用递归神经网络进行序列预测的预定取样

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, Noam Shazeer
美国加州山景城谷歌研
究院
{bengio,vinyals,ndjaitly,noam}@google.com

摘要

递归神经网络可以被训练来产生给定输入的标记序列，最近在机器翻译和图像字幕方面的成果就是一个例子。目前训练它们的方法包括在给定当前（循环）状态和前一个标记的情况下，使序列中每个标记的可能性最大化。在推理中，未知的前一个标记被模型本身生成的标记所取代。训练和推理之间的这种差异会产生错误，这些错误会沿着生成的序列迅速积累起来。我们提出了一个课程学习策略，将训练过程从使用真实的前一个标记的完全指导性方案，转变为主要使用生成的标记的较少指导性方案。在几个序列预测任务上的实验表明，这种方法产生了明显的改进。此外，我们在2015年MSCOCO图像字幕挑战赛的获奖作品中也成功地使用了这种方法。

1 简介

循环神经网络可以用来处理序列，无论是输入、输出还是两者。虽然众所周知，当数据中存在长期依赖关系时，它们很难训练[1]，但一些版本如长短时记忆（LSTM）[2]更适合于此。事实上，它们最近在一些序列预测问题上表现出令人印象深刻的性能，包括机器翻译[3]、上下文解析[4]、图像说明[5]甚至是视频描述[6]。

在本文中，我们考虑了试图生成一个大小可变的标记序列的问题集，如机器翻译问题，其目标是将一个给定的句子从源语言翻译成目标语言。我们还考虑了输入不一定是一个序列的问题，如图像说明问题，其目标是生成一个给定图像的文本描述。

在这两种情况下，递归神经网络（或其变种，如LSTM）通常被训练成在输入的情况下生成目标符号序列的可能性最大化。在实践中，这是通过给定模型的当前状态（总结了输入和过去的输出令牌）和之前的目标令牌，最大化每个目标令牌的可能性来实现的，这有助于模型在目标令牌上学习一种语言模型。然而，在推理过程中，*以前*真正的目标标记是不可用的，因此被模型本身产生的标记所取代，产生了模型在训练和推理中的使用方式的差异。这种差异可以通过使用保持几个生成的目标序列的波束搜索启发式来缓解，但是对于像递归神经网络这样的连续状态空间模型，没有动态编程方法，所以即使使用波束搜索，

考虑的有效序列数量仍然很小。

主要的问题是，在序列生成过程的早期所犯的错误被作为输入输入到模型中，并可能被迅速放大，因为模型可能处于它在训练时从未见过的状态空间的一部分。

在这里，我们提出了一种*课程学习*方法[7]，以缓和地弥补使用递归神经网络进行序列预测任务的训练和推理之间的差距。我们建议改变训练过程，以逐渐迫使模型处理自己的错误，就像它在推理过程中必须处理的那样。这样做，模型在训练过程中会有更多的探索，因此在推理过程中纠正自己的错误更加稳健，因为它在训练中已经学会了这样做。我们将通过实验表明，这种方法在几个序列预测任务上产生了更好的性能。

本文的组织结构如下：在第2节中，我们提出了用递归神经网络更好地训练序列预测任务的方法；接下来是第3节，它与一些相关的方法进行了联系。然后，我们在第4节中介绍了一些实验结果，并在第5节中得出结论。

2 建议的方法

我们考虑的是有监督的任务，其中训练集是以N个输入/输出对的方式给出的$\{X^i, Y^i\}_{i=1}^N$，其中X^i是输入，可以是静态的（如图像）或动态的（如序列），而目标输出Y^i是一个序列$y^i_1, y^i_2, ..., y^i_{T_i}$的数量不等的标记。属于一个固定的已知字典。

2.1 模型

给定一个单一的输入/输出对（X，Y），对数概率P（Y /X）可以计算为：

$$\begin{aligned}\log P(Y/X) &= \log P(y^T_1/X) \\ &= \sum_{t=1}^{T} \log P(y_t/y^{t-1}_1, X)\end{aligned} \tag{1}$$

