

CSDN

博客 (http://blog.csdn.net?ref=toolbar)

学院 (http://edu.csdn.net?ref=toolbar)

下载 (http://download.csdn.net?ref=toolbar)

GitChat (http://gitbook.cn/?ref=csdn) 更多 ▾

1

Q

🔍

📄

登录 (https://passport.csdn.net/account/login?ref=toolbar)

注册 (https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)

【深度学习】One Model to Learn Them All详解

原创

2017年07月04日 21:33:53

标签：深度学习 (http://so.csdn.net/so/search/s.do?q=深度学习&t=blog) /

谷歌 (http://so.csdn.net/so/search/s.do?q=谷歌&t=blog)

🗒️ 1828

👤

Kaiser, Lukasz, et al. "One Model To Learn Them All." arXiv preprint arXiv:1706.05137 (2017).

概述

Google于2017年6月16日在arxiv上提交了这篇论文，甫一问世立刻引发各方关注。除了标题劲爆之外，本文的野心和气魄令人惊叹，实验也确实给出了一些相当有信息量的结果。

项目的github页面 (https://github.com/tensorflow/tensor2tensor)给出了基于tensorflow的源码，完成度一般。本文结合此源码讲解系统结构。

系统

问题

本文尝试用一个通用模型解决跨领域的各类人工智能问题，例如：

- 图像分类（图像 -> 类标）
- 看图说话（图像 -> 自然语言）
- 翻译（自然语言 -> 自然语言）
- 语义分割（自然语言 -> 分割+类标）

各领域输入输出的信息类别不同，在本文中称为不同形态（modality）。

这种向着大一统模型的努力并非本文首创，创新点如下：

工作	领域	任务
以往文章	单一	多个
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！ e.g. 翻译	英翻法+英翻德	

shenxiaolu1984 (http://bl...)

+关注

(http://blog.csdn.net/shenxiaolu1984)

原创 69

粉丝 641

喜欢 13

未开通 (https://gitee.com/shenxiaolu1984)

- 他的最新文章
- 更多文章 (http://blog.csdn.net/shenxiaolu1984)
- 【目标检测】YOLO: You Only Look Once (http://blog.csdn.net/shenxiaolu1984/article/details/78826995)
- 【深度学习】一张图看懂Receptive Field (http://blog.csdn.net/shenxiaolu1984/article/details/78815922)
- 【推荐系统】Factorization Machine (http://blog.csdn.net/shenxiaolu1984/article/details/78740481)
- 【优化】共轭函数(Conjugate Function)超简说明 (http://blog.csdn.net/shenxiaolu1984/article/details/78194053)
- 【优化】对偶上升法(Dual Ascent)超简说明 (http://blog.csdn.net/shenxiaolu1984/article/details/78175382)

QUALCOMM

Unable to Connect

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fblog.csdn.net%2Fshenxiaolu1984%2Farticle%2F78826995

广告

他的热门文章

【目标检测】Faster RCNN算法详解 (http://blog.csdn.net/shenxiaolu1984/article/details/51152614)

🗒️ 97695

登录

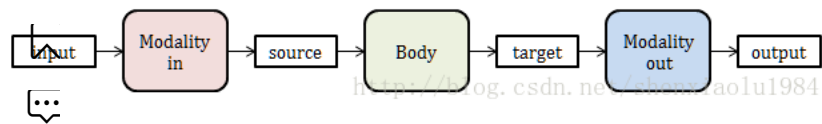
注册

✕

工作	领域	任务
本文	多个	多个
e.g.	翻译+图像分类	英翻法+英翻德+1000类分类

结构

为了适应不同形态的输入和输出，本文的网络被抽象成如下结构：



input、output：相同/不同形态的数据（例如图像和类标）。
source、target：系统内部的表达。

系统通过三个部分：modality_in，modality_out 以及 body 来完成数据流。

三大理念

- 绝大部分计算量都集中在 body 网络中，两个 modality 网络设计尽量精简。
- 系统内部的表达（target，source）尺寸不固定。
- 对于相同形态的不同问题（例如“看图说话”和“英翻德”的输出），使用相同的 modality 网络

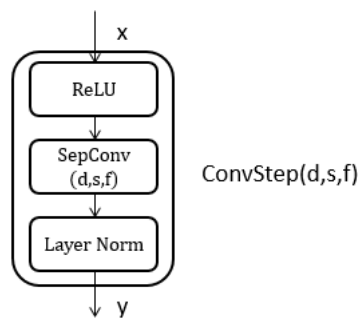
以下分别介绍 body 网络和 modality 网络的具体结构。

body网络-模块

body 网络各部分都由三种基本模块（block）构成，以下一一介绍。

卷积模块 Conv

子模块- ConvStep



SepConv：分层卷积，类似这篇博客介绍过的Factorized卷积 ([http://blog.csdn.net/shenxiaolu1984/article](http://blog.csdn.net/shenxiaolu1984/article/details/52268391) 加入CSDN，享受更精准的内容推荐，与500万程序员共同成长，更多精彩内容，等你来发现！)。

