上交大提出SRNN可并行计算

新智元报道

来源：上海交通大学

作者：Zeping Yu，Gongshen Liu

编辑：肖琴、金磊

【新智元导读】上海交通大学最新提出切片循环神经网络（SRNN），其速度是标准RNN的136倍，并且还能更快！对六个大型情绪分析数据集的实验表明，SRNN的性能均优于标准RNN。

论文和开源代码地址：

https://arxiv.org/pdf/1807.02291.pdf

https://github.com/zepingyu0512/srnn

在许多NLP任务中，循环神经网络（RNN）取得了巨大的成功。但是，这种循环的结构使它们难以并行化，因此，训练RNN需要大量的时间。

上海交通大学的Zeping Yu 和Gongshen Liu，在论文“*Sliced Recurrent Neural Networks*”中，提出了全新架构“切片循环神经网络”（SRNN）。SRNN可以通过将序列分割成多个子序列来实现并行化。SRNN能通过多个层获得高级信息，而不需要额外的参数。

研究人员证明了当使用线性激活函数时，标准RNN是SRNN的一个特例。在不改变循环单元的情况下，**SRNN的速度是标准RNN的136倍，并且当训练更长的序列时可能会更快**。对六个大型情绪分析数据集的实验表明，SRNN的性能优于标准RNN。

提高RNN训练速度的多种方法

循环神经网络（RNN）已经被广泛用于许多NLP任务，包括机器翻译、问题回答、图像说明和文本分类。RNN能够获得输入序列的顺序信息。最受欢迎的两个循环单元是**长短期记忆（LSTM）**和**门控循环单元（GRU）**，两者都可以将先前的记忆存储在隐藏状态，并使用门

控机制来确定应该在何种程度将先前的记忆应与当前的输入结合。但是，由于其循环的结构，RNN不能并行计算。因此，训练RNN需要花费大量时间，这限制了学术研究和工业应用。

为了解决这个问题，一些学者尝试在NLP领域**使用卷积神经网络（CNN）来代替RNN**。但是，CNN无法获得序列的顺序信息，而顺序信息在NLP任务中非常重要。

一些学者试图**通过改进循环单元来提高RNN的速度**，并取得了良好的效果。通过将CNN与RNN相结合，准循环神经网络（QRNN）的速度提高了16倍。2017年Tao Lei等人提出了简单循环单元SRU（simple recurrent unit），比LSTM快5-10倍。类似地， strongly-typed循环神经网络（T-RNN）和最小门控单元（MGU）也是可以改进循环单元的方法。

虽然RNN在这些研究中实现了更快的速度，并且循环单元得到了改善，但整个序列中的循环结构是保持不变的。我们仍然需要等待上一步的输出，因此RNN的瓶颈仍然存在。在本文中，我们提出切片循环神经网络（SRNN），它在不改变循环单元的情况下，能够比标准RNN快得多。我们证明了当使用线性激活函数时，标准RNN是SRNN的一个特例，SRNN能够获得序列的高级信息。

为了将我们的模型与标准RNN进行比较，我们选择GRU作为循环单元。其他的循环单元也可以用于我们的结构，因为我们改进了整个序列中的整体RNN结构，而不仅仅是改变循环单元。我们在6个大型数据集上完成了实验，证明SRNN在所有数据集上的性能优于标准RNN。

我们开源了实现代码:

https://github.com/zepingyu0512/srnn

切片循环神经网络SRNN的结构

我们构建了一个新的RNN结构，称为切片循环神经网络(SRNN)，如图2所示。在图2中，循环单位也称为A。

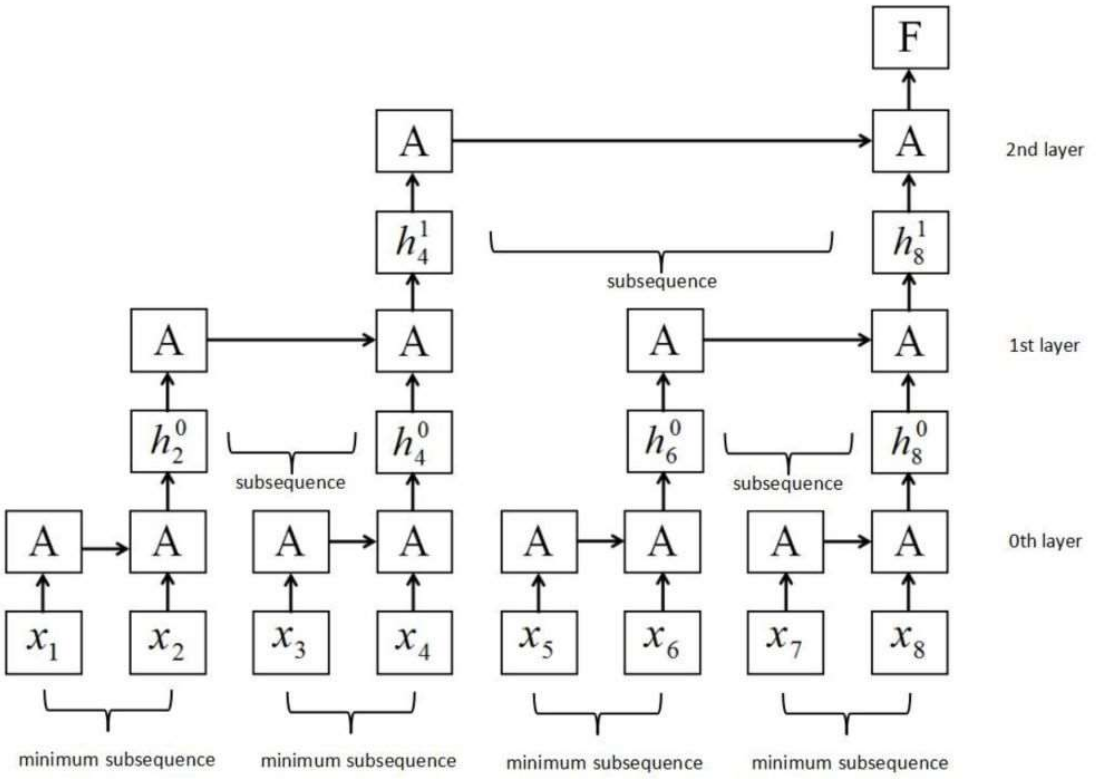

图2：SRNN结构。它是通过将输入序列分割成几个长度相等的最小子序列来构造的。循环单元可以在每层子序列上同时工作，信息可以通过多层传递。

输入序列X的长度为T，输入序列为:

$$X = [x_1, x_2, ..., x_T]$$

其中x是每个步骤的输入，它可能具有多个维度，例如单词嵌入。然后将X分割成长度相等的n个子序列，每个子序列的长度n为:

$$l = \frac{T}{n}$$

其中n为切片数，序列X可表示为:

$$X = [N_1, N_2, ..., N_n]$$

其中每个子序列是:

$$N_p = [x_{(p-1)*l+1}, x_{(p-1)*l+2}, ..., x_{p*l}]$$

类似地，我们将每个子序列N再次切割成相等长度的n个子序列，然后重复该切片操作k次，直到我们在底层有一个适当的最小子序列长度（我们称之为第0层，如图2所示）， 通过切片k次获得k + 1层。 第0层的最小子序列长度为：

$$l_0 = \frac{T}{n^k}$$

第0层的最小子序列数为:

$$s_0 = n^k$$

因为第p层（p> 0）上的每个父序列被切成n个部分，所以第p层上的子序列的数量是：

$$s_p = n^{k-p}$$

并且第p层的子序列长度为：

$$l_p = n$$

以图2为例。 序列长度T为8，切片操作次数k为2，每个第p层的切片编号n为2。将序列切片两次后，在第0层得到4个最小子序列，每个最小子序列的长度为 2。如果序列的长度或其子序列的长度不能除以n，我们可以利用填充法或在每一层上选择不同的切片编号。 不同的k和n可以用于不同的任务和数据集。

SRNN和标准RNN之间的区别在于SRNN将输入序列切分为许多最小子序列，并利用每个子序列上的循环单元。通过这种方式，子序列可以很容易地并行化。在第0层，循环单元通过连接结构对每个最小子序列起作用。 接下来，我们获得第0层上每个最小子序列的最后隐藏状态，这些状态在第1层用作其父序列的输入。 然后我们使用第（p-1）层上每个子序列的最后隐藏状态作为其第p层上的父序列的输入，并计算第p层上子序列的最后隐藏状态：

$$h_t^1 = \overrightarrow{GRU}^0(mss^0_{(t-l_0+1)\sim t})$$

$$h_t^{p+1} = \overrightarrow{GRU}^p(h^p_{t-l_p} \sim h_t^p)$$

其中

h_l^p

为第p层上的数l隐藏状态，mss为第0层上的最小子序列，不同层上可以使用不同的GRU。在每层上的每个子父序列之间重复该操作，直到我们得到顶层（第k层）的最终隐藏状态F：