其中Y是一个长度为T的序列，由tokens $y_1, y_2, ..., y_T$。上式中的后一项是由参数为ϑ的递归神经网络估计的，它引入了一个状态向量h_t，它是前一个状态h_{t-1}和前一个输出标记y_{t-1}的函数，即。

$$\log P(y_t/y^{t-1}, X; \vartheta) = \log P(y_t/h_t; \vartheta) \tag{2}$$

其中，h_t是由一个循环神经网络计算的，如下所示：

$$h_t = \begin{cases} f(X; \vartheta) & \text{如果} t = 1, \\ f(h_{t-1}, y_{t-1}; \vartheta) & \text{否则} \end{cases} \tag{3}$$

。

$P(y_t/h_t; \theta)$通常被实现为状态向量h 的线性投影，即输出词典中的每个符号都有一个分数向量。[1]状态向量h_t到一个分数向量，输出字典的每个符号都有一个分数，然后进行softmax转换，以确保分数被正确归一化（正数和为1）。$f(h, y)$通常是一个非线性函数，将以前的状态和以前的输出结合起来，以产生当前的状态。

这意味着该模型的重点是学习如何在模型的当前状态和前一个标记的情况下输出下一个标记。因此，该模型以最一般的形式表示序列的概率分布--与条件随机场[8]和其他模型不同，这些模型假定在不同的时间步骤中，鉴于潜在的变量状态，输出之间是独立的。该模型的容量只受限于递归层和前馈层的表示能力。LSTM具有学习长距离结构的能力，特别适合这项任务，并使其有可能学习序列上的丰富分布。

为了学习可变长度的序列，一个特殊的标记<EOS>，表示一个序列的结束，被添加到字典和模型中。在训练期间，*<EOS>*被连接到每个序列的结尾。在推理过程中，模型产生标记，直到产生<EOS>。

[1]尽管人们也可以使用多层次的非线性投影。

2.2 培训

训练循环神经网络来解决这样的任务，通常是通过使用小批量随机梯度下降来寻找一组参数ϑ^{*}，该参数在所有训练对（X^{i}，Y^{i}）的输入数据X^{i}的情况下，产生正确目标序列Y^{i}的对数可能性最大化:

$$\vartheta^{*} = \arg\max_{\theta} \sum_{(X^{i},Y^{i})} \log P(Y^{i} \mid X^{i};\vartheta) \text{。} \tag{4}$$

2.3 推论

在推理过程中，该模型可以通过每次生成一个标记，并将时间推进一步来生成完整的序列y^{T}，给定X。当一个<*EOS*>标记生成时，它标志着序列的结束。在这个过程中，在时间t，模型需要上一个时间步骤的输出标记y_{t-1}作为输入，以产生y_{t}。由于我们无法获得真正的前一个标记，我们可以选择我们模型中最有可能的一个，或者根据它进行采样。

由于序列数量的组合增长，搜索具有给定X的最高概率的序列Y过于昂贵。相反，我们使用一个波束搜索程序来生成k个"最佳 "序列。我们通过维护一个由m个最佳候选序列组成的堆来做到这一点。在每个时间步骤中，新的候选序列是通过将每个候选序列扩展一个标记并将其添加到堆中而产生的。在该步骤结束时，堆被重新修剪，只保留m个候选序列。当没有新的序列加入时，波束搜索被截断，并返回k个最佳序列。

虽然波束搜索通常用于基于离散状态的模型，如可以使用动态编程的隐马尔科夫模型，但对于基于连续状态的模型，如递归神经网络，则很难有效使用，因为没有办法在连续空间中对跟随的状态路径进行分解，因此在波束搜索解码过程中可以保留的实际候选数非常少。

在所有这些情况下，如果在时间t-1采取了一个错误的决定，模型可能处于状态空间的一部分，与从训练分布中访问的那些状态空间有很大的不同，而且它不知道该怎么做。更糟糕的是，这很容易导致累积性的错误决定--这是顺序吉布斯抽样法中的一个典型问题，未来的样本对过去没有影响。

2.4 用预定的抽样来弥补差距

在预测标记y_{t}时，序列预测任务的训练和推理的主要区别在于我们是使用真实的前一个标记y_{t-1}还是使用来自模型本身的估计值$\hat{y}_{t-1}$。

