

AI科技大本营
码龄3年 暂无认证

1533

原创

12万+

积分



私信

搜博主文章



热门文章

感动！有人将吴恩达的视频课程做成了文字版 93102

用AI给黑白照片上色，复现记忆中的旧时光 58798

算法工程师过去这一年：理想很丰满，现实很骨感 55884

今晚直播 | 一次性掌握机器学习基础知识脉络 48564

ICLR 2019论文投稿近1600篇，强化学习最热门 45387

分类专栏

最新评论

来了来了！趋势预测算法大PK！
程序员雍正: 很不错分享~进步的路上一起努力！也期待您的点赞支持！来了来了！趋势预测算法大PK！
程序员雍正: 大佬可否认识一下~Python跳槽薪资报告：人生苦短，Python...
成长的Offer: 写的不错，学习了，学习的道路上一共进步，也期待你的关注与支持！Python多阶段框架实现虚拟试衣间，超逼...
乎你: 好文，鉴定完毕！

腾讯AI Lab 2020年度回顾：科技向善，迈向未来to50: 这么好的文章，评论这么少？

最新文章

腾讯首位17级杰出科学家诞生：腾讯AI Lab负责人张正友

百万美元技术大奖，雷军颁给了秒充和隐私保护技术团队

IT基础架构变革，Hitachi Vantara如何解决超融合（HCI）的真正痛点？

2021年 15篇 2020年 1128篇

2019年 1762篇 2018年 1227篇

2017年 317篇

难以置信！LSTM和GRU的解析从未如此清晰（动图+视频）

原创

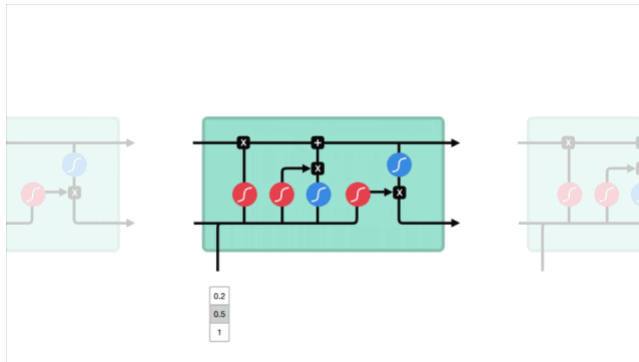
AI科技大本营

2018-09-29 20:05:43

28656

收藏 423

版权



作者 | Michael Nguyen

编译 | 蔡志兴、费棋

编辑 | Jane

出品 | AI科技大本营

【导语】机器学习工程师 Michael Nguyen 在其博文中发布了关于 LSTM 和 GRU 的详细图解指南。博文中，他先介绍了 LSTM 和 GRU 的本质，然后解释了让 LSTM 和 GRU 有良好表现的内部机制。当然，如果你还想了解这两种网络背后发生了什么，那么这篇文章就是为你准备的。

视频详解

短时记忆

RNN 会受到短时记忆的影响。如果一条序列足够长，那它们将很难将信息从较早的时间步传送到后面的时间步。因此，如果你正在尝试处理一段文本进行预测，RNN 可能从一开始就会遗漏重要信息。

在反向传播期间，RNN 会面临梯度消失的问题。梯度是用于更新神经网络的权重值，消失的梯度问题是当梯度随着时间的推移传播时梯度下降，如果梯度值变得非常小，就不会继续学习。