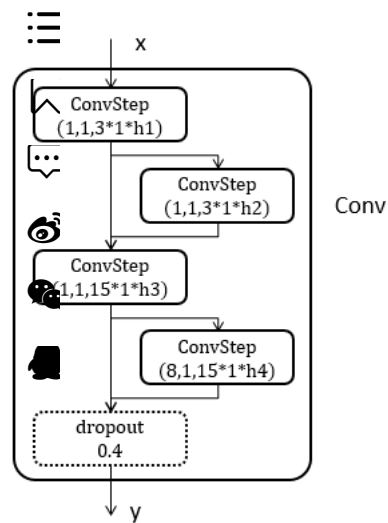
- 【目标检测】Fast RCNN算法详解 (<http://blog.csdn.net/shenxiaolu1984/article/details/51036677>)
71107
- 【目标检测】RCNN算法详解 (<http://blog.csdn.net/shenxiaolu1984/article/details/51066975>)
60867
- 【目标跟踪】KCF高速跟踪详解 (<http://blog.csdn.net/shenxiaolu1984/article/details/50905283>)
36065
- 【深度学习】生成对抗网络Generative Adversarial Nets (<http://blog.csdn.net/shenxiaolu1984/article/details/52215983>)
33681

LayerNorm : 分层归一化。

实现参见/models/common_layers.py ()中 conv_block_internal 函数/ subseparable_conv_block 函数。

Conv 构成

使用上述子模块组成卷积模块 Conv :



主体结构是两个residual结构。最后虚线的dropout只在训练时使用。

实现参见/models/slicenet.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/slicenet.py>)中 multi_conv_res 函数。

注意力模块Attention

在处理时间序列信号时，以往工作多采用RNN, LSTM类型系统。此类系统每一个时刻输出取决于前时刻记忆，天然地不利于并行计算，在训练时尤其耗时。

本文则利用 Attention 模块，能够同时处理输入序列的各个元素。细节参考自同一团队的论文 Attention is All You Need 1。

该文对应模型为/models/transformer.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py>)

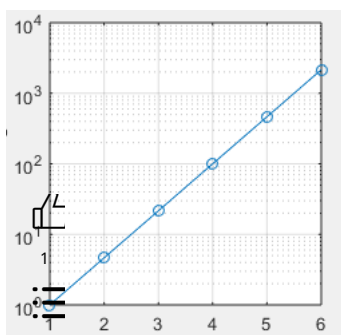
子模块-AddTiming

由于本文不再将时间序列顺序输入系统，所以需要额外告知系统每一元素在序列中的相对位置。

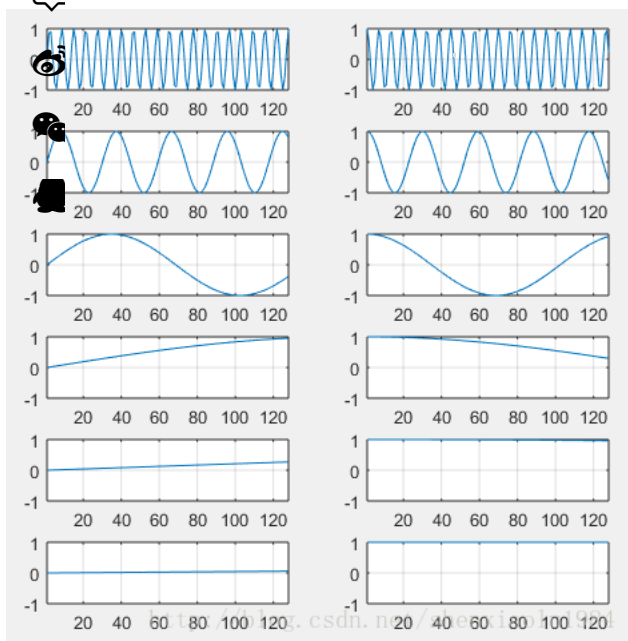
首先考虑1维信号 x ，输入尺寸为 $L \times D$ 。

对于 D 个通道，按照指数坐标均匀设置从 2π 到 $10000 \cdot 2\pi$ 的周期，共有 $D/2$ 个采样 $T_0, T_1 \dots T_{D/2-1}$ 。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



