复旦大学

机器学习课程论文报告

“Deep Visual-Semantic Alignments for Generating Image Descriptions”

版本	更新时间	作者	更新内容
1.0	2015/11/08	严健康	初始版本
1.1	2015/11/18	严健康	补充中位排序概念和文档版本

严健康　**14210720109**

2015-11-18

目录

1. 论文背景

1.1 作者简介

我所选择的论文题目为 *Deep Visual-Semantic Alignments for Generating Image Descriptions*[1]，这是发表在 CVPR 2015 会议上的一篇文章，文章的作者是斯坦福大学计算科学系计算机视觉实验室的一名博士生 Andrej Karpathy 和他的导师 Li Fei-fei（李飞飞，斯坦福人工智能中心及视像实验室主任，计算机系第一位来自大陆的终身教授，主要研究在计算机视觉、机器学习、认知神经科学和人工智能），Karpathy 的主要研究方向是 Convolutional Neural Networks 和 Deep Learning，并且在最近几年的国际会议上发表了较多成果，包括 *ImageNet Large Scale Visual Recognition Challenge*（IJCV 2015），*Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping*（NIPS2014），*Large-Scale Video Classification with Convolutional Neural Networks*（CVPR2014），*Grounded Compositional Semantics for Finding and Describing Images with Sentences*（TACL2013），*Emergence of Object-Selective Features in Unsupervised Feature Learning*（NIPS2012）等。

1.2 论文摘要

对于人类来说，看到一张图片并很快识别中图中的基本元素，并用语言描述出来，是极其简单而自然的事情，但是对于机器来说，这是相关困难的事情。最近几年，随着深度学习的发展，利用神经网络模型来识别图像方面已经取得了很大的进展，同时在自然语言的处理方面也取得了很多的进展，但是如何用自然语言来描述图像依旧是很棘手的问题。因为对图像的描述，不仅要求能够识别出图像中的物体，还要能够识别出图像中物体之间的关系，以及物体所处的环境。同时自然语言的描述也意味着这个模型必须包含对自然语言的理解和生成能力，而自然语言的多义性也对语句描述的生成产生了困扰。 如下图所示，计算机必须识别出图像中的人、蔬菜和他们之间的关系 shopping，以及图片的环境 outdoor market。

A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

图 1-1 图像识别与语句描述生成实例[2]

本文主要致力于对图像的语句描述的生成。本论文主要介绍了两方面的内容。

一是**对其在 NIPS 2014 会议上发表的论文 *Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping*[3]中的算法进行了改进**。NIPS 2014 会议上的这篇论文主要介绍了一种将图像区域与图像描述语句中的词组关系进行双向映射和搜索的算法，如下图所示。图

中黄色背景的句子是人工生成的对图片内容进行描述的例子。通过将图片和语句描述输入作者实现的神经网络模型后，可以得到如下的图片区域与文本描述短语的对应关系（其中语句单词之间的连接关系是使用斯坦福大学的 CoreNLP parser 工具生成的，后文将介绍）。基于对该模型的大批量图片和对应语句数据的训练，作者实现了图片和文本的双向检索，并且提供了在线的 Demo 演示[4]。而 CVPR 2015 会议上的这篇论文对其中语句矢量模型的生成方式进行了改进，使用双向的递归神经网络（BRNN，Bidirectional recurrent neural network）来实现，并对图像区域-描述短语之间的匹配衡量指标和误差分析进行了改进，从而得到了更好的匹配结果。

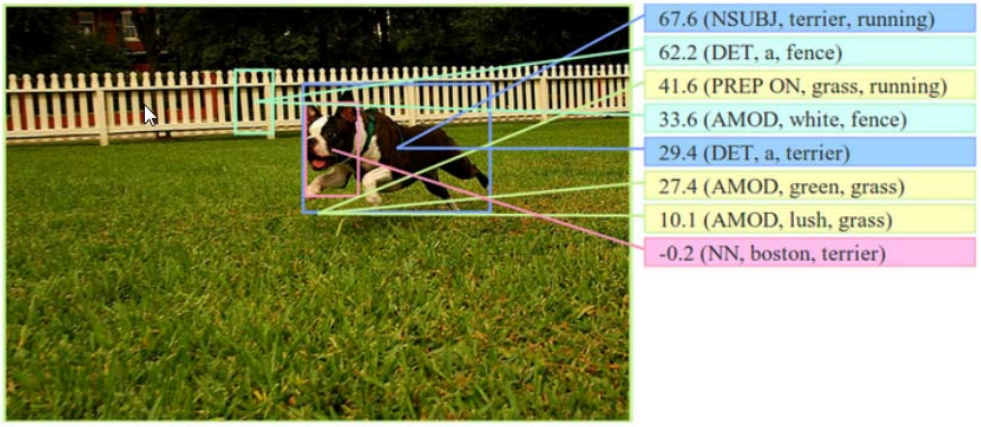


图 1-2 图像区域与图像描述语句中的词组关系进行双向映射和搜索的算法实例

二是**在其基础上实现了图像区域（或者全图）相对应的语句描述短语（或者整句）的生成**。即给定一副图像或者图像的一个区域，能够生成该图像的完整语句描述或者图像区域的短语描述。如下图所示。图 1-3 显示了图像区域与模型生成的文本描述短语的匹配，在这里，我们看到，这里不仅仅包含识别的物体，而且包含一些物体的运动情况，如人和走在走，或者形态描述等，如颜色修饰等。图 1-4 显示了对整幅图片自动生成的语句描述。作者采用的模型架构为 CNN+RNN，其中 CNN(Convolutional Neural Network)即卷积神经网络，用于将图片转化为空间矢量，而 RNN(Recurrent neural network)即递归神经网络，输入为图像向量和语句向量，递归神经网络会输出语句序列。

图 1-3 自动生成图像区域的文本描述示例

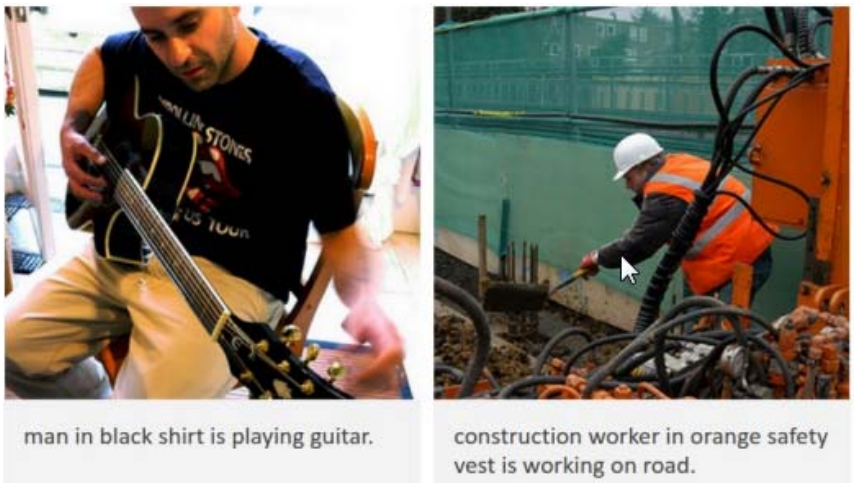


图 1-4 自动生成图像的描述语句示例

鉴于深度学习和图像识别技术的发展，很多高校和企业也都在进行与本文研究内容相似的研究。无独有偶，在 2015 年的 CVPR 会议上。Google 公司的机器学习研究人员 Vinyals 等也展示了他们的研究成果 *Show and Tell: A Neural Image Caption Generator*[2]，研究结果几乎与本文的完全相同，但他们的研究工作采用 CNN+LSTM 的架构组成，CNN 负责图像的编码，将输入图像转换为定长的空间矢量，然后 LSTM 用于解码生成语句。由于 LSTM 在序列化学习任务中具有更好的性能，因此 Google 的测试结果比 Karpathy 的测试结果稍好一些。同样的，在 ICML 2015 会议上，也有一篇文章 *Show, attend and tell: Neural image caption generation with visual attention*[5]也是实现了完全相同的功能，他们的研究工作也同样采用了 CNN+LSTM 的架构。这更加说明了目前这个领域是非常前沿，并且具有研究价值的。

1.3 本文安排

本文的基本结构安排如下：

第一章主要介绍论文的背景，包括作者简介和论文实现内容的简介；

第二章主要介绍论文中使用到的神经网络模型的基本背景知识，从而便于理解作者的系统设计；

第三章主要介绍作者论文中提到的整个系统框架的设计，其中主要包括两部分内容，第一节介绍作者在 NIPS 2014 会议上发表的图像区域与单词描述的双向检索模型及其改进方案；第二节主要介绍作者提出的图像（或者全图）相对应的语句描述短语（或者整句）生成模型；

第四章主要是作者对上述模型的测试结果及分析；

第五章对本文所包含的内容进行了总结，并对论文工作内容的进一步研究提出了一些个人意见；

最后为参考文献。

2. 神经网络介绍

2.1 神经网络

人工神经网络的研究在一定程度上是受到了生物学的启发，因为生物的学习系统是有相互连接的神经元组成的异常复杂的网络，而人工神经网络与此大体相似，它是由一系列简单的单元相互密集连接构成的，其中每一个单元有一定数量的实值输入（可能是其他单元的输出），并产生单一的实数值输出（可能成为其他很多单元的输入）。[6]

2.1.1 人工神经元

人工神经网络是以人工神经元为基础的，每一个人工神经元的基本结构如图 2-1 所示。假设这个神经元的编号为j， 神经元的输入可以是原始输入，也可以是上一个神经元的输入，我们记为$x_{j'}$，从神经元j'到神经元j的连接边权值为$w_{jj'}$（jj'代表从j'输入j），然后首先经过一个加权的加法器，输出记为$a_j = \sum_{j'} (w_{jj'} x_{j'})$，然后再经过一个激活函数σ(x)，最终得到的神经元的输出为$\sigma(a_j)$。

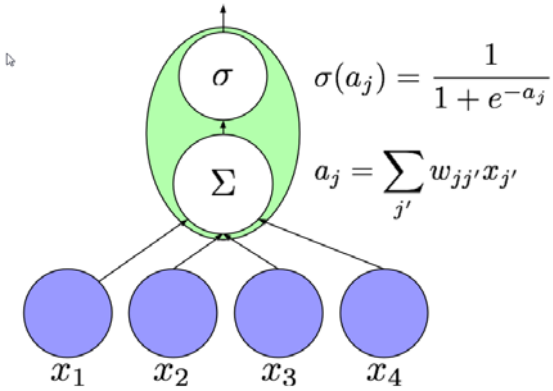


图 2-1 一个神经元的基本结构[7]

其中激活函数的作用在于模拟非线性，因为神经网络的作用在于模拟现实世界，而加权和是线性结构，多个线性网络的连接仍然是线性函数，这是远远不够，因此需要引入非线性函数。同时，激活函数一般也充当了限幅和调幅的作用。常用的激活函数有以下三种：

（1） sigmoid 函数：$\sigma(z) = \frac{1}{1+e^{-z}}$

（2） tanh 函数：$\emptyset(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

（3） Relu 函数：$l(z)=\max(0,z)$

其中 Relu 函数是在最近的论文中被证明能有效的提高深度神经网络的性能[8]。

有时候，不同神经网络训练任务的不同，将对输出有不同的要求。比如在用于分类时，有 K 个可选的类别，那么上述激活函数的输出并不能符合要求。这时候经常使用的一中分类器为 Softmax 回归模型，其表达式如下所示：

$$\widehat{y_k}=\frac{e^{a_k}}{\sum_{k\prime=1}^{K}e^{a_{k\prime}}},\qquad k=1\ldots K$$

其中$\widehat{y_k}$代表第 k 个输出节点的输出，a_k为第 k 个输出节点在经过激活函数输出前的计算值。分母是所有输出节点的 e 指数的总和。

多个神经元的前向全连接组成了一个前馈神经网络，如图 2-2 所示。其中蓝色的斑点代表输入层，红色的为输出层，中间为隐层。

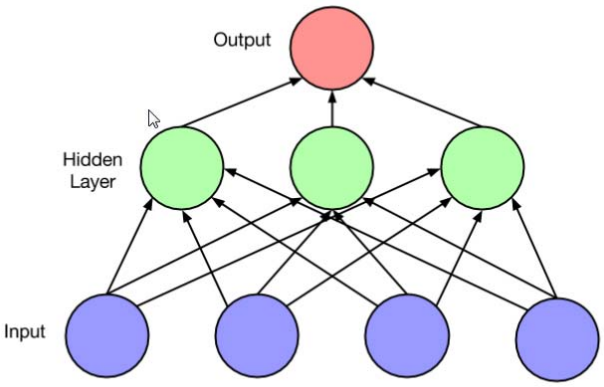


图 2-2 一个两层的前馈神经网络示意图

2.1.2 梯度下降准则

从上面的介绍中我们可以知道，人工神经网络的难点在于每条连接边权值的确定方法。假设我们使用随机初始化或者指定的初始权值w_{ji}（代表从神经元 i 到 j 的边），那么在给定训练数据$< x_k, y_k >$后，神经网络的计算结果$\widehat{y_k}$与y_k之间存在误差，如何表示这个误差。