new weight = weight - learning rate*gradient**2.0999 = 2.1**

Not much of a difference

-

0.001

update value

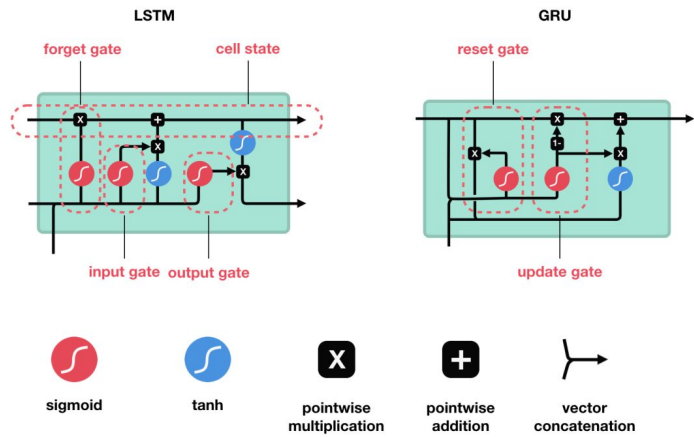
梯度更新规则

因此，在递归神经网络中，获得小梯度更新的层会停止学习——那些通常是较早的层。由于这些层不学习，RNN 可以忘记它在较长序列中看到的内容，因此具有短时记忆。

作为解决方案的 LSTM 和 GRU

LSTM 和 GRU 是解决短时记忆问题的解决方案，它们具有称为“门”的内部机制，可以调节信息流。





这些“门”可以知道序列中哪些重要的数据是需要保留，而哪些是要删除的。随后，它可以沿着长链序列传递相关信息以进行预测，几乎所有基于递归神经网络的技术成果都是通过这两个网络实现的。


LSTM 和 GRU 可以在语音识别、语音合成和文本生成中找到，你甚至可以用它们为视频生成字幕。对 LSTM 和 GRU 擅长处理长序列的原因，到这篇文章结束时你应该会有充分了解。

下面我将通过直观解释和插图进行阐述，并避免尽可能多的数学运算。

本质

让我们从一个有趣的小实验开始吧。当你在网上购买生活用品时，一般都会查看一下此前已购买该商品用户的评价。

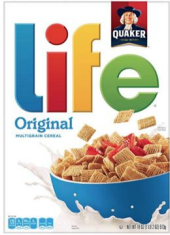
Customers Review 2,491



Thanos

September 2018
Verified Purchase


Amazing! This box of cereal gave me a perfectly balanced breakfast, as all things should be. I only ate half of it but will definitely be buying again!



A Box of Cereal
\$3.99

当你浏览评论时，你的大脑下意识地只会记住重要的关键词，比如“amazing”和“awesome”这样的词汇，而不太会关心“this”、“give”、“all”、“should”等字样。如果朋友第二天问你用户评价都说了什么，那你可能不会一字不漏地记住它，而是会说出让大脑里记得的主要观点，比如“下次肯定还会来买”，那其他一些无关紧要的内容自然会从记忆中逐渐消失。


Customers Review 2,491



Thanos

September 2018
Verified Purchase

Amazing! This box of cereal gave me a perfectly balanced breakfast, as all things should be. I only ate half of it but will definitely be buying again!

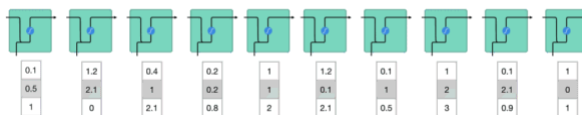


A Box of Cereal
\$3.99

而这基本上就像是 **LSTM** 或 **GRU** 所做的那样，它们可以学习只保留相关信息来进行预测，并忘记不相关的数据。

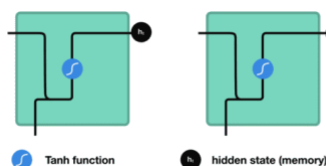
RNN 述评

为了了解 **LSTM** 或 **GRU** 如何实现这一点，让我们回顾一下递归神经网络。**RNN** 的工作原理如下：第一个词被转换成了机器可读的向量，然后 **RNN** 逐个处理向量序列。



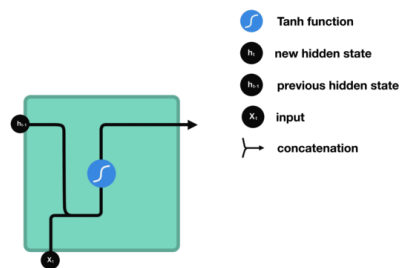
逐一处理向量序列

处理时，**RNN** 将先前隐藏状态传递给序列的下一步。而隐藏状态充当了神经网络记忆，它包含相关网络之前所见过的数据的信息。



将隐藏状态传递给下一个时间步

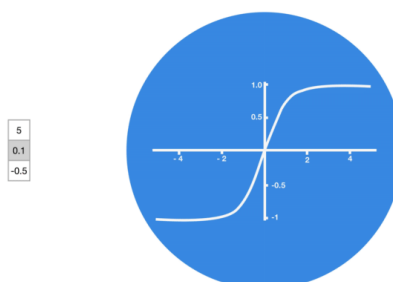
让我们看看 **RNN** 的一个细胞，了解一下它如何计算隐藏状态。首先，将输入和先前隐藏状态组合成向量，该向量包含当前输入和先前输入的信息。向量经过激活函数 **tanh** 之后，输出的是新的隐藏状态或网络记忆。