我们在此提出一种抽样机制，在训练过程中随机决定我们是使用y_{t-1}还是$\hat{y}_{t-1}$。假设我们使用基于迷你批的随机梯度下降方法，对于训练算法的i^{th}迷你批中预测y_{t} ⍰ Y的每一个标记，我们建议掷硬币，使用概率为ϵ_{i}的先前真实标记，或概率为（1 - ϵ_{i}）的来自模型本身的估计。[2]模型的估计值可以通过根据$P(y_{t-1} \mid h_{t-1})$建模的概率分布抽样得到，也可以取为$\arg\max_{s} P(y_{t-1} = s \mid h_{t-1})$。这个过程如图1所示。

当$\epsilon_{i} = 1$时，模型的训练与之前完全一样，而当$\epsilon_{i} = 0$时，模型的训练与推理的设定相同。我们在此提出一个*课程*学习策略，从一个到另一个：直观地说，在训练开始时，从模型中抽样会产生一个随机的标记，因为模型没有被很好地训练，这可能导致非常缓慢的收敛，所以更经常地选择以前的真实标记应该有帮助；另一方面，在训练结束时，$_{i}$应倾向于从模型中更频繁地取样，因为这对应于真实的推理情况，人们期望模型已经足够好地处理它并取到合理的令牌。

[2]请注意，在实验中，我们对每一个标记都抛掷硬币。我们也尝试在每个序列中抛一次硬币，但结果要差得多，很可能是因为连续的错误在第一轮的训练中被放大了。

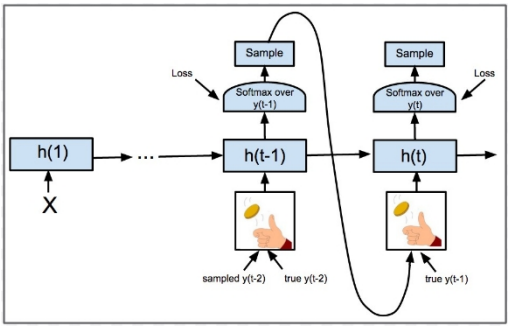
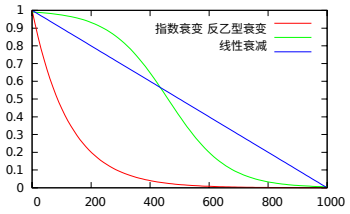


图1：计划抽样方法的说明，在每一个时间步骤中抛出一枚硬币，以决定是使用以前的真实标记还是从模型本身抽出的标记。



图： 衰减时间表的例子

因此，我们建议使用一个时间表来减少ϵ_i，作为i本身的函数，其方式类似于大多数现代随机梯度下降方法中用于降低学习率。在图2中可以看到这种时间表的例子，如下所示：

- 线性衰减：$\epsilon_i = \max(\epsilon, k - ci)$ 其中，$0 \leq \epsilon < 1$是给予模型的最小真理量，k和c提供衰减的偏移量和斜率，这取决于预期收敛速度。
- 指数衰减：$\epsilon_i = k^i$ 其中$k<1$是一个常数，取决于预期收敛速度。
- 反西格玛衰减：$\epsilon_i = k/(k + \exp(i/k))$ 其中$k \geq 1$取决于预期的收敛速度。

我们把我们的方法称为"*预定采样*"。请注意，当我们在训练时从模型本身抽出前一个标记y_{t-1}，我们可以通过该决策来反向传播$t \to T$时间的损失梯度。这一点在本文描述的实验中没有做到，而是留待以后的工作。

3 相关工作

文献中已经注意到训练和推理分布之间的差异，特别是对于控制和强化学习任务。

SEARN[9]的提出是为了解决这样的问题：当每个例子是由一连串的决定组成时，监督训练的例子可能与实际测试的例子不同，就像在一个复杂的环境中行动，模型在连续的决定过程中早期的几个错误可能会复合，产生一个非常糟糕的整体性能。他们提出的方法涉及一个元算法，在每个元迭代中，根据当前的*策略*（基本上是每种情况下的预期决策）训练一个新的模型，将其应用于测试集，并修改下一个迭代策略，以考虑到之前的决策和错误。因此，新的策略是以前的策略和模型的实际行为的结合。

与SEARN和相关的想法[10, 11]相比，我们提出的方法是完全在线的：一个单一的模型被训练，*政策*在训练过程中慢慢演变，而不是一个批处理的方法，这使得它的训练速度更快。[3] 此外，SEARN是在强化学习的背景下提出的，而我们考虑的是使用随机梯度下降法对总体目标进行训练的监督学习环境。