我们以最简单的线性神经元为例，假设$o(\vec{x})=\vec{w}\cdot\vec{x}$, 即输出为加权和，对于训练数据 D 中的每一组训练数据$< x_d, t_d >$，那么可以定义该神经元在该训练集上的误差为：

$$E(\vec{w})=\frac{1}{2}\sum_{d\in D}(t_d-o_d)^2$$

其中o_d是线性神经元对输入x_d的输出。训练数据的目的就是为了使这个误差最小化。那么如何寻找最优权值。常用的方法是梯度下降算法。可以可视化的表示包含所有可能的权向量和相关联的 E 值得整个假设空间，如下图 2-3 所示。这里坐标轴ω_0，ω_1表示一个简单的线性单元中的两个权所有可能的取值。在误差 E 计算方法已知的情况下，使 E 最小即需要找到一个全局最低点。梯度下降搜索确定一个使 E 最小化的权向量是从一个任意的初始权向量开始，然后以很小的步伐，反复修改这个向量，每一步都沿误差曲面产生最陡峭下降的方向修改权向量，继续这个过程直到得到全局的最小误差点。[6]

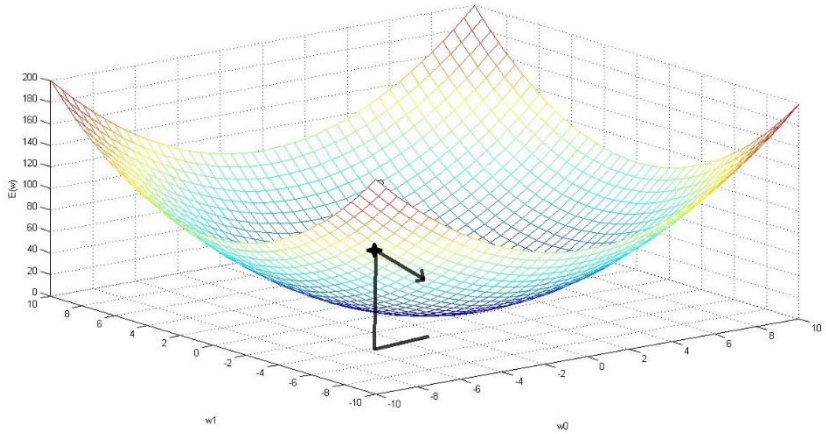


图 2-3 权向量和相关联的 E 值对应的假设空间示意图

因此，对于单个神经元来说，梯度下降的训练法则为：

$$\vec{\omega} \leftarrow \vec{\omega} + \Delta\vec{\omega}$$

$$\Delta\vec{\omega_i} = -\eta\frac{\partial \mathrm{E}}{\partial \omega_i}$$

其中η是一个正常数，叫做学习速率，它决定梯度下降搜索中的步长，其中偏导数可以根据输出的表达式求得。

上述 E 误差为对整个训练集进行误差计算，因此权值的更新也是在整个训练集完成之后，这通常不能保证找到全局最优解，如果存在多个局部最优解的时候，而且收敛过程可能相当漫长。有一种称之为随机梯度下降（Stochastic Gradient Descent, SDG）的方法，是通过为每个单独的训练样例 d 定义单独的误差函数$E_d(\vec{\omega})$，然后在每次训练后都进行权值更新，这样有可能通过随机的扰动，来跳过局部最优解，从而更快的收敛。但是这种方法权值更新频繁，运算量太大，实际中常使用 mini-batch SGD，即折中后的批量式随机梯度下降。

2.1.3 反向传播算法(Backpropagation algorithm, BP)

反向传播算法首次给出了如何有效的训练人工神经网络的方法，从而导致了人工神经网络研究的热潮。如何根据传递误差来调整多层神经网络中各条边的权重是 BP 算法的核心思想。

BP 算法由信号的正向传播和误差的反向传播两个过程组成。正向传播时，输入样本从输入层进入网络，经隐层逐层传递至输出层，如果输出层的实际输出与期望输出存在的误差大于误差阈值，则转至反向误差传播，否则算法结束；

反向误差传播时，将输出误差按原通路反传计算，通过隐层反向直至输入层，在反传过程中将误差分摊给各层的各个单元，获得各层各单元的误差信号，并将其作为修正各单元权值的根据。这一过程使用随机梯度下降准则来完成。根本原则就是求导的链式法则，如下图所示。

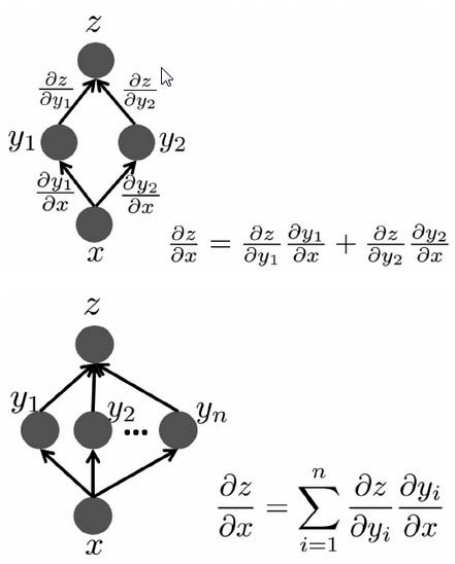


图 2-4 求导链式法则在神经网络中的应用

首先对于多层网络，我们同样可定义误差函数如下：

$$\mathrm{E}_d(\vec{\omega}) = \frac{1}{2}\sum_{k}^{o}(d_k - y_k)^2$$

其中 d 是训练数据集的实际输出，y 为神经网络的计算输出，o 是神经网络输出单元数。注意此处使用的是随机梯度下降，即对于每一个样本输入 d，都有对应的误差表达式。

为了推导反向传播算法，我们首先做如下约定：

- x_{ji} = 单元 j 的第 i 个输入
- w_{ji} = 与单元 j 的第 i 个输入相关联的权值
- $net_j = \sum_i w_{ji}\mathrm{x}_{ji}$ （单元 j 的输入加权和）
- d_j = 单元 j 的计算输出
- y_j = 单元 j 的目标输出
- g 函数代表输出层的激活函数，f 函数代表隐层的激活函数

根据梯度下降准则，我们可以得到：

$$\Delta(\omega) = -\eta\frac{\partial(\mathrm{E}_d)}{\partial(\omega)}$$

注意权值w_{ji}仅能通过net_j来影响网络的其他部分，所以根据链式法则有：

$$\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{w}_{ji})} = \frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}\frac{\partial(\mathrm{net}_j)}{\partial(\mathrm{w}_{ji})}$$
$$= \frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}\mathrm{x}_{ji}$$

1） 对于输出单元来说：

$$\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}=\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{y}_j)}\frac{\partial(\mathrm{y}_j)}{\partial(\mathrm{net}_j)}$$
$$=\frac{\partial(\mathrm{E}_d)}{\partial(y_j)}g'(\mathrm{net}_j)$$

$$\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{y}_j)}=\frac{\partial}{\partial y_j}\frac{1}{2}\sum_k^o(d_k-y_k)^2$$
$$=-(d_j-y_j)$$

因此：

$$\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}=-(d_j-y_j)\,g'(\mathrm{net}_j)$$

记：

$$\delta_j=-\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}=(d_j-y_j)\,g'(\mathrm{net}_j)$$

2） 对于隐层来说，权值w_{ji}通过 j 单元的所有直接下游单元的集合(也就是直接输入中包含单元 j 的输出中的所有单元，记为Downstream(j))来影响网络输出。

$$\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}=\sum_{k\epsilon\mathrm{Downstream(j)}}\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_{kj})}\frac{\partial(\mathrm{net}_k)}{\partial(\mathrm{net}_j)}$$

$$=\sum_{k\epsilon\mathrm{Downstream(j)}}-\delta_k\frac{\partial(\mathrm{net}_k)}{\partial(\mathrm{net}_j)}$$

$$=\sum_{k\epsilon\mathrm{Downstream(j)}}-\delta_k\frac{\partial(\mathrm{net}_k)}{\partial(\mathrm{y}_j)}\frac{\partial(\mathrm{y}_j)}{\partial(\mathrm{net}_j)}$$

$$=\sum_{k\epsilon\mathrm{Downstream(j)}}-\delta_k\omega_{kj}\;f'(\mathrm{net}_j)$$

$$=\sum_k^o-\delta_k\omega_{kj}\,f'(\mathrm{net}_j)$$

此处，假设Downstream(j) = outpus，对于其他情况可依此类推。

重新组织后可得：

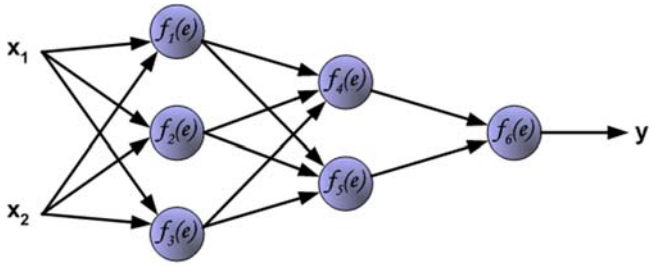
$$\delta_j=-\frac{\partial(\mathrm{E}_d)}{\partial(\mathrm{net}_j)}=\sum_k^o\delta_k\omega_{kj}\,f'(\mathrm{net}_j)$$

综上所述，可以得到：

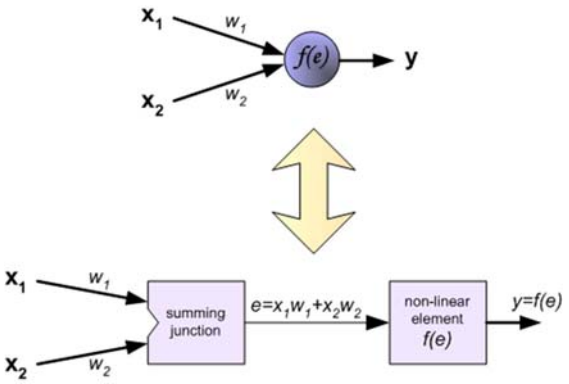
$$\delta_j=\begin{cases}(d_j-y_j)g'(\mathrm{net}_j) & ，\ \mathrm{j}\text{ 为输出单元}\\ \sum_k^o\delta_k\omega_{kj}\,f'(\mathrm{net}_j) & ，\ \mathrm{j}\text{ 为隐藏单元}\end{cases}$$

$$\Delta\omega_{ji}=\eta\delta_j\mathrm{x}_{ji}$$

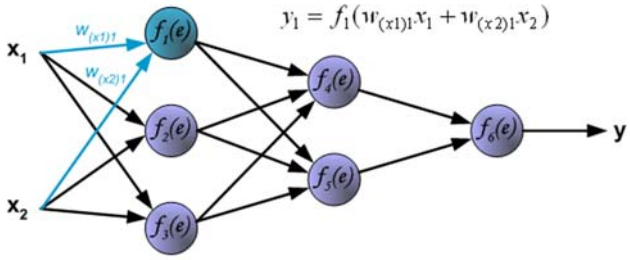
可以通过下面的例子来理解这一过程，这个实例来自[9]。具体的理论证明可以参考[6]。