RNN 细胞

激活函数 Tanh

激活函数 **Tanh** 用于帮助调节流经网络的值。**tanh** 函数将数值始终限制在 **-1** 和 **1** 之间。



当向量流经神经网络时，由于有各种数学运算的缘故，它经历了许多变换。因此想象让一个值继续乘以 **3**，你可以想到一些值是如何变成天文数字的，这让其他值看起来微不足道。



没有 \tanh 函数的向量转换

\tanh 函数确保值保持在 $-1\sim 1$ 之间，从而调节了神经网络的输出。 你可以看到上面的相同值是如何保持在 \tanh 函数所允许的边界之间的。

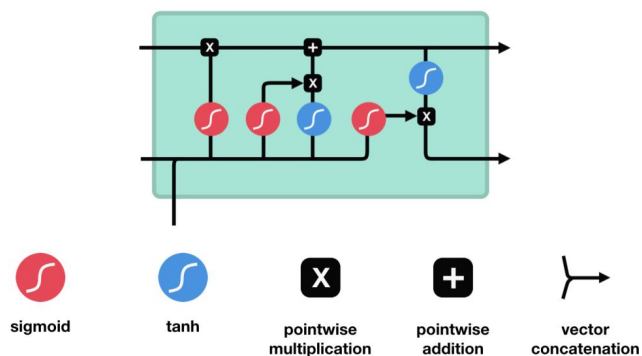


有 \tanh 函数的向量转换

这是一个 **RNN**。它内部的操作很少，但在适当的情形下（如短序列）运作的很好。**RNN** 使用的计算资源比它的演化变体 **LSTM** 和 **GRU** 要少得多。

LSTM

LSTM 的控制流程与 **RNN** 相似，它们都是在前向传播的过程中处理流经细胞的数据，不同之处在于 **LSTM** 中细胞的结构和运算有所变化。



LSTM 的细胞结构和运算

这一系列运算操作使得 **LSTM** 具有能选择保存信息或遗忘信息的功能。乍一看这些运算操作时可能有点复杂，但没关系下面将带你一步步了解这些运算操作。

核心概念

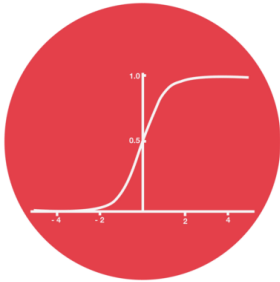
LSTM 的核心概念在于细胞状态以及“门”结构。细胞状态相当于信息传输的路径，让信息能在序列连中传递下去。你可以将其看作网络的“记忆”。理论上讲，细胞状态能够将序列处理过程中的相关信息一直传递下去。

因此，即使是较早时间步长的信息也能携带到较后时间步长的细胞中来，这克服了短时记忆的影响。信息的添加和移除我们通过“门”结构来实现，“门”结构在训练过程中会去学习该保存或遗忘哪些信息。

Sigmoid

门结构中包含着 **sigmoid** 激活函数。**Sigmoid** 激活函数与 **tanh** 函数类似，不同之处在于 **sigmoid** 是把值压缩到 $0\sim 1$ 之间而不是 $-1\sim 1$ 之间。这样的设置有助于更新或忘记信息，因为任何数乘以 0 都得 0 ，这部分信息就会剔除掉。同样的，任何数乘以 1 都得到它本身，这部分信息就会完美地保存下来。这样网络就能了解哪些数据是需要遗忘，哪些数据是需要保存。