其他方法从排序的角度考虑问题，特别是对于目标输出是树的解析任务[12]。在这种情况下，作者提议在训练和推理过程中都使用波束搜索，这样两个阶段就能保持一致。训练波束被用来寻找

[3]事实上，在我们本文报告的实验中，我们提出的方法在训练上并不比基线慢（也不快）。

模型的当前最佳估计值，使用排名损失将其与指导性解决方案（真相）进行比较。不幸的是，当使用像递归神经网络这样的模型（现在是许多序列任务中最先进的技术）时，这是不可行的，因为状态序列不容易被因子化（因为它是一个多维的连续状态），因此梁式搜索在训练时间（以及推理时间，事实上）很难有效使用。

最后，[13]提出了一种用于解析问题的在线算法，该算法通过使用一个考虑到模型决策的*动态神谕*来适应目标。训练的模型是一个感知器，因此不像递归神经网络那样基于状态，而且在训练过程中选择真理的概率是固定的。

4 实验

我们在本节中描述了三个不同任务的实验，以表明预定采样在不同的环境中是有帮助的。我们报告了关于图像说明、选区解析和语音识别的结果。

4.1 图片说明

在过去的一年中，图像说明引起了很多人的注意。这项任务可以被表述为将图像映射到用某种自然语言描述其内容的文字序列上，大多数提议的方法采用某种形式的递归网络结构和简单的解码方案[5, 6, 14, 15, 16]。一个值得注意的例外是[17]中提出的系统，它没有直接优化给定图像的标题的对数可能性，而是提出了一个流水线方法。

由于一张图片可以有很多有效的标题，这项工作的评估仍然是一个开放的问题。一些人已经尝试设计与人类评价正相关的指标[18]，MSCOCO团队已经发布了一套通用的工具[19]。

我们使用来自[19]的MSCOCO数据集来训练我们的模型。我们在7500张图片上进行了训练，并在另外的5千张图片的开发集上报告了结果。语料库中的每张图片都有5个不同的标题，因此训练过程中随机挑选一个，创建一个小型的例子批，并优化（4）中定义的目标函数。图像被一个预训练的卷积神经网络（没有最后的分类层）预处理，类似于[20]中描述的那个，所得到的图像嵌入被当作是模型开始生成语言的第一个词。产生词语的递归神经网络是一个LSTM，有一层512个隐藏单元，输入的词语由大小为512的嵌入向量表示。词典中的单词数量为8857个。我们对ϵ，采用了反西格玛衰变时间表的预定采样方法。

表1显示了开发集上各种指标的结果。这些指标中的每一个都是估计所获得的词的序列和目标词的序列之间的重叠的变体。由于每张图片有5个目标标题，所以总是选择最好的结果。据我们所知，基线结果与目前该任务的最先进水平一致（略好）。虽然dropout在对数似然方面有帮助（如预期但未显示），但它对实际指标有负面影响。另一方面，预定抽样成功地训练了一个对训练和推理不匹配造成的失败更有弹性的模型，根据所有的指标，这可能产生更高质量的字幕。集合模型也产生了更好的性能，无论是基线还是计划抽样方法。同样有趣的是，一个总是从自己身上取样（因此在一个类似于推理的系统中）的训练模型，在表格中被称为 "*总是取样*"，产生了非常差的性能，正如预期的那样，因为模型在这种情况下很难学习任务。我们还用计划抽样训练了一个模型，但不是从模型中抽样，而是从一个均匀分布中抽样，以验证建立在当前模型上是很重要的，而且性能提升不仅仅是一种简单的正则化形式。我们把这称为*统一调度采样*，结果比基线好，但不如我们提出的方法好。我们还试验了每个序列翻转一次硬币，而不是每个标记翻转一次，但是结果和*总是抽样的*方法一样差。

表1：在MSCOCO开发集上用于图像字幕任务的各种指标（越高越好）。

方法与度量	BLEU-4	梅泰尔	CIDER
基准线	28.8	24.2	89.5
辍学的基线	28.1	23.9	87.0
始终保持取样	11.2	15.7	49.7
预定的取样	**30.6**	**24.3**	**92.1**
统一的计划抽样	29.2	24.2	90.9
10人的基线组合	30.7	25.1	95.7
预定的5个取样组合	**32.3**	**25.4**	**98.7**