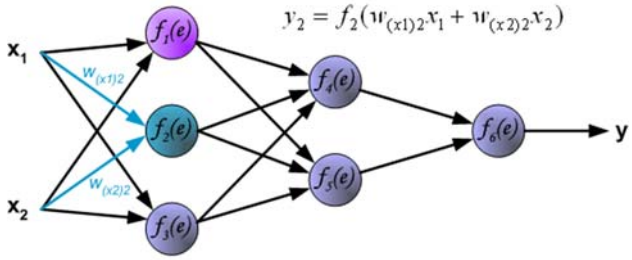
(1) 这是一个三层神经网络的示例



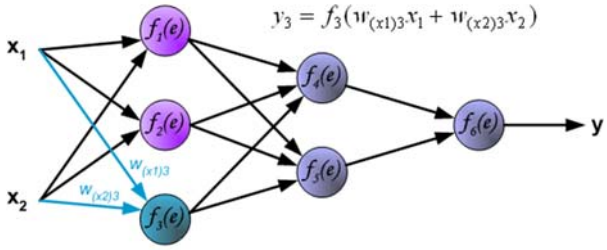
(2) 其中神经元节点的定义如图所示



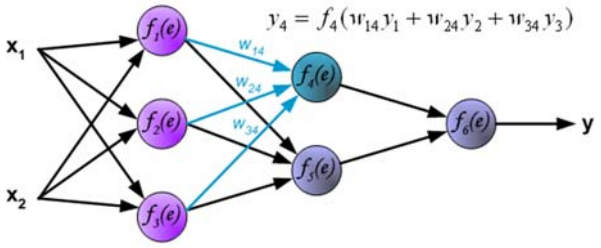
(3) 第一层第一个节点的计算方式



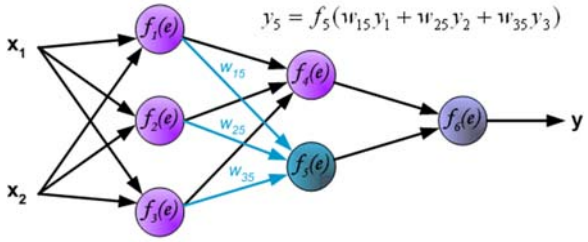
(4) 第一层第二个节点的计算方式



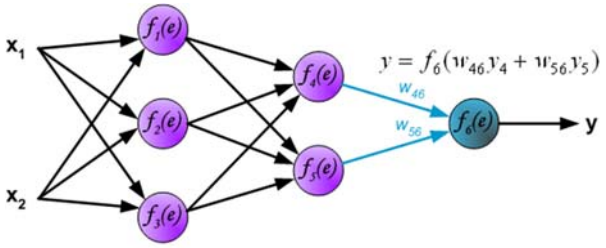
(5) 第一层第三个节点的计算方式



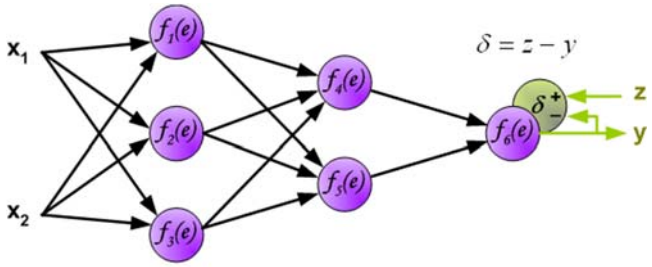
(6) 第二层第一个节点的计算方式



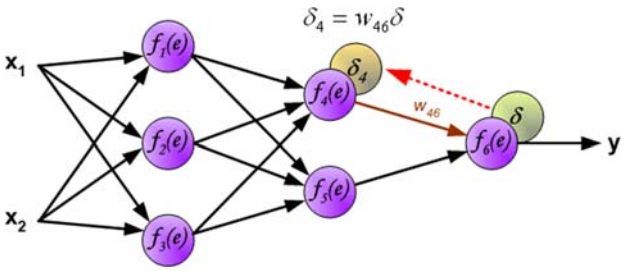
(7) 第二层第二个节点的计算方式



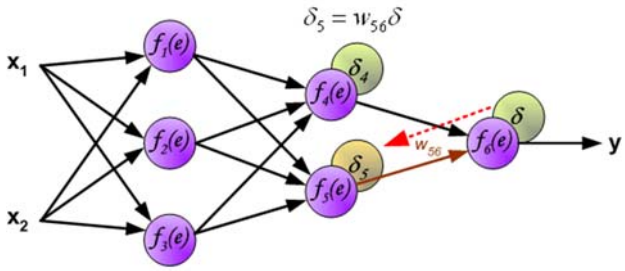
(8) 输出节点的计算方式



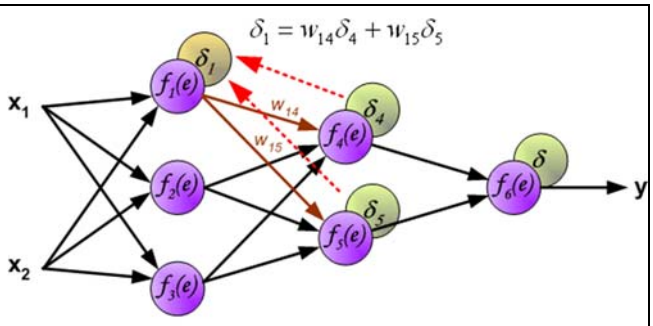
(9) 输出误差的计算。



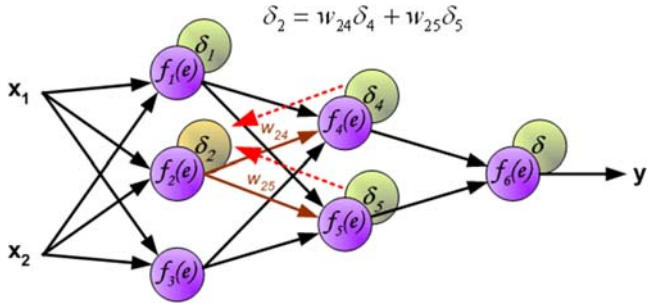
(10) 误差向隐层传递，因为没有办法直接计算内部节点的误差



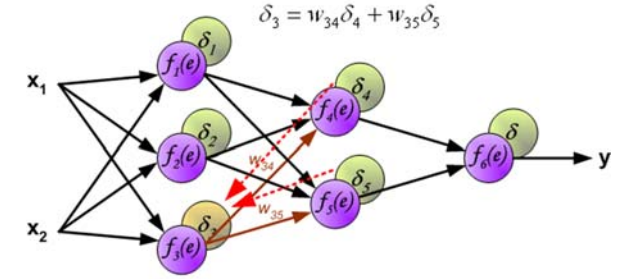
(11)误差向隐层传递



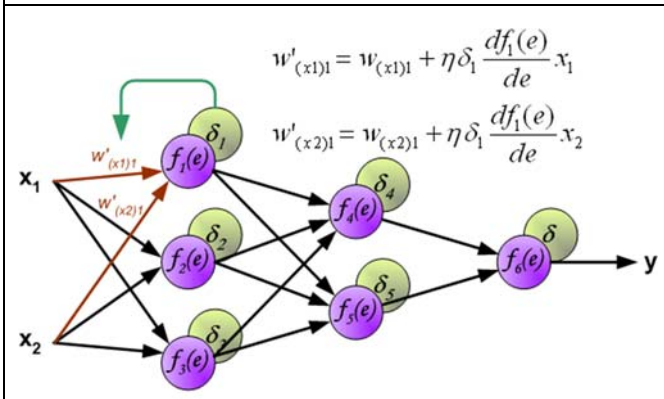
(12)误差继续向下一级传递



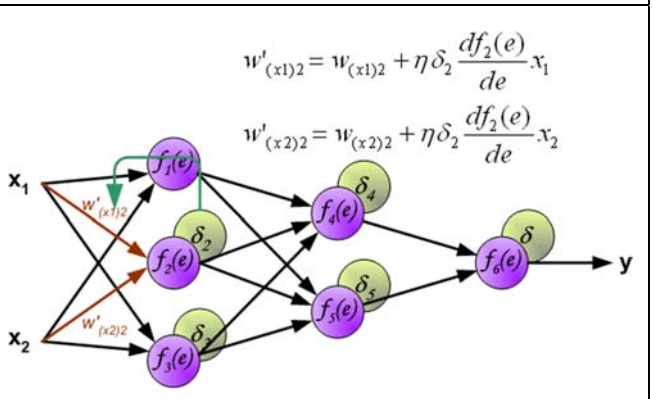
(13)误差向第一层第二个节点传递



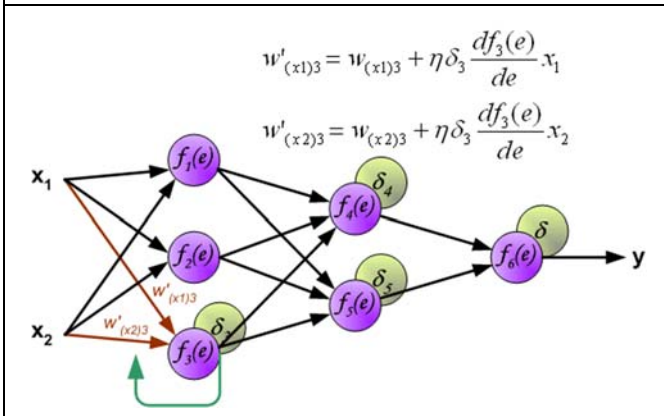
(14)误差向第一层第三个节点传递



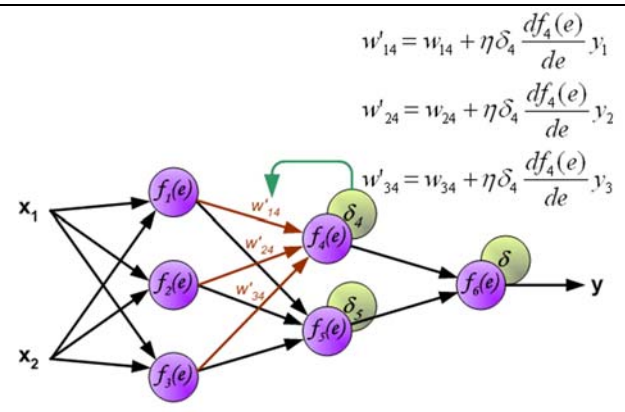
(15)计算第一层节点权值的更新



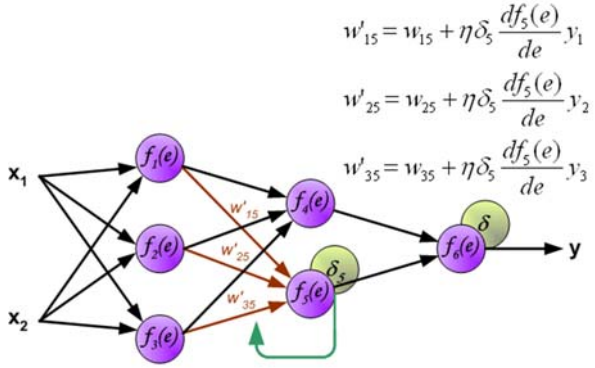
(16)计算第一层节点权值的更新



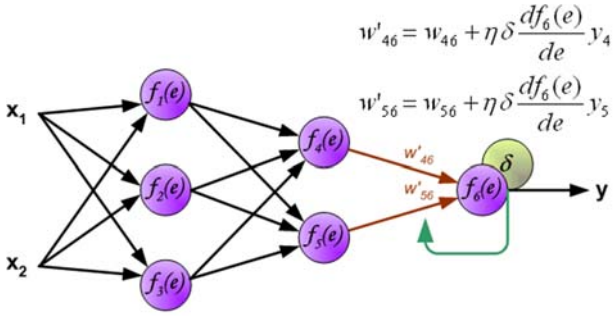
(17)计算第一层节点权值的更新



(18)计算第二层节点权值的更新



(19)计算第二层节点权值的更新



(20)计算第三层节点权值的更新

2.1.4 误差函数

上面介绍的常用的误差函数为平方差函数。但是，在使用 sigmod 激活函数σ的时候，由于σ'在两端的部分几乎是平坦的，会导致权值更新速度很慢，为了克服这个缺点，交叉熵代价函数（cross-entropy）被引入，表达式如下：

$$\mathrm{Cost}=-\sum_{p}^{n}\sum_{k}^{0}(d_{pk}\ln(y_{pk})+(1-d_{pk})\ln(1-y_{pk}))$$

其中 d 是训练数据集的实际输出，y 为神经网络的计算输出，o 是神经网络输出单元数，n 是训练样本总数，d_{pk}代表了第 k 个输出单元在第 p 个训练样本输入下的输出。

这个误差函数在使用 sigmoid 激活函数时，输出梯度有很好的特性：

$$\delta_{pj}=-\frac{\partial(\mathrm{C})}{\partial(\mathrm{net}_{pj})}=-\frac{\partial(\mathrm{C})}{\partial(\mathrm{y}_{pj})}\frac{\partial(\mathrm{y}_{pj})}{\partial(\mathrm{net}_{pj})}=\mathrm{d}_{pj}-\mathrm{y}_{pj}$$

这里可以看出导数中没有σ'的影响，直接受误差的影响，当误差大的时候权重更新快，误差小的时候权重更新小。

2.2 卷积神经网络 CNN

卷积神经网络是人工神经网络的一种，已成为当前语音分析和图像识别领域的研究热点。它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。该优点在网络的输入是多维图像时表现的更为明显，使图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。[10]

在图像处理中，往往把图像表示为像素的向量，比如一个 1000×1000 的图像，可以表示为一个 1000000 的向量。如果隐含层数目与输入层一样，即也是 1000000 时，那么输入层到隐含层的参数数据为 1000000×1000000=10^12，这样就太多了，基本没法训练。因此卷积神经网络就在于通过局部感知野和权值共享来减少参数数目。

2.2.1 局部感知

一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息 [10]。如图 2-5 所示。左边为全连接，右边为局部连接，每个神经元只连接一个 10x10 的区域。这样只需要 10^8 个参数。局部连接一般用在 CNN 的特征提取层，每个神经元的输入与前一层