5
0.1
-0.5

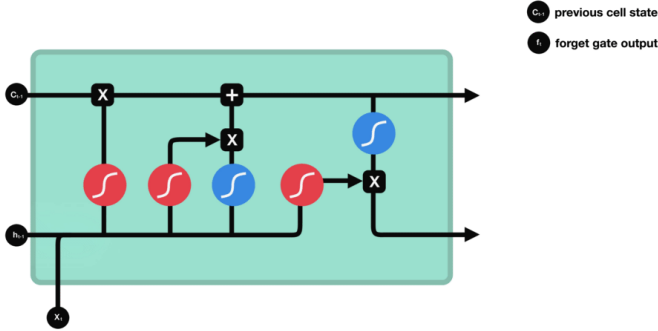


Sigmoid 将值压缩到 0~1 之间

接下来了解一下门结构的功能。LSTM 有三种类型的门结构：遗忘门、输入门和输出门。

遗忘门

遗忘门的功能是决定应丢弃或保留哪些信息。来自前一个隐藏状态的信息和当前输入的信息同时传递到 sigmoid 函数中去，输出值介于 0 和 1 之间，越接近 0 意味着越应该丢弃，越接近 1 意味着越应该保留。

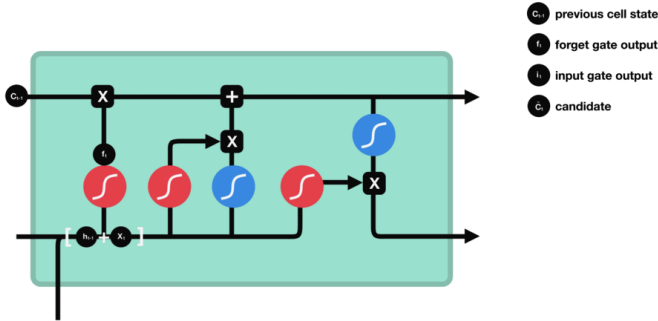


遗忘门的运算过程

输入门

输入门用于更新细胞状态。首先将前一层隐藏状态的信息和当前输入的信息传递到 sigmoid 函数中去。将值调整到 0~1 之间来决定要更新哪些信息。0 表示不重要，1 表示重要。

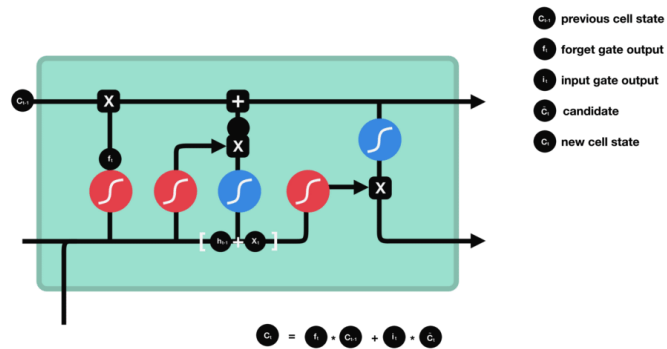
其次还要将前一层隐藏状态的信息和当前输入的信息传递到 tanh 函数中去，创建一个新的候选值向量。最后将 sigmoid 的输出值与 tanh 的输出值相乘，sigmoid 的输出值将决定 tanh 的输出值中哪些信息是重要且需要保留下来的。



输入门的运算过程

细胞状态

下一步，就是计算细胞状态。首先前一层的细胞状态与遗忘向量逐点相乘。如果它乘以接近 0 的值，意味着在新的细胞状态中，这些信息是需要丢弃掉的。然后再将该值与输入门的输出值逐点相加，将神经网络发现的新信息更新到细胞状态中去。至此，就得到了更新后的细胞状态。

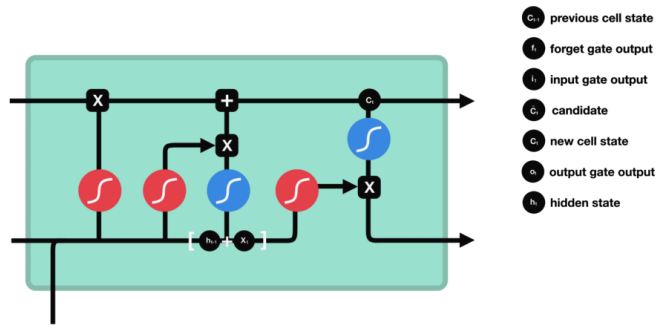


细胞状态的计算

输出门

输出门用来确定下一个隐藏状态的值，隐藏状态包含了先前输入的信息。首先，我们将前一个隐藏状态和当前输入传递到 `sigmoid` 函数中，然后将新得到的细胞状态传递给 `tanh` 函数。