值得注意的是，我们用我们的预定抽样方法参加了2015年MSCOCO图像字幕挑战赛[21]，并在最后的排行榜上排名第一。

4.2 选区解析

另一个与*任意到序列*范式不太明显的联系是构词法解析。最近的工作[4]提出了一种将解析树解释为建立该树的线性 "操作 "序列。这种线性化程序使他们能够训练一个模型，该模型可以将一个句子映射到其解析树上，而不需要对任意序列的表述进行任何修改。

训练后的模型有一层512个LSTM单元，单词由大小为512的嵌入向量表示。我们使用了类似于[22]中描述的注意机制，该机制有助于在考虑下一个输出标记产生y_t，只通过对输入序列对应的LSTM状态向量应用softmax来关注输入序列的一部分。输入词词典包含大约9万个词，而目标词典包含128个用于描述树的符号。在预定采样方法中，我们对ϵ, , 使用了反西格玛衰变时间表。

解析与图像说明有很大不同，因为人们要学习的功能几乎是确定的。与图像有大量有效的标题相比，大多数句子都有一个独特的解析树（尽管存在一些非常困难的情况）。因此，该模型的运行几乎是确定的，这可以通过观察训练和测试的复杂度与图像字幕相比非常低（1.1比7）而看出。

这种不同的运行机制使得比较很有趣，因为人们不会期望基线算法会犯很多错误。然而，从表2中可以看出，计划抽样有一个积极的影响，这个影响是与辍学相加的。在这个表中，我们报告了WSJ 22开发集[23]上的F1得分。我们还应该强调，只有40k个训练实例，所以过拟合对我们系统的性能有很大的贡献。训练过程中的抽样效果是否对过拟合或训练/推理不匹配有帮助还不清楚，但结果是积极的，与放弃相加。再一次，通过总是从自身取样而不是使用groundtruth以前的token作为输入来训练的模型产生了非常糟糕的结果，事实上是如此糟糕，以至于产生的树往往不是有效的树（因此相应的F1指标中的"-"）。

表2：在解析任务的验证集上的F1得分（越高越好）。

办法	F1
基线LSTM 基线LSTM	86.54
与辍学率	87.0
经常性的抽样调查	-
	88.08
	88.68

预定的抽样调查 有辍学的预定抽样	

4.3 语音识别

对于语音识别实验，我们使用了与本文其他部分略有不同的设置。每个训练实例是一个输入/输出对（X，Y），其中X是一串T输入向量x_1，x_2，$--x_T$，Y是一串T标记y_1，y_2，$--y_T$，因此每个y_t与相应的x_t对齐。这里，x_t是在第t帧由log Mel滤波器组谱表示的声学特征，而y_t是相应的目标。使用的目标是HMM-状态标签，产生于一个GMM-HMM配方，使用Kaldi工具包[24]，但很有可能是音素标签。这一设置与其他实验不同，我们使用的模型如下：

$$\begin{aligned}\log P(Y|X;\vartheta) &= \log P(y_1^T|x_1^T;\vartheta)\\ &= \sum_{t=1}^{T} \log P(y_t|y_1^{t-1}, x_1^t;\vartheta)\\ &= \sum_{t=1}^{T} \log P(y_t|h_t;\vartheta)\end{aligned} \tag{5}$$

其中，h_t是由一个循环神经网络计算出来的，如下所示：

$$h_t = \begin{cases} f(\mathbf{0}_n, S, x_1;\vartheta) & \text{如果} t = 1, \\ f(h_{t-1}, y_{t-1}, x_t;\vartheta) & \text{否则。}\end{cases} \tag{6}$$

其中$\mathbf{0}_n$是一个0的向量，其维度与h_t相同，S是一个添加到字典中的额外标记，代表每个序列的开始。

我们使用TIMIT语料库和KALDI工具包为这些实验生成数据。[4]语料库和KALDI工具包，如文献[25]所述，为这些实验生成数据。实验中使用了标准配置--40维对数梅尔滤波器组及其一阶和二阶时间导数被用作每帧的输入。每一时间段使用强制对准转录本，使用训练好的GMM-HMM系统生成180维的目标。训练、验证和测试集分别有3696、400和192个序列，其平均长度为304帧。验证集被用来选择训练中的最佳时间段，而该时间段的模型参数被用来评估测试集。