的局部接受域相连，并提取该局部的特征。一旦该局部特征被提取后，它与其它特征间的位置关系也随之确定下来。

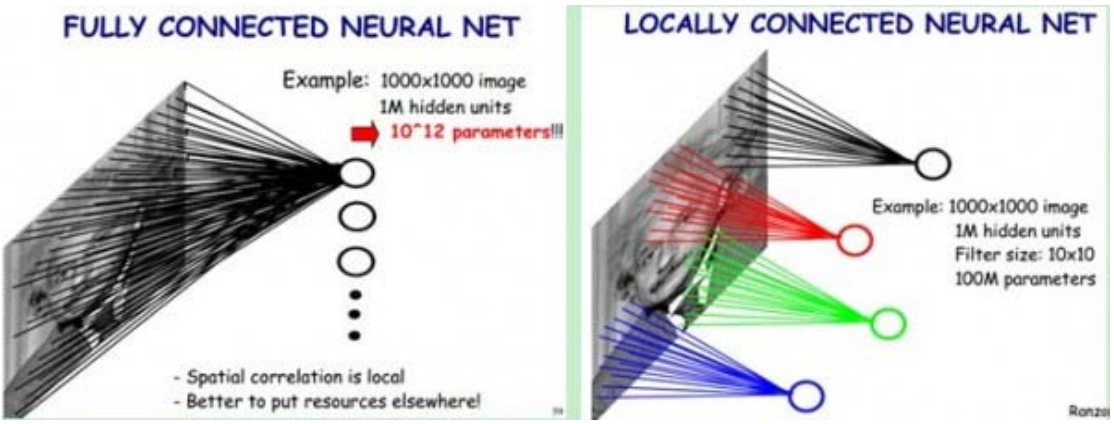


图 2-5 左图为全连接，右图为局部连接

2.2.2 参数共享

在上面的局部连接中，每个神经元都对应 100 个参数，一共 1000000 个神经元，如果这 1000000 个神经元的 100 个参数都是相等的，那么参数数目就变为 100 了。我们可以将这 100 个参数（也就是卷积操作）看成是提取特征的方式，该方式与位置无关。这其中隐含的原理则是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。当从一个大尺寸图像中随机选取一小块，比如说 8×8 作为样本，并且从这个小块样本中学习到了一些特征，这时我们可以把从这个 8×8 样本中学习到的特征作为探测器，应用到这个图像的任意地方中去，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。这其实就是一个做卷积的过程，权矩阵称为卷积核，对同一副图片可以使用多个卷积核。

2.2.3 Down-pooling

在通过卷积获得了特征 (features) 之后，下一步我们希望利用这些特征去做分类。理论上讲，人们可以用所有提取得到的特征去训练分类器，但这样做面临计算量的挑战，而且容易出现过拟合。一个解决办法就是对不同位置的特征进行聚合统计，例如，我们可以计算图像中一个区域上的某个特定特征的平均值 (或最大值)。这些概要统计特征不仅具有低得多的维度，同时还会改善结果(不容易过拟合)。这种聚合的操作就叫做池化 (pooling)，有时也称为平均池化或者最大池化 (取决于计算池化的方法)。图 2-6 给出了一个 Down-pooling 的实例，使用的是一个 2x2 的计算最大值的 pool。

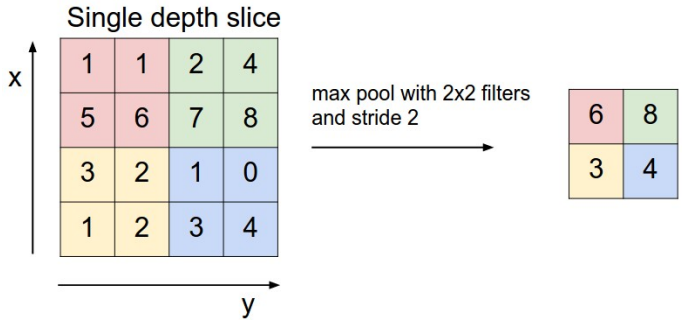


图 2-6 Down-pooling 的实例，使用一个 2x2 的 max pool

最终，我们可以得到一个典型的卷积神经网络的结构，如图 2-7 所示。输入图像首先经过三个卷积核卷积后在 C1 层产生三个特征映射图，然后特征映射图中再经过 Down-pooling 得到三个 S2 层的特征映射图。这些映射图再经过卷积得到 C3 层。C3 再经过下采样产生 S4。最终，这些像素值被光栅化，并连接成一个向量输入到传统的神经网络，得到输出。

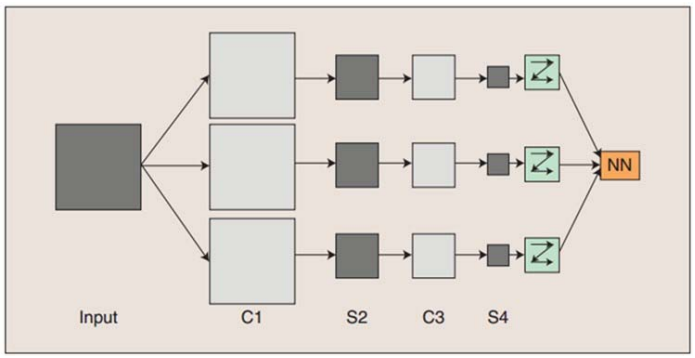


图 2-7 **卷积神经网络的典型结构** [10]

2.3 递归神经网络（Recurrent Neural Network, RNN）

传统的神经网络（包括 CNN）的局限之处在于：

1） 它们将固定大小的向量作为输入，然后输出一个固定大小的向量，而且模型中的层级关系一旦训练好之后都是固定的，因此传统的神经网络无法处理序列化的输入输出[11]；

2） 传统的神经网络依赖于训练数据与测试数据的独立性。模型一旦训练好之后，每一个测试用例处理完之后，测试过程中的网络状态即被丢失。如果测试集中的每个测试用例之间存在关联，那么传统的网络模型是不能接受的 [7]。

递归神经网络 RNN 主要指时间上的递归（还有一种结构上的递归，称为 Recursive Neural Network，这里不予考虑），它可以描述动态时间行为，因为和前馈神经网络（feedforward neural network）接受较特定结构的输入不同，RNN 将状态在自身网络中循环传递，因此可以接受更广泛的时间序列结构输入。手写识别是最早成功利用 RNN 的研究结果[12]。

2.3.1 基本结构

RNN 与传统 NN 的不同之处在于，其网络中存在从输出单元流回隐藏单元的边。一个简单的递归神经网络的例子如图 2-8 所示。

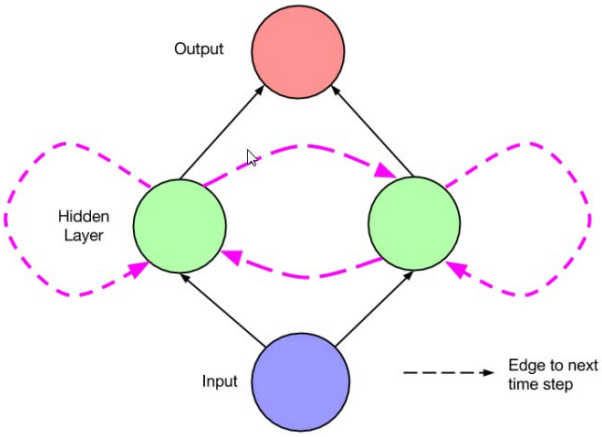


图 2-8 一个简单的 RNN 网络实例 [7]

可以看到隐层的输出不仅到达输出层，而且回流到自身和相邻隐层。因此 RNN 网络中存在时间变量 t，我们把 t 时刻的输入数据记为$\mathrm{x}^{(t)}$，隐层的上一个时间点输出$\mathrm{h}^{(t-1)}$与$\mathrm{x}^{(t)}$同时输入产生新的隐层输出。可以下面的两个方程来表示 RNN 的基本原理：

$$h^{(t)} = \sigma\left(W^{hx}\mathrm{x}^{(t)} + \mathrm{W}^{hh}\mathrm{h}^{(t-1)}\right) + b_h$$

$$\hat{y}^{(t)} = softmax(\mathrm{W}^{yh}\mathrm{h}^{(t)} + b_y)$$

其中W^{hx}是输入层到隐层的权值向量，W^{hh}是隐层反馈到隐层的权值向量，W^{yh}是隐层到输出层的权值向量，b_h是隐层的偏置向量，b_y是输出层的偏置向量。$h^{(t)}$是 t 时刻的隐层输出，$\hat{y}^{(t)}$是 t 时刻的输出层输出。激活函数的选取可以视具体情况而定。

2.3.2 训练方法

在传统的神经网络，我们可以通过误差一层层反传来训练网络权值，在 RNN 中隐层的反馈使得其不能直接使用 BP 算法进行权值更新。一种可能的解决方案是基于时间展开 RNN，如图 2-9 所示。这样的话，如果我们将 t=2 看成是 t=1 的上一层，那么这个网络就变成了普通的神经网络，就可以使用 BP 算法来进行权值更新，这种 BP 算法称之为 BPTT（Backpropagation Through Time）。

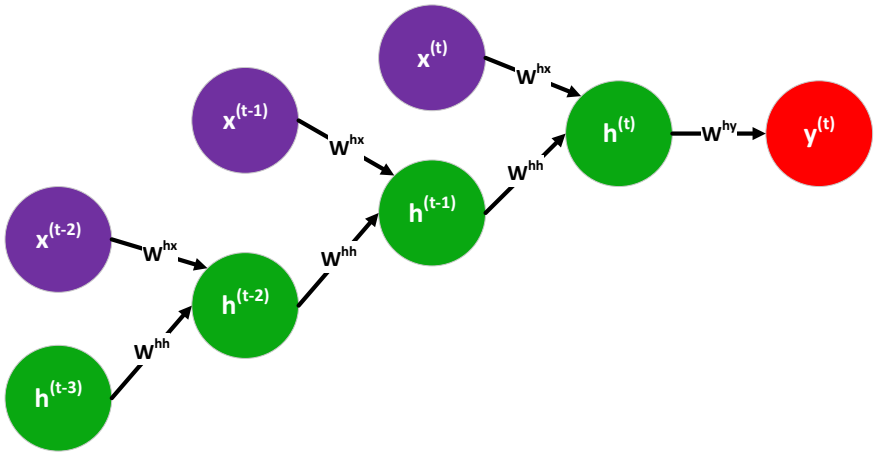


图 2-9 RNN 基于时间展开的网络图

为了展开网络，我们假定在展开步长内权值是相同的，这类似于 Mini-batch SGD。展开步长记为 T，根据前面推导的 Backpropagation 权值更新公式，BPTT 更新的相关公式为：

$$\delta_j^{(t-1)} = \sum_k^o \delta_k^{(t)} W^{hh} f'(\mathrm{h}(\mathrm{t}-1))$$

其中，k 代表 t 时刻的隐层，j 是 t-1 时刻隐层单元，o 是 t 时刻所有的输出。最终更新的权值作为折叠后的权值一次更新。

具体的更新步骤可以用矩阵表示如下(此处只考虑 T=1)：

t 时刻误差为：

$$\mathrm{e}_o(t) = \mathrm{d}(\mathrm{t}) - \mathrm{y}(\mathrm{t})$$

输出矩阵W^{yh}更新为：

$$\mathrm{W}^{yh}(\mathrm{t}+1) = \mathrm{W}^{yh}(t) + \eta \mathrm{h}(\mathrm{t}) \mathrm{e}_o(t)^T$$

然后，梯度误差传递到输出层后的隐层：

$$\mathrm{e}_h(t) = \mathrm{e}_o(t)^T W^{hy} f'(\mathrm{net}_j)$$

其中net_j代表 t 时刻 j 节点(产生 h(t)的节点)的线性加权输入，f 为 j 节点的激活函数。

因此输入 x(t)到隐层 h(t)的权值矩阵更新为：

$$\mathrm{W}^{hx}(\mathrm{t}+1) = \mathrm{W}^{hx}(t) + \eta \mathrm{x}(\mathrm{t}) \mathrm{e}_h(t)^T$$

输入 h(t-1)到隐层 h(t)的权值矩阵更新为：

$$\mathrm{W}^{hh}(\mathrm{t}+1) = \mathrm{W}^{hh}(t) + \eta \mathrm{h}(\mathrm{t}-1) \mathrm{e}_h(t)^T$$

依此类推，当**展开层数为 T** 时，BPTT 算法就这样递归的展开：

记：

$$\mathrm{d}_{hj}(x,t) = x f'(net_j)$$

其中 j 为 t 时刻输出 h(t)的隐层单元，net_j为隐层单元的线性加权输入。

$$\mathrm{e}_h(t-\tau-1) = d_h(\mathrm{e}_h(t-\tau)^T \mathrm{W}^{hh}, t-\tau-1)$$

这样最终的W^{hx}和W^{hh}的更新表达式为：

$$\mathrm{W}^{hx}(\mathrm{t}+1) = \mathrm{W}^{hx}(t) + \eta \sum_{z=0}^{T} \mathrm{x}(\mathrm{t}-\mathrm{z}) \mathrm{e}_h(t-z)^T$$

$$\mathrm{W}^{hh}(\mathrm{t}+1) = \mathrm{W}^{hh}(t) + \eta \sum_{z=0}^{T} \mathrm{h}(\mathrm{t}-\mathrm{z}-1) \mathrm{e}_h(t-z)^T$$