最后将 `tanh` 的输出与 `sigmoid` 的输出相乘，以确定隐藏状态应携带的信息。再将隐藏状态作为当前细胞的输出，把新的细胞状态和新的隐藏状态传递到下一个时间步长中去。



输出门的运算过程

让我们再梳理一下。遗忘门确定前一个步长中哪些相关的信息需要被保留；输入门确定当前输入中哪些信息是重要的，需要被添加的；输出门确定下一个隐藏状态应该是什么。

代码示例

对于那些懒得看文字的人来说，代码也许更好理解，下面给出一个用 `python` 写的示例。

```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct
```

```
ct = [0, 0, 0]
ht = [0, 0, 0]

for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

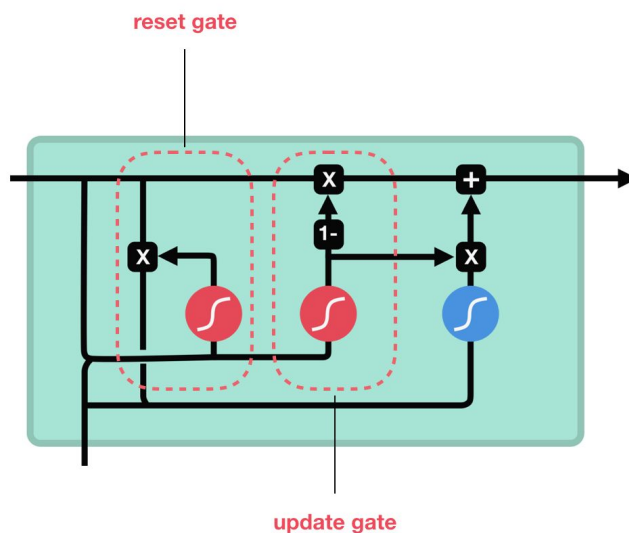
python 写的伪代码

- 1.首先，我们将先前的隐藏状态和当前的输入连接起来，这里将它称为 **combine**；
- 2.其次将 **combine** 丢到遗忘层中，用于删除不相关的数据；
- 3.再用 **combine** 创建一个候选层，候选层中包含着可能要添加到细胞状态中的值；
- 4.**combine** 同样要丢到输入层中，该层决定了候选层中哪些数据需要添加到新的细胞状态中；
- 5.接下来细胞状态再根据遗忘层、候选层、输入层以及先前细胞状态的向量来计算；
- 6.再计算当前细胞的输出；
- 7.最后将输出与新的细胞状态逐点相乘以得到新的隐藏状态。

是的，LSTM 网络的控制流程就是几个张量和一个 **for** 循环。你还可以使用隐藏状态进行预测。结合这些机制，LSTM 能够在序列处理中确定哪些信息需要记忆，哪些信息需要遗忘。

GRU

知道了 LSTM 的工作原理之后，来了解一下 GRU。GRU 是新一代的循环神经网络，与 LSTM 非常相似。与 LSTM 相比，GRU 去掉了细胞状态，使用隐藏状态来进行信息的传递。它只包含两个门：更新门和重置门。



GRU 的细胞结构和门结构

更新门

更新门的作用类似于 LSTM 中的遗忘门和输入门。它决定了要忘记哪些信息以及哪些新信息需要被添加。

重置门

重置门用于决定遗忘先前信息的程度。

这就是 GRU。GRU 的张量运算较少，因此它比 LSTM 的训练更快一下。很难去判定这两者到底谁更好，研究人员通常会两者都试一下，然后选择最合适的。

结语

总而言之，RNN 适用于处理序列数据用于预测，但却受到短时记忆的制约。LSTM 和 GRU 采用门结构来克服短时记忆的影响。门结构可以调节流经序列链的信息流。LSTM 和 GRU 被广泛地应用到语音识别、语音合成和自然语言处理等。

原文链接：<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>

2018 AI开发者大会



拒绝空谈，技术争鸣



2018 AI开发者大会（AI NEXTCon）由中国IT社区CSDN与硅谷AI社区AICamp联合出品的AI技术与产业年度盛会。多年经验淬炼，如今蓄势待发：将有近百位中美顶尖AI专家、知名企业代表以及千余名AI开发者齐聚北京，进行技术解读和产业论证。我们只讲技术，拒绝空谈，诚挚邀请AI业内人士一起共转人工智能新篇章！

2018 AI开发者大会首轮重磅嘉宾及深度议题现已火热出炉，扫码抢“鲜”看。国庆特惠，购票立享 5 折优惠！



点赞 147 评论 18 分享 收藏 423 举报 关注 一键三连

深度学习—几种常见的循环神经网络视频教学（RNN+LSTM+GRU）

05-24

深度学习的顶级循环神经网络的工作方式包括 LSTM、GRU 和 RNN。循环神经网络(RNN)在自然语言处理、语音识别等有很广泛的用途。LSTM和GRU是目前使用最广泛的两个循环神经网络的模型变种。该视频教程内容主要分为三大部分，机器学习神经网络RNN教程、LSTM、GRU。



优质评论可以帮助作者获得更高权重



评论

GRU与LSTM总结