训练的模型有两层250个LSTM单元和一个softmax层，分别为五种配置--基线配置，即总是向模型提供地面实况；一种配置（总是采样），即模型只从上一个时间步骤中提供自己的预测；以及三种预定采样配置（预定采样1-3），其中ϵ_i在十个历时中从最大值到最小值直线上升，然后保持在最终值不变。对于每个配置，我们训练了3个模型，并报告了它们的平均性能。每个模型的训练都是通过GMM的帧目标进行的。基线配置通常在大约14个历时后达到最佳验证精度，而采样模型在大约9个历时后达到最佳精度，此后验证精度下降。这可能是因为我们训练模型的方式并不精确--它没有考虑到我们对目标进行采样的采样概率的梯度。未来解决这个问题的努力可能会进一步改善结果。

测试是通过从波束搜索解码中找到最佳序列（使用10个波束的大小）并计算序列的错误率。我们还报告了每个模型在验证集上的下一步错误率（即模型被送入地面真相以预测下一帧的类别），以总结模型在训练目标上的表现。表3显示了结果的摘要

可以看出，基线的下一步预测比抽样输入标记的模型表现得更好。这是可以预期的，因为前者可以接触到groundtruth。然而，我们可以看到，在解码过程中，用抽样训练的模型比基线表现得更好。我们还可以看到，对于这个问题，"总是采样 "模型的表现相当好。

[4]https://catalog.ldc.upenn.edu/LDC93S1。

好。我们假设，这与数据集的性质有关。HMM对齐的状态有很多相关性--同一状态在多个帧中作为目标出现，而且大多数状态只被限制在其他状态的一个子集上。在这个任务中，用真实的标签进行下一步预测，最终会对标签的结构（y^{t-1}）给予过多的关注，而对声学输入（x^t）则关注不够。因此，当groundtruth序列与声学信息一起输入时，它取得了很好的下一步预测误差，但当groundtruth序列没有输入时，它就不能充分地利用声学信息。对于这个模型
测试条件与训练条件相差太远，它无法做出好的预测。只给自己预测的模型（总是采样）最终利用了它在声学信号中能找到的所有信息，并有效地忽略了它自己的预测来影响下一步的预测。因此在测试时，它的表现与训练时一样好。像[26]的注意力模型，直接预测电话序列，而不是高度冗余的HMM状态序列，就不会有这个问题，因为它需要充分地利用声音信号和语言模型来进行预测。尽管如此，即使在这种情况下，增加预定采样仍然有助于提高解码帧的错误率。

请注意，通常情况下，语音识别实验使用HMM来解码混合模型中的神经网络预测。这里我们完全避免使用HMM，因此我们没有HMM架构和语言模型所带来的平滑优势。因此，其结果不能直接与典型的混合模型结果相比较。

表3：语音识别实验的帧错误率（FER）。在下一步预测中（在验证集上报告），基础事实被输入以预测下一个目标，就像在训练中做的那样。在解码实验中（在测试集上报告），进行波束搜索以找到最佳序列。我们报告了四种不同的线性采样调度的结果，其中，从ϵ_s 到ϵ_e，呈线性递减。对于基线，模型只被输入了地面实况。对结果的分析见第4.3节。

办法	ϵ_s	ϵ_e	下一步 FER	解码FER
始终保持取样	0	0	34.6	35.8
预定取样1	0.25	0	34.3	**34.5**
预定取样2	0.5	0	34.1	35.0
预定取样 3	0.9	0.5	19.8	42.0
基线LSTM	1	1	15.0	46.0

5 总结

使用递归神经网络来预测标记序列有许多有用的应用，如机器翻译和图像描述。然而，目前训练它们的方法，即以状态和前一个正确的标记为条件，一次预测一个标记，与我们实际使用它们的方式不同，因此很容易在决策路径上积累错误。在本文中，我们提出了一种*课程学习*方法，将训练目标从一个简单的任务，即前一个标记是已知的，慢慢变为一个现实的目标，即由模型本身提供。在几个序列预测任务上的实验产生了性能的改善，同时没有产生更长的训练时间。未来的工作包括通过抽样决策反向传播错误，以及探索更好的抽样策略，包括对模型本身的一些信心测量进行调节。

参考文献

[1] Y.Bengio, P. Simard, and P. Frasconi.学习长期的依赖性是很难的。*IEEE Transactions on Neural Networks*, 5(2):157-166, 1994.