$$F = \overrightarrow{GRU}^k(h^k_{t-4_k} \sim h^k_t)$$

实验：序列长度越长，SRNN处理的速度优势就越大

数据集

我们在六个大规模情绪分析数据集上评估SRNN。表1列出了数据集的信息。我们选择80%的数据用于训练，10%用于验证，10%用于测试。

Dataset	Classes	Documents	Max words	Average words	Vocabulary
Yelp 2013	5	468,608	1060	129.2	202,058
Yelp 2014	5	670,440	1053	116.1	210,353
Yelp 2015	5	897,835	1092	108.3	228,715
Yelp_P	2	598,000	1073	139.7	308,028
Amazon_F	5	3,650,000	441	82.7	1,274,916
Amazon_P	2	4,000,000	257	80.9	1,348,126

表1：数据集信息。Max words表示最大序列长度，Average words表示每个数据集中句子的平均长度。

结果和分析

每个数据集的结果如表2所示。我们选择不同的n和k值，得到不同的SRNN。例如，SRNN（16,1）表示n = 16且k = 1，当T为512时，可以得到长度为32的最小子序列；当T为256时，可以得到长度为16的最小子序列。我们将4个SRNN与标准RNN进行了比较。每个数据集中，粗体字表示性能最高的模型和速度最快的模型。

Dataset	Model	Parameters	Validation	Test	Time/Epoch
Yelp 2013	GRU	5.76M	66.56	66.12	3172s
	SRNN (16,1)	5.77M	67.18	**67.03**	270s
	SRNN (8,2)	5.79M	67.11	66.80	**145s**
	SRNN (4,3)	5.80M	67.26	66.72	164s
	SRNN (2,8)	5.85M	66.30	66.41	204s
	DCCNN	5.78M	64.91	64.79	67s
Yelp 2014	GRU	5.76M	70.37	70.63	4142s
	SRNN (16,1)	5.77M	70.53	70.70	388s
	SRNN (8,2)	5.79M	70.35	**70.76**	**201s**
	SRNN (4,3)	5.80M	70.25	70.48	238s
	SRNN (2,8)	5.85M	69.50	69.70	284s
	DCCNN	5.78M	68.46	68.66	96s
Yelp 2015	GRU	5.76M	72.52	72.89	4434s
	SRNN (16,1)	5.77M	73.09	**73.50**	510s
	SRNN (8,2)	5.79M	72.84	73.30	319s
	SRNN (4,3)	5.80M	72.98	73.29	**309s**
	SRNN (2,8)	5.85M	72.37	72.75	367s
	DCCNN	5.78M	70.69	70.94	131s
Yelp_P	GRU	5.76M	96.02	95.96	3170s
	SRNN (16,1)	5.77M	95.83	95.92	401s
	SRNN (8,2)	5.79M	95.87	95.99	**205s**
	SRNN (4,3)	5.80M	95.90	**96.04**	236s
	SRNN (2,8)	5.85M	95.69	95.88	297s
	DCCNN	5.78M	95.03	95.26	98s
Amazon_F	GRU	5.76M	61.54	61.36	8953s
	SRNN (16,1)	5.77M	61.65	**61.65**	1584s
	SRNN (8,2)	5.79M	61.58	61.41	**1147s**
	SRNN (4,3)	5.80M	61.50	61.40	1166s
	SRNN (2,7)	5.85M	61.04	60.88	1344s
	DCCNN	5.78M	59.64	59.60	401s
Amazon_P	GRU	5.76M	95.27	95.22	11062s
	SRNN (16,1)	5.77M	95.29	**95.26**	2144s
	SRNN (8,2)	5.79M	95.21	95.18	**1309s**
	SRNN (4,3)	5.80M	95.12	95.12	1567s
	SRNN (2,7)	5.85M	94.98	95.02	1886s
	DCCNN	5.78M	94.72	94.69	448s

表2：每个数据集上模型验证和测试的准确度和训练时间。我们构建了四种不同的SRNN结构。DCCNN是dilated casual卷积神经网络。

结果表明，在几乎没有额外参数的条件下，SRNN在所有数据集上的性能和速度都优于标准RNN。不同的SRNN在不同的数据集上都实现了最佳性能：

- SRNN（16,1）在Yelp 2013，Yelp 2015，Amazon_F和Amazon_P上都获得最高的准确度；
- SRNN（8,2）在Yelp 2014上的性能最佳；
- SRNN（4,3）在Yelp_P上表现最好。
- K大于1时，SRNN在Yelp数据集上比标准RNN快了将近15倍，可见速度的优势取决于k，n和T。
- SRNN（4,3）在Yelp 2015上速度最快， 而SRNN（8,2）在其余数据集最快（DCCNN除外）。