Ireaderl的博客 6万+

一、LSTM（长短期记忆网络）LSTM是一种特殊的RNN类型，一般的RNN结构如下图所示，是一种将以往学习的结..果应用到当前学习的模型，但是这种一般的RNN存在着许多的弊端。举个例子，如果我们要预测“the clouds are in t he sky”的最后一个单词，因为只在这一个句子的语境中进行预测，那么将很容易地预测出是这个单词是sky。在这样的场景中，相关的信息和预测的词位置之间的间隔是非常小的

LSTM 和GRU的区别

Adrianna的专栏 4万+

Reference: https://cs224d.stanford.edu/lecture_notes/LectureNotes4.pdf Empirical Evaluation of Gated Recur... nt Neural Networks on Sequence Modeling https://feature.engineering/difference-between-lstm-a

mn,lstm与GRU详解

05-20

三种循环神经网络的介绍与比较，帮助大家理解对循环神经网络的理解

一文了解LSTM和GRU背后的秘密（绝对没有公式）

weixin_33672109的博客 226

你好，欢迎阅读长短期记忆网络（LSTM）和门控循环单元（GRU）的图解文章。我是Michael，是AI语音助理领域...的机器学习工程师。在这篇文章中，我们将从LSTM和GRU背后的原理出发。然后我将解释允许LSTM和GRU表现良好的内部机制。如果你想了解这两个网络的背后到底是什么，那么这篇文章就是为你准备的。问题根源短期记忆递归神经网络(RNN)...

带你深入AI（5）- 自然语言处理领域：RNN LSTM GRU

谢杨易的博客 1万+

1 引言深度学习算法模型大致分为三类，物体分类，目标检测和自然语言处理。前面两章我们分析了物体分类算法...目标检测算法，着重讲解了算法执行流程，优缺点，以及他们的优化技巧。本文分析最后一个大类，即自然语言处理领域。与物体分类和目标检测不同，自然语言处理中，前一个时刻和后一个时刻会对我们当前的输出结果产生影响，也就是网络模型是与时序相关的。比如“我是法国人，我会说”这个句子中，我们要预测最后的词语，需要

深度学习笔记——RNN（LSTM、GRU、双向RNN）学习总结

mpk_no1的博客 5万+

本文是关于RNN和RNN的变种LSTM、GRU以及BIRN的学习总结。

史上最小白之LSTM 与 GRU详解

Tink1995的博客 826

1.前言 上一篇介绍了循环神经网络 RNN，正好今天周日在家闲着也是闲着，就干脆趁热打铁，把LSTM和GRU也介绍一下吧。不太清楚RNN原理的同学可以参考我上一篇博客：史上最小白值RNN详解 2.LSTM（Long short-term memory）2.1为什么需要LSTM Long short-term memory，也就是长短期记忆，那么从字面意思来理解LSTM就是一种不仅具有短期记忆而...

深度学习与自然语言处理(7)_斯坦福cs224d 语言模型，RNN，LSTM与GRU

寒小阳 5万+

说明：本文为斯坦福大学CS224d课程的中文版内容笔记，已得到斯坦福大学课程@Richard Socher教授的授权翻译。与发表 1.语言模型 语言模型用于对特定序列的一系列词汇的出现概率进行计算。一个长度为m的词汇序列{w1,...,wm}的联合概率被表示为P(w1,...,wm)。由于在得到具体的词汇之前我们会先知道词汇的数量，词汇wi的属性变化会根据其在输入文档中的位置而定，而联合概率P(w1,...,wm)的计

RNN、GRU、LSTM

Hayden的博客 486

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/weixin_42432468 学习心得： 1、.每周的视频课程看一到两遍 2、做笔记 3、做每周的作业练习，这个里面的含金量非常高。掌握后一定要自己做一遍，这样以后用起来才能得心应手。对RNN、Simplified GRU、Full GRU、LSTM单元的理解： 1、RNN Unit 2、S...