[2] S.Hochreiter和J. Schmidhuber.长短期记忆.*神经计算*, 9(8), 1997.

[3] I.Sutskever, O. Vinyals, and Q. Le.用神经网络进行序列到序列的学习。In *Advances in Neural Information Processing Systems, NIPS*, 2014.

[4] O.Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton。语法是一门外语。在 *arXiv:1412.7449*, 2014.

[5] O.Vinyals, A. Toshev, S. Bengio, and D. Erhan.Show and tell：一个神经性的图像标题生成器。在*IEEE计算机视觉和模式识别会议，CVPR*，2015。

[6] J.Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell.用于视觉识别和描述的长期递归卷积网络。在*IEEE计算机视觉和模式识别会议上，CVPR*，2015。

[7] Y.Bengio, J. Louradour, R. Collobert, and J. Weston.课程学习。载于*《国家间机器学习会议论文集》，ICML*，2009。

[8] J.D. Lafferty, A. McCallum, and F. C. N. Pereira.条件随机场：用于分割和标记序列数据的概率模型。在*第十八届国际机器学习会议论文集，ICML*，第282-289页，美国加州旧金山，2001年。摩根考夫曼出版公司 Inc.

[9] H.Daume´ III, J. Langford, and D. Marcu.基于搜索的结构化预测的分类。*机器学习杂志》*，2009年。

[10] S.Ross, G. J. Gordon, and J. A. Bagnell.模仿学习和结构化预测对无悔在线学习的还原。*人工智能和统计学研讨会论文集，AISTATS*，2011。

[11] A.Venkatraman, M. Herbert, and J. A. Bagnell.改进学习型时间序列模型的多步骤预测。在*第二十九届AAAI人工智能会议上，AAAI*，2015。

[12] M.Collins and B. Roark.用感知器算法进行增量解析。在*计算语言学协会的会议上，ACL*，2004。

[13] Y.Goldberg and J. Nivre.弧形依存关系解析的动态神谕。In *Proceedings of COL- ING*, 2012.

[14] J.Mao, W. Xu, Y. Yang, J. Wang, Z. H. Huang, and A. Yuille.用多模态递归神经网络（m-rnn）进行深度字幕。In *International Conference on Learning Representations, ICLR*, 2015.

[15] R.Kiros, R. Salakhutdinov, and R. Zemel.用多模态神经语言模型统一视觉-语义嵌入。在*TACL*, 2015.

[16] A.Karpathy和F.-F.Li.用于生成图像描述的深度视觉语义排列。在*IEEE计算机视觉和模式识别会议上，CVPR*，2015。

[17] H.Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig。从标题到视觉概念再到后面。在*IEEE计算机视觉和模式识别会议上，CVPR*，2015。

[18] R.Vedantam, C. L. Zitnick, and D. Parikh.CIDEr：基于共识的图像描述评估。在*IEEE计算机视觉和模式识别会议，CVPR*，2015。

[19] T.-Y.Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla´r, and C. L. Zitnick.Microsoft coco：*arXiv:1405.0312*, 2014.

[20] S.Ioffe and C. Szegedy.批量归一化：通过减少内部协变量偏移来加速深度网络训练。在*国际机器学习会议论文集，ICML*，2015。

[21] Y.Cui, M. R. Ronchi, T. -Y.Lin, P. Dollr, and L. Zitnick. 微软coco字幕挑战。http://mscoco.org/dataset/#captions-challenge2015，2015。

[22] D.Bahdanau, K. Cho, and Y. Bengio.通过联合学习对准和翻译的神经机器翻译。In *International Conference on Learning Representations, ICLR*, 2015.

[23] E.Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel.Ontonotes：90%的解决方案。在*NAACL人类语言技术会议论文集中，短文*，第57-60页，美国纽约市，2006年6月。计算语言学协会。

[24] D.Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek、Y.Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely.kaldi语音识别工具包。在*IEEE 2011年自动语音识别和理解研讨会*上。IEEE信号处理协会，2011年12月。IEEE目录号：CFP11SRW-USB。

[25] N.Jaitly.*探索发现语音信号中特征的深度学习方法。*博士论文，多伦多大学，2014年。

[26] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.使用基于注意力的递归NN进行端到端连续语音识别：*arXiv preprint arXiv:1412.1602*, 2014.