我们注意到，Yelp数据集上的SRNN（2,8）和Amazon数据集上的SRNN（2,7）没有达到最佳性能，但也没有在准确性方面减弱太多。这意味着SRNN能够通过多个层传输信息，因此，当我们训练非常长的序列时，SRNN可以获得显着的效果。当n为2时，SRNN具有与DCCNN相同的层数，并且SRNN的精度比DCCNN高得多。因此，这表明SRNN中的循环结构优于dilated casual卷积神经网络结构。

我们使用NVIDIA GTX 1080 GPU在5120个文档上训练模型，因为如果使用更多的数据，训练标准RNN需要花费太多时间。训练时间如表3所示。

从表3中得到惊人的结果：**序列长度越长，SRNN实现的速度优势就越大**。当序列长度为32768时，SRNN仅需52s，而标准RNN需要花费近2小时。SRNN的速度是标准RNN的136倍！并且如果使用更长的序列，速度优势可能更大。因此，SRNN可以在长序列任务上实现更快的速度，例如语音识别、字符级文本分类和语言建模。

Model	Sequence length		
	$8^3 = 512$	$8^4 = 4096$	$8^5 = 32768$
SRNN (8,2)	4s	-	-
SRNN (8,3)	-	9s	-
SRNN (8,4)	-	-	52s
GRU	54s	476s	7074s
Speed advantage	13.5x	52.9x	136.0x

表3：不同序列长度的训练时间和速度优势。对于每个序列长度，我们选择不同的SRNN结构。

SRNN的优势和重要意义

在这一部分，我们将讨论SRNN的优势和重要意义。随着RNN在许多NLP任务中取得成功，许多学者提出了不同的结构来提高RNN的速度。通过改善循环单元，很多研究都能加快RNN的速度。但是，RNN的传统连接结构几乎没有被质疑过，而这种结构里每个步骤都与它的前一步骤相关联。正是这种连接结构限制了RNN的速度。SRNN改进了传统的连接方法。我们构建了一种切片结构（sliced structure）来实现RNN的并行化。在六个大规模情感分析数据集的实验结果表明，SRNN比标准RNN具有更好的性能。原因如下：

（1）当使用标准RNN连接结构时，具门控结构的循环单元（例如GRU和LSTM）是有用的，但是当序列很长时，它们就无法存储所有的重要信息。SRNN可以将长序列分成许多短的子序列，并用短序列来获取重要信息。SRNN能够通过从第0层到顶层的多层结构传输重要信息。

（2）SRNN能够从序列中获取高级信息，而不仅仅是单词级信息。我们在512个单词的文本中使用SRNN（8,2）时，第0层可以从单词嵌入中获得句子级信息，第1层可以从句子级信息中获得段落级信息，第2层可以生成段落级信息的最终文档级表示。而标准RNN只能获得词汇级信息。虽然不可能每个文档有8个段落，每个段落有8个句子，每个句子有8个单词，但是总体的顺序信息和结构信息是统一的。以段落信息为例，人们总是在文章的开头或结尾表达自己的观点，并在文章中间举例说明。与标准RNN相比，SRNN更容易在顶层获得这些信息。

（3）在处理序列方面，SRNN类似于人脑的机制。例如，我们作为人类在得到一篇文章，并被要求回答一些问题时，我们通常不需要深入阅读整篇文章。我们会试图找到提及具体信息的段落，然后在段落中找到可以回答问题的句子和单词。SRNN可以通过多个层轻松做到这一点。

除了提高准确性之外，SRNN的最大优势是可以并行计算，实现更快的速度。我们在不同序列长度的实验表明SRNN的运行速度比标准RNN快得多。而且，在更长的序列上，SRNN可能更快。随着互联网的发展，每天都有数以亿计的数据产生，SRNN可以作为处理这些数据的新方法。

结论和未来工作

在这篇论文中，我们提出切片循环神经网络（SRNN），这是RNN的整体结构改进。 SRNN可以达到比标准RNN快得多的速度，并在六个大规模情感数据集上实现更好的性能。

SRNN在文本分类方面取得了成功。在未来的工作中，我们希望将其推广到其他NLP应用，例如问答、文本摘要和机器翻译。在序列到序列模型中，SRNN可以用作编码器，并且可以通过使用反向SRNN结构来改进解码器。此外，我们希望在一些长序列任务中使用SRNN，例如语言模型、音乐生成和音频生成。我们想探索更多SRNN的变体，例如，可以添加双向结构和注意力机制。