上述推导参考[13]。

2.3.3 应用模型

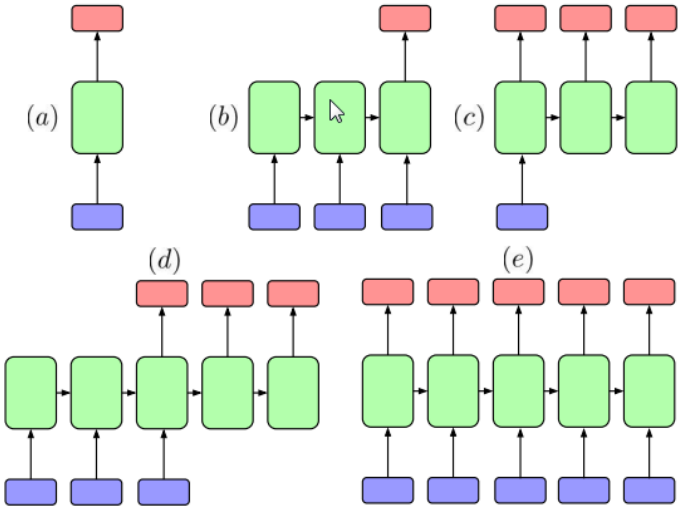


图 2-10 RNN 使用模型 [7]

RNN 已经在很多序列化的学习中发挥了重要作用。如图 2-10 所示，蓝色矩形代表输入层，红色矩形代表输出层，绿色矩阵代表隐藏层的神经元，箭头代表信息流向。(a)为没有使用 RNN 的普通神经模型，从固定大小的输入得到固定大小输出（比如图像分类）；(b)为序列输入（比如情感分析，输入一段文字然后将它分类成积极或者消极情感）；(c)为序列输出（比如图片字幕，输入一张图片输出一段文字序列）；(d)为异步的序列输入和序列输出（比如机器翻译：一个 RNN 读取一条英文语句然后将它以法语形式输出）；(e)为同步序列输入输出（比如视频分类，对视频中每一帧打标签）。我们可以注意到在每一个案例中，都没有对序列长度进行预先特定约束。

2.3.4 存在的问题

RNN 不能解决长期依赖问题， 容易导致“梯度的爆发与消失”（exploding and vanishing gradients）。可以通过以下图和表达式简单理解。

图 2-11 梯度消息现象图解

$$\frac{\partial E_t}{\partial \omega}=\sum_{k=1}^{t}\frac{\partial E}{\partial y_t}\frac{\partial y_t}{\partial h_t}\frac{\partial h_t}{\partial h_k}\frac{\partial h_k}{\partial \omega}$$

$$\frac{\partial h_t}{\partial h_k}=\prod_{i=k+1}^{t}\frac{\partial h_i}{\partial h_{i-1}}$$

对于上图而言，对 t-3 时刻的权值更新为(假设激活函数为 sigmoid 函数)：

$$\frac{\partial \mathrm{C}}{\partial net_{(t-3)}}=\frac{\partial \mathrm{C}}{\partial \mathrm{y(t)}}\cdot\omega_{hy}(t)\cdot\omega_{hh}(t)\cdot\sigma'(t-1)\cdot\omega_{hh}(t-1)\cdot\sigma'(t-2)\cdot\omega_{hh}(t-2)\cdot\sigma'(t-3)\cdot\omega_{hh}(t-3)$$

当展开层数过多时，如果$||\omega_{hh}|| < 1$,那么后级的误差梯度会很快减到 0，即称为梯度消息现

象，如果$||\omega_{hh}|| > 1$，那么会出现误差爆炸现象，这两者都会导致 RNN 训练过程的失败。也就是说，RNN 还不能处理长期依赖问题。

2.4 长短时记忆网络(Long-Short Term Memory Network, LTSM)

LTSM 是一种特殊的 RNN，它能够学习长时间依赖，最早由 Hochreiter 和 Schmidhuber 在 1997 年提出[14]，后来很多人加以改进和推广，现在已经被广泛使用。

2.4.1 基本单元结构

LSTM 的基本单元结构如下图 2-12 所示，它有以下基本单元：

1） **输入神经单元：**图中标识为$g_c^{(t)}$的单元，它从 t 时刻的输入层$x^{(t)}$或者t-1 时刻的隐层输出$h^{(t-1)}$获得输入数据，加权求和后经过一个激活函数(tanh 或者 sigmoid)；

2） **输入门：**图中标识为$i_c^{(t)}$的单元，它是一个 sigmoid 单元，像输入单元一样从$x^{(t)}$或者t-1 时刻的隐层输出$h^{(t-1)}$获得输入数据。 它之所以称之为门是因为它的值是用来乘以另一个节点的值从而控制节点输出。如果门的输出为 0，那么输入节点的值即被阻塞无法输入；如果门的输出为 1，那么输入节点的值就可以完全输入；

3） **内部状态单元：**内部状态单元是用来储存数据的，是图中标识为$s_c^{(t)}$的单元，在不考虑遗忘门的情况下，它的输出值在下一个时间点会经过一条权重为 1 的边反馈到自身的输入，权值为 1 刚好解决了我们在 RNN 网络中遇到的梯度消失和爆炸问题。内部状态单元也称为 CEC（Constant Error Carousel，恒定误差流）。$s_c^{(t)}$的输出结果如下所示：

$$s_c^{(t)} = g_c^{(t)} \odot i_c^{(t)} + s_c^{(t-1)} \odot f_c^{(t)}$$

式中的乘法为点乘。

4） **遗忘门：**图中标识为$f_c^{(t)}$的单元，用于控制是否忘记内部状态单元中的数据，它提供了网络能够学习遗忘的能力。原始的 LSTM 单元中并没有提出遗忘门，内部状态是直接通过权值为 1 的边直接反馈的，后来的改进中提出了遗忘门，并且证明遗忘门在神经网络学习中非常重要。

5） **输出门：**图中标识为$o_c^{(t)}$的单元，与输入门单元的作用类似，也是控制内部状态单元是否输出的。

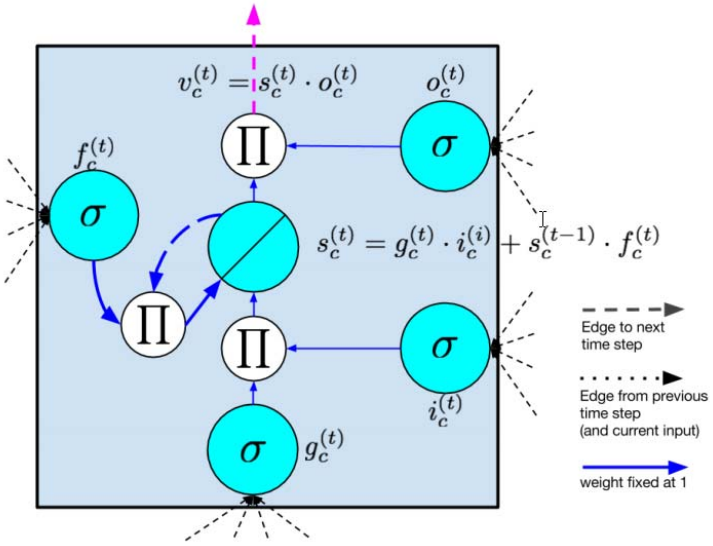


图 2-12 LSTM 基本单元结构图 [7]

注意：图中蓝色的边代表权值为 1 的边。

2.4.2 训练方法

下面的方程给出了 LSTM 的基本计算方法：

$$g^{(t)} = \phi(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g)$$
$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i)$$
$$f^{(t)} = \sigma(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f)$$
$$o^{(t)} = \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o)$$
$$s^{(t)} = \mathrm{g}^{(t)} \odot \mathrm{i}^{(t)} + \mathrm{s}^{(t-1)} \odot \mathrm{f}^{(t)}$$
$$h^{(t)} = \phi(s^{(t)}) \odot \mathrm{o}^{(t)}$$

其中 h(t)是 t 时刻的隐层输出。另外，如果需要计算没有遗忘门的简单 LSTM，只需要将 $f^{(t)}$置为 1 即可。

同样的，因为 LSTM 本身就是一种 RNN 网络，因此通过展开并应用 BPTT 算法来训练网络是完全可行的，无非就是表达式稍微复杂一些。

2.5 双向递归神经网络（Bidirectional recurrent neural networks，BRNN）

另外一种经常使用的 CNN 的结构是双向的递归神经网络 BRNN，这最早是 1997 年由 Schuster 和 Paliwal 提出来的[15]。可能会有点奇怪，RNN 递归网络是将 t-1 时刻的信息往 t 时刻传播，那双向呢，难道将 t+1 时刻的消息往 t 时刻传递吗？这不是当前能接受未来的消息吗？BRNN 确实是这么做的，它的结构如下图所示。可以看到 BRNN 中存在两条传递路径，一条是从 t-1 时刻的隐层向 t 时刻隐层再到 t+1 时刻隐层传递，另一条是从 t+1 时刻的隐层向 t 时刻再向 t-1 时刻传递的。

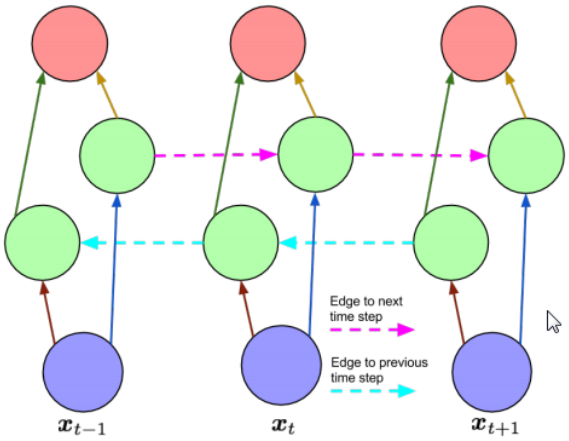


图 2-13 BRNN 神经网络的基本结构图

下列方程可以用来描述一个 BRNN 网络的结构：

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{\mathrm{hh}}h^{(t-1)} + b_h)$$
$$z^{(t)} = \sigma(W^{zx}x^{(t)} + W^{\mathrm{zz}}h^{(t+1)} + b_z)$$
$$\hat{y}^{(t)} = softmax(W^{yh}h^{(t)} + W^{\mathrm{yz}}z^{(t)} + b_y)$$

其中$h^{(t)}$，$z^{(t)}$分别是前向和后向传播的隐层输出。$\hat{y}^{(t)}$最终经过一个 softmax 激活函数输出。

BRNN 的一个限制就是它不能持续的运行下去，它需要在过去时刻和未来时候都有一个

固定的结束点或标志。BRNN 在语句预测方面表现是非常好的，因为它综合考虑了上下文之间的联系。

3. 论文系统框架

第二章花了很大的篇幅来解释神经网络和递归神经网络的一些基本知识，之所以这么详细讲述，因为这些知识全在第三章中使用到了，因此为了第三章不过于冗长，在第二章中详细介绍了基本的神经网络原理和数学推导。

第一章中提到，这篇论文主要完成了两方面的工作，两方面的工作又密切相关，因此本章将分成两个小节来分别阐述这两个框架。3.1 讲述作者 2014 年论文中提到的图像区域与单词描述映射算法的改进；3.2 讲述作者在 2015 年论文中提到的使用 RNN 网络来生成图像语句描述算法。

3.1 图像区域与单词描述映射算法改进

NIPS 2014 会议上的论文 ***Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping*** 主要介绍了一种将图像区域与图像描述语句中的词组关系进行双向映射和搜索的算法。通过将图片和语句描述输入作者实现的神经网络模型后，可以得到图片区域与文本描述短语的对应关系。

3.1.1 图像输入

作者采用的将输入图像转为空间矢量的方法是卷积神经网络，作者直接借鉴了[16]中提到的区域卷积神经网络(Region Convolutional Neural Network，RCNN)方法，该方法的示意图如下图所示。

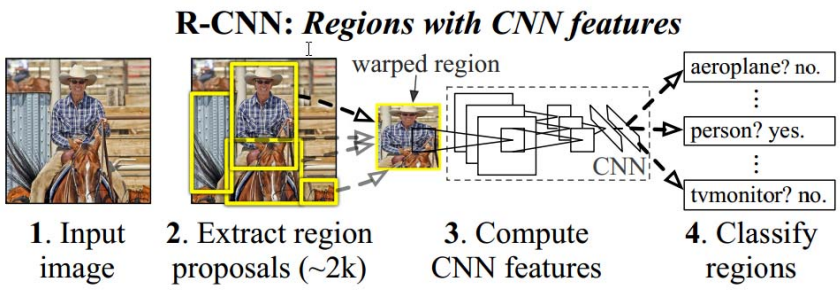


图 3-1 RCNN 系统结构框图

该 CNN 网络中，对于一张图片输入，首先从图片中提取出 2000 个区域，提取使用的算法是 selective search[17]，这里不做详细解释。对于提取到的每一个区域，使用 Caffe[18]工具生成一个 4096 维的向量，在输入到作者的 CNN 中进行识别训练。RCNN 是由 Berkeley 大学在 14 年提出来的，并提供源代码[19]。

