巨大的成功。同样，CNN在文本分析和自然语言处理方面也有对应的应用情景。RNN及其衍生模型对于处理文本序列效果较佳，但是对于序列中的局部信息处理不尽如人意，因为这些模型需要较长的文本来学习其中的前后逻辑关系。而CNN恰好可以处理这一问题。在实际操作时，也常常将CNN模型和RNN模型结合使用，以期达到更好的预测效果。

卷积神经网络

CNN在进行图像处理时需要对矩阵进行卷积运算，采用的是二维卷积。在进行文本处理时，由于每一个词是一个一维向量，所以用CNN处理文本数据时，大部分时候使用的是一维卷积。类似于图像处理，在用CNN进行文本处理时，同样需要进行卷积、池化等操作。

卷积神经网络

Yoon Kim在2014年的文章Convolutional Neural Networks for Sentence Classification中就应用了卷积神经网络来进行文本的分类。具体的原理图如右图所示。令输入为对应词语的词向量 $x_i \in \mathbb{R}^k$ ，其中，k是词向量的维数，i表示词语所在文档中的位置。



卷积神经网络

首先进行卷积操作，第二部分是进行卷积操作以后的卷积层。卷积层是对输入数据进行特征提取，令其卷积核的权重系数为 $w \in \mathbb{R}^{h \times k}$，偏差为$b \in \mathbb{R}$，其中，$h$为窗口大小，即每一次对$h$个词向量进行卷积操作。则卷积向量的各个位置取值为$c_i = f(wx_{i:i+h-1} + b)$，其中$x_{i:i+h-1} \in \mathbb{R}^h$表示的是从文档中的第i个到第i+h-1个词语，$f(\cdot)$表示一个非线性变换，与前述前馈神经网络相同。由此类推可得卷积向量$c = [c_1, c_2, \dots, c_{n-h+1}] \in \mathbb{R}^{n-h+1}$。在卷积操作中会有多个卷积核，每一个卷积核会取不同的窗口大小，从图中第二部分可以看到其有四列，红色卷积对应的卷积尺寸为 $h = 2$，黄色卷积的尺寸为$h = 3$。

卷积神经网络

接下来进行伴随时间的最大值池化操作（Max-over-time pooling），即图中的第三部分，所谓最大池化就是在卷积核操作之后的结果中取最大值。最大池化之后的特征通常非常多，可以进一步通过dropout技术随机去掉部分特征，剩余特征形成全连接层，并最终经过softmax变换输出该文本在各个类别上的预测概率。

END