理解LSTM网络

Deep Learning and NLP Farm 1696

英文原地址：http://colah.github.io/posts/2015-08-Understanding-LSTMs/ 中文原地址：http://www.jianshu.com/p...dc9f41f0b29 Recurrent Neural Networks 人类并不是每时每刻都从一片空白的大脑开始他们的思考。在你阅读这篇文章时候，你都是基于自己已经拥有的对先前所见词的理

LSTM与GRU

weixin_42421001的博客 4451

很多博客已经详细讲述了lstm和gru的结构及公式，这里就不一一介绍了，参考下面链接，讲的挺详细 https://blog...dn.net/gd_1101/article/details/79376798 这篇文章主要讲自己对lstm与gru的区别及联系的理解。在传统RNN中，由于反向传播过程中出现激活函数的累乘，容易造成梯度消失和梯度爆炸，这就造成在较长的time-steps下，后面...

RNN, LSTM, GRU 公式总结

张小彬的专栏 3万+

RNN参考 RNN wiki 的描述，根据隐层 hth_t 接受的是上时刻的隐层（hidden layer）ht-1h_{t-1} 还是上时刻的输出（output layer）yt-1y_{t-1}，分成了两种 RNN。定义如下： Elman network 接受上时刻的隐层 ht-1h_{t-1} Jordan network 接受上时刻的输出 yt-1y_{t-1} 但是看了很多的教程，感觉应

LSTM和GRU的对比和分析

Kevin.Shi 5596

先给出一些结论： GRU和LSTM的性能在很多任务上不分伯仲。 GRU 参数更少因此更容易收敛，但是数据集很大的情况下，LSTM表达性能更好。从结构上来说，GRU只有两个门（update和reset），LSTM有三个门（forget, input, output），GRU直接将hidden state 传给下一个单元，而LSTM则用memory cell 把hidden state 包装起来。基...

RNN、LSTM、GRU学习资料记录

夏至是个程序媛 328

RNN、LSTM：人人都能看懂的LSTM GRU：人人都能看懂的GRU 其他参考： 1、深度学习笔记——RNN（LSTM、GRU、双向RNN）学习总结 一、RNN RNN：循环神经网络（Recurrent Neural Network，RNN）是一种用于处理序列数据的神经网络。 1、单个输入xxx； 公式：f:h'y=f(h,x)f: h'y = f(h,x)f:h'y=f(h,x) 输入...

【串讲总结】RNN、LSTM、GRU、ConvLSTM、ConvGRU、ST-LSTM

Al瞬牛车 4380

前言平时很少写总结性的文章，感觉还是需要阶段性总结一些可以串在一起的知识点，所以这次写了下。因为我写...的内容主要在时序、时空预测这个方向，所以主要还是把rnn、lstm、gru、convl...

torch.nn.GRU()函数解读

qq_40178291的博客 1万+

参考链接 代码示例 一个序列时： >>> import torch.nn as nn >>> gru = nn.GRU(input_size=50, hidden_size=50, b...tch_first=True) >>> embed = nn.Embedding(3, 50) >>> x = torch.LongTen...

LSTM与GRU的一些比较—论文笔记

meanme的专栏 6万+

reference: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling 1.概要：传统的RN...在训练long-term dependencies 的时候会遇到很多困难，最常见的便是vanish gradient problem。期间有很多解决这个问题方法被发表。大致可以分为两类：一类是以新的方法改

LSTM、GRU

算法探索之路 124