回到本文，本文作者对 RCNN 的主要借鉴在于参考其**提取区域的方式**和**区域转换为向量**的方法。本文对于每幅图片，提取了 19 个区域（RCNN 中识别对分最高的），包括整幅图片，一共 20 个区域，然后使用 Caffe 生成一个20×4096维的向量，这就是本文图片向量的生成方式。对于每一个图像区域I_b，有：

$$\mathrm{v} = W_m[CNN_{\theta c}(\mathrm{I}_b) + b_m]$$

其中$CNN_{\theta c}$代表上面提到的 RCNN 网络，这个 RCNN 网络是在 ImageNet 图片数据集上已经训练和调优的，$CNN_{\theta c}(\mathrm{I}_b)$会产生一个 4096 维度的向量。然后$W_m$和$b_m$是本论文神经网络的权值和偏置参数，它们将这个 4096 维度的向量再次转换为一个 h 维空间向量，2014 年论文中给的是参考数是 1000，实际的代码中设置为 700，2015 年论文中给的参考值是 1000-1600。

3.1.2 语句输入

本文的语句输入采用的 BRNN 来进行转化的，这是和 2014 年的论文中采用的 Dependency Tree Relations 是不一样的。

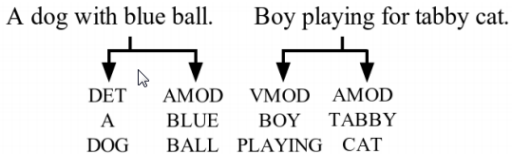
给定一个语句，如何生成向量？首先必须对句子中的单词进行编码，最简单的是 1-of-k 编码方式。假定我们存在一个单词库，假设其中有 5000 个已经排序的单词，那么对于每一个单词，我们都可以用一个 5000 维的向量来进行表示，记单词 word 在单词表中的索引为 index，那么单词 word 可以表示为 index 为 1，其他维数据都是 0 的一个 5000 维向量。

3.1.2.1 Dependency Tree Relations 方法

Dependency Tree Relations 的具体原理可以参考[20]。它的本质是对一个语句中的两个单词，以及它们之间的二元关系$(\mathrm{R}, \mathrm{W}_1, \mathrm{W}_2)$，计算向量表达如下：

$$\mathrm{s} = \mathrm{f}(\mathrm{W}_R \begin{bmatrix} W_e w_1 \\ W_e w_2 \end{bmatrix} + \mathrm{b}_R)$$

其中，假设单词库为 400,000，W_e是一个 dx400,000 维的向量(一般取 d=200)，w1 和 w2 是 400,000x1 维的向量。W_R和b_R是基于关系 R 的权值和偏置向量。f 函数是 ReLU 激活函数。W_R转换将 s 转换为一个 h 维空间的向量，取值与图像区域相同，为 700，即W_R是一个 700x400 维的矩阵。



最终对于每一个三元组，会生成一个向量。其中语句单词之间的连接关系在测试时是使用斯坦福大学的 CoreNLP parser 工具生成的，所有的单词连接关系都可以在[21]中找到。

3.1.2.2 BRNN 方法

而在 2015 年的 CVPR 中的改进是采用了 BRNN 来进行语句单词向量的生成。BRNN 的原理我们已经在第二章中进行了解释。

BRNN 网络的输入是一个语句，每个单词的编码也是 1-of-k 编码方式。BRNN 的结构图如下图 3-2 所示。

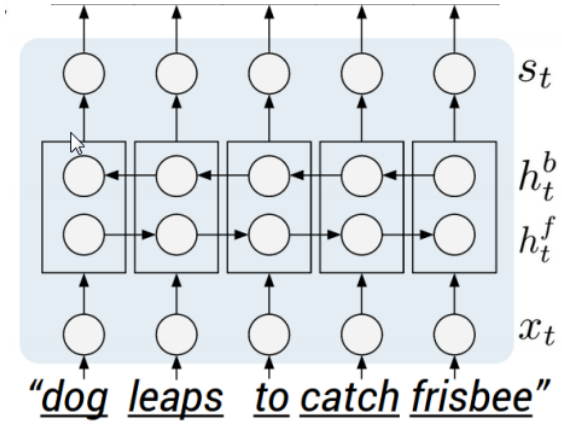


图 3-2 BRNN 单词向量生成方式网络图

该网络的基本公式推导如下：

$$x_t = W_w I_t$$
$$e_t = f(W_e x_t + b_e)$$
$$h_t^f = f(e_t + W_f h_{t-1}^f + b_f)$$
$$h_t^b = f(e_t + W_b h_{t+1}^f + b_b)$$
$$s_t = f(W_d(h_t^f + h_t^b) + b_d)$$

其中，I_t即为第 t 个单词的 1-of-k 编码输入，W_w是一个 300xN 的矩阵，N 是所有句子组成的单词库中所有单词的总数，每一个单词都通过 word2vec 工具[22]编码成一个 300 维的向量。该向量经过一个 BRNN 网络，BRNN 网络是由两个独立的前向和反向传播流组成的，最终输出一个与图片相同维数的向量s_t。向量s_t是第t 个单词及其上下文共同训练得到的。其中的函数 f 代表 ReLU 激活函数。

这里与 2014 年论文的**不同之处**在于，2014 年输入的向量代表的是语句中两个单词之间的关系，然后与图像区域进行匹配；而 2015 年采用的 BRNN 决定了了单词是逐个进行输入的，也就意味着输出单词也是逐个进行输出的。因此 2015 年文章中是将语句中的每个单词与图像中的区域进行对应。

3.1.3 图像区域与单词描述匹配算法

3.1.3.1 2014 年论文匹配算法

在图像向量和语句向量都已经获得前提下，两者进行映射的系统框图如下所示。左侧是 RCNN 实现的图像区域分配和向量生成，右侧是语句生成的关系向量。

对于一张图片，有一个区域向量{v}，v 的维数为700x20（700 为代码中给定的建议维数，20 代表有 20 个区域），对于该图片对应的一条语句，有一个单词关系向量{s}，s 的维数为 700xNum，Num 表示该语句的单词三元关系，一般等于(单词数-2)，两者进行内积运算得到$v_i^T s_j$。$v_i^T s_j$被定义为 i 区域和 j 短语之间的匹配得分。

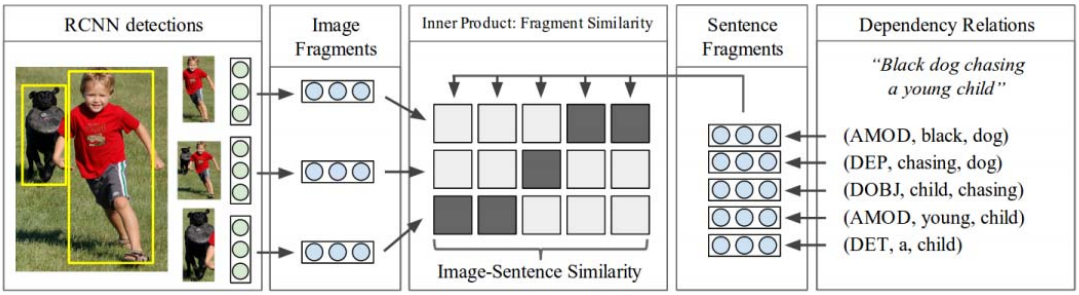


图 3-3 计算图像区域和语句描述匹配的框架

然后为了训练网络，需要有一个机制来计算匹配度的误差函数，2014 年的论文中给出的误差函数定义如下：

$$C(\theta) = C_F(\theta) + \beta C_G(\theta) + \alpha||\theta||_2^2$$

其中θ代表所有参数的集合$\theta = \{W_e, W_R, b_R, W_m, b_m, \theta_c\}$，$\alpha$和$\beta$都是在交叉验证时需要调节的超参数。这三项的含义分别为：第一项是段匹配的得分误差，即一张图片的某个区域和其描述语句中对应的单词关系三元组之间应该有比较高的得分；第二项是全局匹配的排序得分误差，即一张图片和其对应的语句之间的应该有比较高的得分；第三项是一个参数惩罚项。

1） 段匹配得分

误差定义如下：

$$C_0(\theta) = \sum_i \sum_j \max(0, 1 - y_{ij} v_i^T s_j)$$

$$y_{ij} = sign(v_i^T s_j)$$

$$C_F(\theta) = \min_{y_{ij}} C_0(\theta)，\quad 当 \sum_{i \in P_j} \frac{y_{ij} + 1}{2} \geq 1 \ \forall j$$

$$\forall i, j \ 当 \ m_v(i) \neq m_s(j) \ 并且 \ y_{ij} \in \{-1, 1\}时，\ y_{ij} = -1$$

这里的求和$C_0(\theta)$是对每批量的训练集中任一张图片的所有区域和任一语句的所有语句单词关系求和，因为这里采用的批量的随机梯度下降算法来进行训练，因此每次都是有 N 张图片和 N 个语句同时输入网络。

y_{ij}被定义为，如果 i 区域所属的图片和 j 单词关系所属的语句属于一个图像-语句对的话，则输出结果为+1；如果不匹配，输出为-1。其中$m_v(i)$和$m_s(j)$分别返回的是图像区域 i 所属图像在所有图像中的索引和单词关系 j 所属语句在该语句所有单词关系中索引，当它们不相同时，y_{ij}就等于-1 。

然后我们定义P_j为对于单词关系 j 使y_{ij}为正的所有 i 的集合，要求$\sum_{i \in P_j} \frac{y_{ij}+1}{2} \geq 1$也就是要求至少存在一个$y_{ij}$=1，比如说，在句子中提到了一个物体，但是在图上并不存在与这个物体匹配的，那么这个句子中的物体对应的单词关系就是无效的。 这个条件要求每个单词关系都至少存在一个与之匹配的图像区域。

直观的理解为：如果一个语句中出现了<R,a,dog>这样的关系（再假设该语句中只存在这一组关系），那么在进行匹配时我们只需要考虑所有图像中有狗出现的部分图像，其他的图像基本上是完全不相关，不需要训练它们之间的匹配关系。

很显然，训练的目的是为了让匹配区域得分高，那么当匹配时，$y_{ij} v_i^T s_j$这一项将远大于

1，因此误差为 0；当不匹配时，$y_{ij}{v_i}^T s_j$为负数，因此总的误差为$1-y_{ij}{v_i}^T s_j$。可以看下图所示的实例。图中一次批量输入两张图片，可以看到绿色的区域是匹配的区域，$y_{ij}=1$，因而得分为正；黄色和红色的区域是不匹配的区域，$y_{ij}=-1$，得分为负。

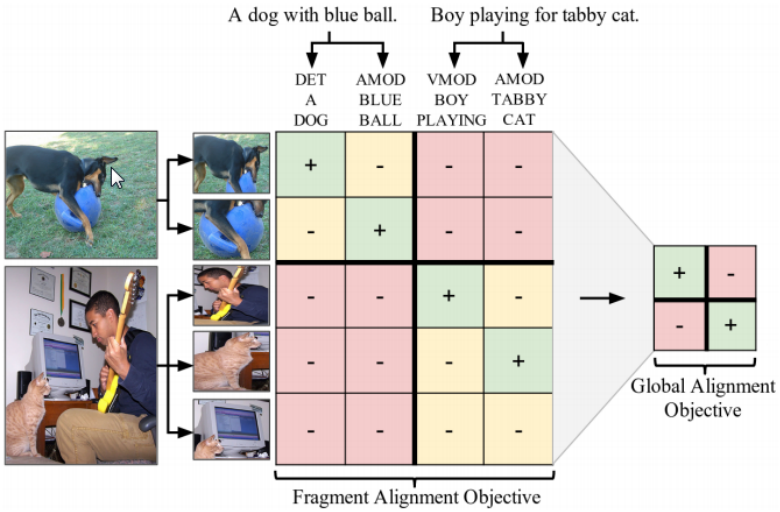


图 3-4 batch=2 时候的区域匹配实例

最后，在所有的图像语句对中，其中满足条件：**图像-语句对中至少存在一个相匹配的图像区域-单词关系对** 的计算出的所有误差$C_0(\theta)$中，取极小值，即为段匹配得分误差。

2） 全局匹配得分误差

首先定义图像 g 和语句 l 的匹配对的平均得分如下：

$$S_{kl}=\frac{1}{|g_k|(|g_l|+n)}\sum_{i\in g_k}\sum_{j\in g_l}\max(0,{v_i}^T s_j)$$

其中g_k是图像 k 所包含的图像区域数， g_l是语句 l 所包含的单词关系数。K 和 l 的范围都是从 1 到 N。n 是一个平滑因子，加入 n 的原因是考虑到一些短句子会有比较大的平均匹配得分，n 一般取 5。然后全局误差定义如下：

$$C_G(\theta)=\sum_k\left[\sum_l\max(0,S_{kl}-S_{kk}+\Delta)+\sum_l\max(0,S_{lk}-S_{kk}+\Delta)\right]$$

其中Δ是一个交叉验证的超参数。因为 k 和 l 是一一对应的，因此 k=l 时图像和语句是匹配的，此时应该得分最高，即S_{kl}和S_{lk}的值都应该小于S_{kk}。这个误差要想使其最小，要求S_{kk}的值至少比S_{kl}和S_{lk}的值大Δ。

3.1.3.2 2015 年论文匹配算法

对于 2015 年的论文，同样也是将图片区域向量和语句单词向量进行内积，如图 3-5 所示。

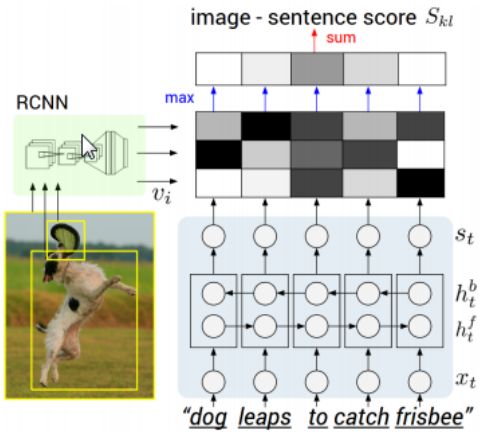


图 3-5 2015 年论文图像语句匹配模型框图

通过 RCNN 产生的图像区域向量v_i和通过 BRNN 产生的单词s_j向量进行内积$v_i{}^T s_j$，然后在 2015 年的论文中发现如果将匹配得分和误差函数定义为如下形式，可以大大简化模型，并且不需要引入额外的超参数。

$$S_{kl} = \sum_{j \in g_l} \max_{i \in g_k} (v_i{}^T s_j)$$

$$C(\theta) = \sum_k \left[\sum_l \max(0, S_{kl} - S_{kk} + 1) + \sum_l \max(0, S_{lk} - S_{kk} + 1) \right]$$

其中对于每个单词s_j，只需要选取内积$v_i{}^T s_j$最大，然后将一个语句的所有单词得分累加起来作为图像-语句对的得分即可。然后，对于误差$C(\theta)$，也不再使用平均匹配得分，直接用累加的得分，并且直接设置超参数Δ=1，在实际训练可以得到更好的结果。

3.1.4 输出模型

到现在为止，我们还没有讨论输出数据的形式。我们可以考虑，在训练期间，我们的输入是图像-语句对（包括图像内各区域与语句中单词描述的对应关系），在测试期间，我们的输入只有图像-语句对，因此我们的输出是得分向量S_{kl}，其维数为g_kxg_l，其中g_k是图像 k 所包含的图像区域数，g_l是语句 l 所包含的单词关系数，然后排序后进行输出。

3.2 生成图像描述语句算法

3.2.1 算法模型

前面提到，生成序列化学习结果的最好网络是 RNN，因为 RNN 网络可以加入上下文的影响作为下一次预测的依据。本文作者即采用了 RNN 网络用来生成一个图像的完整语句描述，整个系统的完整框图如下所示。

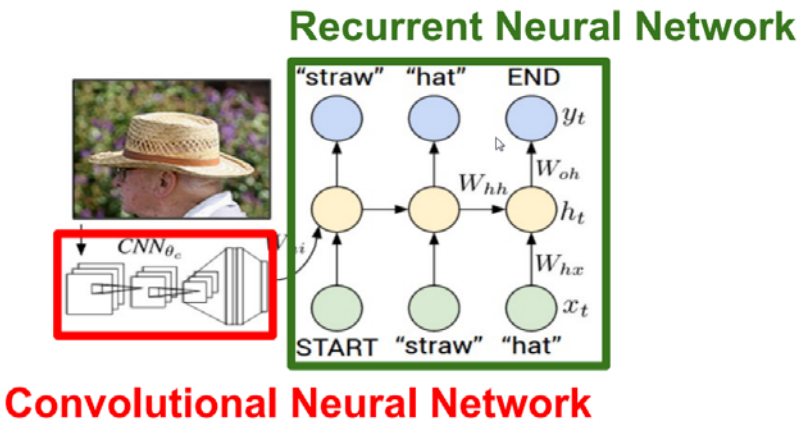


图 3-6 生成图像描述语句的算法框图

可以看出，对于图像的输入采用的是与上一节中图像区域和单词描述映射算法的完全相同的卷积神经网络。而对于语句，不需要采用上一节的 Dependency Tree Relations 或者 BRNN 模型，只需要直接输入图 3-6 描述的模型中即可。在训练时，输入图像和相匹配的语句，通过输出结果误差对网络权值进行更新即可；在测试时，直接输入图像，然后就会输出相应的预测语句。

3.2.2 模型详细结构

RCNN 将图像输入映射到一个 h 维空间中，对于每个单词的输入也通过前面提到的 word2vec 工具将其转换为 300x1 维的向量。

记输入语句向量$(x_1, \ldots, x_T)$，图像输入为I，则输出向量$(y_1, \ldots, \mathrm{y}_t)$可以通过以下迭代从$\mathrm{t}=1$到T生成：

$$\mathrm{b_v} = W_{hi}[CNN_{\theta c}(\mathrm{I})]$$

$$\mathrm{h}_t = f(W_{hx}x_t + W_{hh}h_{t-1} + \mathrm{b_h} + \Gamma(t=1)\odot \mathrm{b_v})$$

$$\Gamma(t=1) = \begin{cases} 1, & t=1 \\ 0, & \text{other} \end{cases}$$

$$y_t = softmax(W_{oh}h_t + b_o)$$

其中，W_{hi}，W_{hx}，W_{hh}，$\mathrm{b_h}$，W_{oh}，b_o都是需要学习的参数。而且上面方程的形式与我们前面讲过的 RNN 网络基本类似。有一点不同就在于$\mathrm{b_v}$，$CNN_{\theta c}(\mathrm{I})$产生的是一个 4096 维的向量，通过$W_{hi}$转变为 h 维的向量，然后 t=1 时刻，$\mathrm{b_v}$作为偏置向量进入 RNN 网络，并在其后的迭代过程中进行传递。

同时从图 3-6 可以发现，输入语句的第一个单词是一个奇怪的 START 符号，输出语句的最后一个单词是 END，这也是本文作者模型的一个关键点。因为我们需要的是输出一张图片的文本描述，因此长度是肯定有限的，因此加入了 START 为语句的开始，END 为语句的结束，首先输入 START 启动 RNN，在输出到 END 之后结束 RNN，如果是训练集，则进行参数更新，如果是测试，则输出结果。

然后再考虑一下输出向量y_t的维度。因为是基于当前已有的数据来预测下一个单词的输出，因此在本模型中会保存已经训练过的所有语句中出现的单词为一个单词库，记单词总数为 N，则向量y_t的维度为 N，y_t每一项为单词库中对应单词的可能概率，y_t是通过前面提到的 softmax 激活函数输出，因此只需要取最大概率对应单词作为输出即可。

3.2.3 训练方法

首先设置$\mathrm{h}_0 = 0$，$x_1 = START$符号对应的向量，在初始化权值和偏置参数后，计算得到

输出h_0和y_1，然后x_2是语句的第一个单词，输入后依次迭代下去，误差计算可以根据我们前面提到的 RNN 的误差反向传播方式进行迭代计算即可。

4. 论文实验结果

4.1 训练和测试数据集

论文采用的测试集是 Flickr8K[23]，Flickr30K[24] and MSCOCO[25]。这三个图片数据集中分别含有 8000,31000 和 123000 张图片。对于 Flickr8K 和 Flickr30K，作者使用 1000 张作为验证，1000 张作为测试，其余的作为训练图片；对于 MSCOCO，作者使用 5000 张图片作为验证，5000 张用于测试，其余的 113000 作为训练。

对于每张图片的语句输入，作者通过亚马逊的 Amazon Mechanical Turk 发布任务，雇佣人工来实现为每张图片添加 5 个不同的描述语句。

在生成文本描述阶段，需要对训练数据中的所有语句中出现的单词生成一个单词库，作者采用了一个单词出现此处必须大于等于 5 的过滤法则，从而去掉比较偏很少使用的单词，最终三个数据集中的单词书分别为 2538,7414 和 8791。

4.2 测试指标

4.2.1 BLEU 得分

随着统计机器翻译方法的兴起，一个有趣的问题摆在人们的面前：如何评价一个机器翻译方法的好坏？

最开始人们选用的是人工评测的方法，那就是一个翻译结果得到后，找一批专家来给每个句子翻译结果打分，然后统计均分。这种方法存在问题。首先，如果两个翻译结果给两批专家打分，如何保证打分的尺度一致？退一步讲就是一批专家给两个翻译结果打分，也无法保证打分尺度一致。其次，人工测评太耗费人力物力，并且无法再利用。在这样的背景下，需要有一种能够自动的客观的评价方法来代替人工评价。

一个用来衡量自然语言结构化输出的参考指标是 **BLEU 分数**。BLEU 分数是在 2002 年由 Papineni 提出来的[26]，并且用在机器翻译领域。BLEU 方法的工作原理为：比较候选译文（candidate）与参考译文（reference）中相同的片段数量。即用翻译结果中连续出现的 N 元组（n 个单词/字或标点）与参考译文中出现的 N 元组进行比较，计算完全匹配的 N 元组的个数与翻译结果中 N 元组的总个数的比例。这是一种类似准确率的计算方法，它允许一个原文有多个参考译文。

假设一个句子 S 可以被切分为 1 个词序列，记为$S = \omega_1\omega_2 \dots \omega_k$，其中$\omega_i$为句子中的一个词，令$\omega_j^i = \omega_j\omega_{j+1} \dots \omega_i$，则$\mathrm{S} = \omega_i^k$，，称词串$\omega_j^{j+\mathrm{n}-1}$ 为一个 n 元组，根据 n 取值的不同，可以得到不同的 n 元组。

BLEU 的计算公式如下：

$$\mathrm{B} = \begin{cases} 1, & if\ c > r \\ e^{(1-r/c)}, & if\ c \le r \end{cases}$$

$$\mathrm{BLEU} = \mathrm{B} \cdot \exp(\frac{1}{N}\sum_{n=1}^{N} logp_n)$$

其中 n 代表 n-元组，n 的范围从 1 到 N，一般取 N=4。B 是一个惩罚因子，因为 BLEU 分数可以通过连续的短语翻译来提高，因此加入 B 后，当候选译文句子长度不小于参照译文时，B 取值为 1，不影响句子的最终 BLEU 值；当候选译文句子过短时，将被判罚，引入句子过短的判罚因子后，修正了 BLEU 得分.。c 是候选翻译的平均长度，r 是参考翻译的平均长度。p_n是 n 元组的准确度，它等于候选翻译中 n 元组在参考翻译中出现的次数除以该 n 元组在候选翻译中出现的次数。

对于本文的生成图片描述，与机器翻译类似，可以使用 BLEU 作为一个衡量标准。

4.2.2 召回率 Recall

在图像区域和单词描述映射模型中，可以看成是图像和文本的双向检索。事实也确实如此，作者实现了在线的图像区域和文本描述检索，可以参考 [4]。

因此可以使用信息检索领域的基本指标召回率来进行衡量。

$$\text{召回率 Recall} = \frac{\text{系统检索到的相关文件}}{\text{系统所有相关的文件总数}}$$

Recall 意味着在所有相关文件中识别到相关文件的概率。

在本论文中，作者实现了图像区域和文本描述的双向检索，给定一个图像或者文本，检索到的文本或者图像按S_{kl}分数进行排序，Recall@K 就意味着在分数较高的 K 个结果中找到相关文本或者图像的比例。这就是 Recall@K 的含义，显然 Recall@K 越高越好。

4.2.3 中位排序 median rank

中位排序指的是在测试集上检索到的所有语句对应图片的次序的中位数。

4.3 测试结果

4.3.1 图像区域-文本描述双向映射模型测试结果

图像区域-文本描述双向检索模型采用的测试指标为 Recall@K 和 Median Rank。测试结果如图 4-1 所示。

Model	Image Annotation R@1	R@5	R@10	Med r	Image Search R@1	R@5	R@10	Med r
Flickr30K								
SDT-RNN	9.6	29.8	41.1	16	8.9	29.8	41.1	16
Kiros et al.	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Mao et al.	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Donahue et al.	17.5	40.3	50.8	9	-	-	-	-
DeFrag Karpathy et al.	14.2	37.7	51.3	10	10.2	30.8	44.2	14
Our implementation of DeFrag	19.2	44.5	58.0	6.0	12.9	35.4	47.5	10.8
Our model: DepTree edges	20.0	46.6	59.4	5.4	15.0	36.5	48.2	10.4
Our model: BRNN	**22.2**	**48.2**	**61.4**	**4.8**	**15.2**	**37.7**	**50.5**	**9.2**
Vinyals et al. (more powerful CNN)	23	-	63	5	17	-	57	8
MSCOCO								
Our model: 1K test images	38.4	69.9	80.5	1.0	27.4	60.2	74.8	3.0
Our model: 5K test images	16.5	39.2	52.0	9.0	10.7	29.6	42.2	14.0

图 4-1 图像区域-文本描述双向检索模型测试结果

其中与之比较的模型从上到下依次参考 [20]、[27]、[28]、[29]、[3]、[3]、[1]、[1]、[2]。

从表中可以看到，作者实现的模型比以往所有的模型在各个指标上的得分都要好，唯一不能

匹敌的就是今年 Google 工程师在 CVPR 2015 上发表的相同成果，因为 Google 采用的是 CNN+LSTM 模型，首先他们的 CNN 模型就是一个 27 层的网络模型[30]，非常强大，然后 LSTM 模型在我们前面的介绍中知道是比 RNN 在序列化记忆方面表现要好，因此 Google 模型表现更好是合情合理的。

该模型的验证测试结果如图 4-2 所示。图中用不同颜色标出了得分比较高的几个区域与对应单词描述的结果。

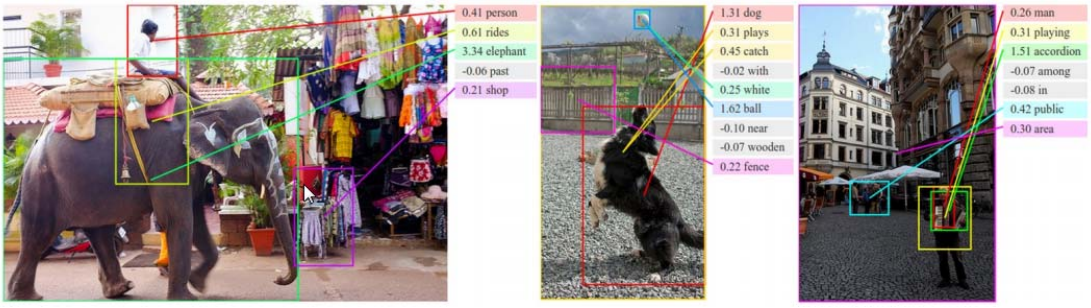


图 4-2 图像区域-文本描述映射模型测试图片结果

该模型的测试结果显示：

1） 该模型的表现比以往的几乎所有模型都有更好的表现，Google CVPR 2015 的模型除外；

2） 与 2014 年的结果相比，通过改进的更简洁的匹配得分和误差函数，该模型有了更好的表现；

3） BRNN 比 Dependency tree relations 模型表现要好；

4.3.2 生成图像描述语句模型测试结果

生成图像描述语句模型采用 BLEU 作为测试指标，其测试结果如下图所示：

	Flickr8K				Flickr30K				MSCOCO 2014					
Model	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	METEOR	CIDEr
Nearest Neighbor	—	—	—	—	—	—	—	—	48.0	28.1	16.6	10.0	15.7	38.3
Mao et al.	58	28	23	—	55	24	20	—	—	—	—	—	—	—
Google NIC	63	41	27	—	66.3	42.3	27.7	18.3	66.6	46.1	32.9	24.6	—	—
LRCN	—	—	—	—	58.8	39.1	25.1	16.5	62.8	44.2	30.4	—	—	—
MS Research	—	—	—	—	—	—	—	—	—	—	—	21.1	20.7	—
Chen and Zitnick	—	—	—	14.1	—	—	—	12.6	—	—	—	19.0	20.4	—
Our model	57.9	38.3	24.5	16.0	57.3	36.9	24.0	15.7	62.5	45.0	32.1	23.0	19.5	66.0

图 4-3 生成图像描述语句模型测试结果

其中与之比较的模型从上到下依次参考 作者实现的一个基准模型、[28]、[2]、[29]、[31]、[32]、[1]。B-1 到 B-4 依次代表一元组到四元组的 BLEU 分数，显然分数越高表现越好。从上图可以看出，作者的模型除了与 Google 的模型相比稍有差距外，比其他模型在大部分方面都要好。

下图是实际的预测图，可以看到作者的模型给了一些比较好的描述，但仍存在一些问题。图 4-4 中前两幅图像的描述都没有问题，但第三幅和第四幅图像显然描述错了，作者的模型还没有完全学习到 baby 的特征，会把 baby 错认为是 girl。同时，作者也指出对于第一幅图片，作者的训练集中并没有出现过“man in black shirt is playing a guitar”这句完整的话，但是“ man in black shirt”出现了 20 次，“is paying guitar”出现了 60 次。显然这次的正确预测说明作者的模型确实学习到了“man in black shirt”和“is paying guitar”这样的两个图片区域的特

征。

图 4-4 生成图像描述语句模型的实际预测结果图

同理，作者的模型不仅能够预测整幅图片，对于图片中的不同区域，使用 RNN 模型同样能够正确预测，如下图所示。

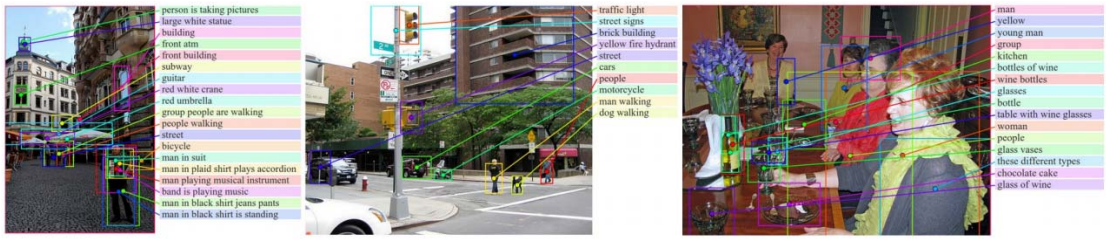


图 4-5 生成图像描述语句模型的区域描述预测结果图

5. 结论和展望

5.1 结论

综上所述，作者在本篇论文中一共完成了两方面的工作：

1） 对其在 NIPS 2014 会议上发表的论文 Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping 中的算法进行了改进。NIPS 2014 会议上的这篇论文主要介绍了一种将图像区域与图像描述语句中的词组关系进行双向映射和搜索的算法。通过将图片和语句描述输入作者实现的神经网络模型后，图片区域与文本描述短语的对应关系。而 CVPR 2015 会议上的这篇论文对其中语句矢量模型的生成方式进行了改进，使用双向的递归神经网络（BRNN，Bidirectional recurrent neural network）来取代原有的 Dependency Tree Relations 方法，并对图像区域-描述短语之间的匹配得分函数和误差函数进行了简化，从而得到了更好的测试结果。

2） 实现了图像区域（或者完整图像）的语句描述短语（或者整个语句）的生成。即给定一副图像或者图像的一个区域，能够生成该图像的完整语句描述或者图像区域的短语描述。如下图所示。作者采用的模型架构为 RCNN+BRNN，其中 RCNN，用于将图片或者图片区域转化为 h 维空间向量，同时使用 word2vector 将图片对应语句中的每一个单词转化为 h 维的空间矢量，然后两者输入 RNN(Recurrent neural network)，这个递归神经网络会输出语句序列。

作者的工作在最后的测试中显示表现是相当优秀的，但是也存在一些局限：

1） 只能输入分辨率固定的图片，不能进行动态的自适应；

2） RNN 只在初始时刻的偏置中接收图片向量，这使得 RNN 不能产生更加具有表现力的描

述；

3） 作者的模型是完全分隔的两块 CNN+RNN，如果能够直接从图片输出到文本描述输出，或许会有更加好的表现，但是实现将比较困难。

5.2 展望

作者工作的展望之一就是使用 Google 模型中使用 LSTM 网络来取代现有的 RNN。LSTM 模型的基本单元结构已经在第二章详细解释了。Google 模型的 LSTM 网络结构图如下图 5-1 所示。左侧是其使用的 27 层 CNN 网络，右侧是 LSTM 网络，其结构与本文作者描述的 RNN 十分相似，唯一的不同的就是每一个 LSTM 单元的不同，因此本文工作中可以将 RNN 替换成 LSTM 做进一步的测试。

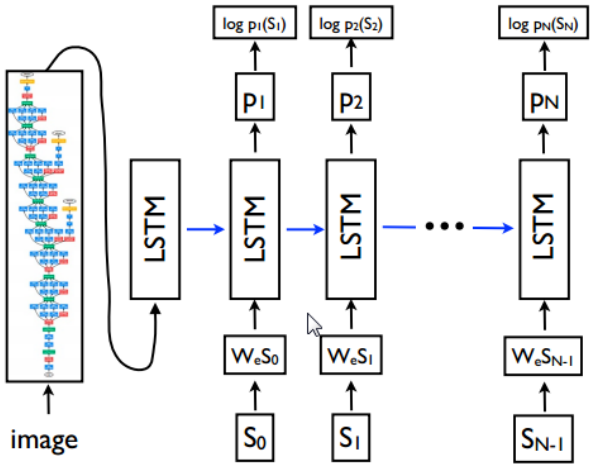


图 5-1 Google 公司 CVPR2015 论文模型框图

同时本论文的基本代码是使用 python 代码写的，在使用 CPU 运行的情况下，训练速度非常慢，我在实验室的服务器上需要 90s 才能训练完一张图片，因此速度有必要进行改进，目前可能的方案是使用 cpython，转移到 Torch，或者改为分布式运行。这也是一个可以改进的方向。

5.3 感受

由于专业不相关，因此对机器学习和神经网络方面几乎是陌生的，但是在 CVPR 上找到这篇论文和 Google 的那篇论文后，对其实现的功能相当好奇，因此选了这个题目。因为论文中提到的基本是神经网络的内容，所以一直在看 RNN 和 LSTM 方面的文章，而国内这方面的中文资料比较少，因此对于英文不是很好的我来说进展相当缓慢，最终在看了很多文章之后，对论文中的一切都豁然开朗的时候还是很开心的。第二章中把所有的基本知识全部罗列出来，不仅仅是因为在后面的章节中需要用到，更是对自己近阶段的一个总结，所以也要求自己把所有的公式都自己推导一遍，只有自己推导过后，才会真正理解。同时对 LSTM 的理解还不是很深刻，需要继续学习。

参考文献

[1] Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions[J]. arXiv preprint arXiv:1412.2306, 2014.
[2] Vinyals O, Toshev A, Bengio S, et al. Show and Tell: A Neural Image Caption Generator[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3156-3164.
[3] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. arXiv
preprint arXiv:1406.5679, 2014.
[4] DeFrag online web demo, http://cs.stanford.edu/people/karpathy/defragvis/, 2015. Accessed: 2015-10-18.
[5] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[J]. arXiv preprint arXiv:1502.03044, 2015.
[6] Mitchell TM. 机器学习[M]. 曾华军, 张银奎, 等译. 北京: 机械工业出版社, 2003: 60-90.
[7] Lipton Z C. A Critical Review of Recurrent Neural Networks for Sequence Learning[J]. arXiv preprint arXiv:1506.00019, 2015.
[8] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 807–814, 2010.
[9] Backpropagation, http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html, 2015. Accessed: 2015-11-01.
[10] Deep Learning（深度学习）学习笔记整理系列之（七）, http://blog.csdn.net/zouxy09/article/details/8781543 , 2015. Accessed: 2015-10-26.
[11] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015. Accessed: 2015-10-28.
[12] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, 2009.
[13] Guo J. BackPropagation Through Time[J]. 2013.
[14] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[15] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 45(11):2673–2681, 1997.
[16] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014: 580-587.
[17] Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2): 154-171.
[18] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the ACM International Conference on Multimedia. ACM, 2014: 675-678.
[19] RCNN. http://www.cs.berkeley.edu/~rbg/rcnn, 2014.
[20] Socher R, Karpathy A, Le Q V, et al. Grounded compositional semantics for finding and describing images with sentences[J]. Transactions of the Association for Computational Linguistics, 2014, 2: 207-218.
[21] De Marneffe M C, Manning C D. Stanford typed dependencies manual[R]. Technical report, Stanford University, 2008.
[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, 2013.
[23] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: data,

models and evaluation metrics. Journal of Artificial Intelligence Research, 2013.
[24] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. TACL, 2014.
[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. arXiv preprint arXiv:1405.0312,2014.
[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics, 2002.
[27] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539, 2014.
[28] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. arXiv preprint arXiv:1410.1090, 2014.
[29] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach,S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. arXiv preprint arXiv:1411.4389, 2014.
[30] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[J]. arXiv preprint arXiv:1409.4842, 2014.
[31] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. arXiv preprint arXiv:1411.4952, 2014.
[32] X. Chen and C. L. Zitnick. Learning a recurrent visual representation for image caption generation. CoRR, abs/1411.5654, 2014.