每通道的时间信号为上述波长的正弦/余弦，其自变量范围为 $[0, L - 1]$ (e.g. $D=12, L=128$) :



得到的时间信号和输入信号尺寸相同，直接相加。

$$y = x + \text{timing}$$

对于n维信号，输入尺寸为 $L_1, L_2 \dots L_n, D$ 。

采样的周期数量只需要是1维情况的 $1/n$: $T_0, T_1 \dots T_{(D/2n)-1}$ 。

对于每一维度，生成 $D/(2n)$ 对不同频率的正弦/余弦信号，扩展为 $L_1 \cdot L_2 \dots L_n$ 大小。

共有 D 个时间信号，分别加到 D 个通道上。

实现参见/models/common_attention.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/common_attention.py)中 add_timing_signal_1d 和 add_timing_signal_2d 函数。

子模块- Dot-Prod Attention

注意力网络有三个输入

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！
Q (Query) : 想要考察的一组当前对象属性。尺寸为 $L_q \times D_k$ 。

登录

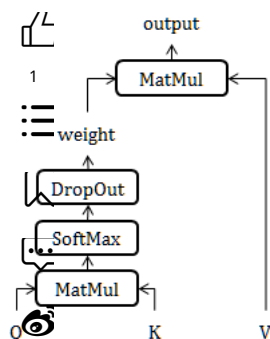
注册

×

K (Key) : 已经存在的一组参考对象属性。尺寸为 $L_{kv} \times D_k$ 。

V (Value) : 参考对象的值。尺寸为 $L_{kv} \times D_v$ 。

输出 : 当前对象的值。尺寸为 $L_q \times D_v$ 。如下图计算。



其物理意义是，考察Q和K中元素的两两相似程度，用相似程度作为权重，将V的加权和作为输出。

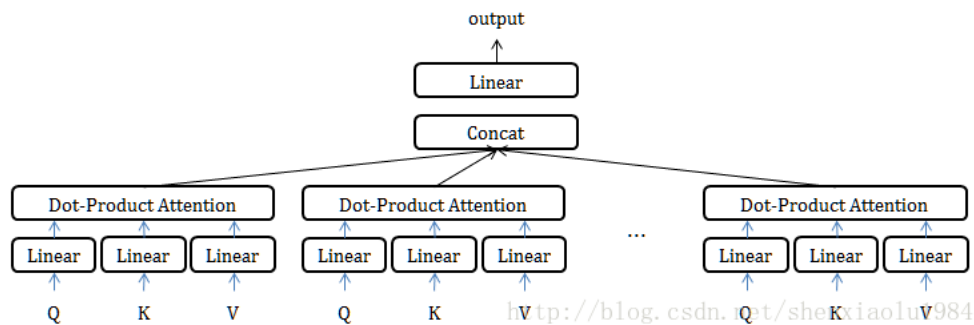
实现参见/models/common_attention.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/common_attention.py)中的 dot_product_attention 函数。

子模块- Multi-Head Attention

首先在前述 Dot-Product Attention 的三个输入端添加线性投影；

之后将 g 个这样的结果串接起来；

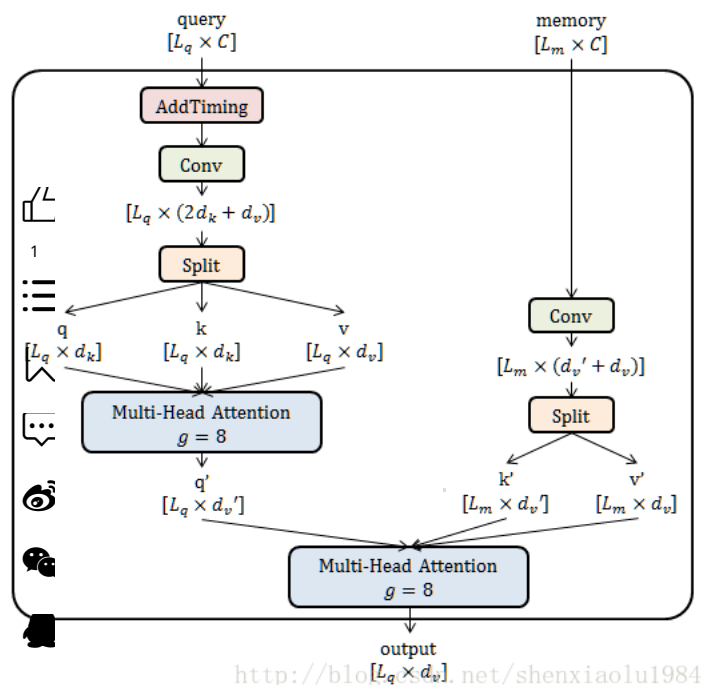
最后重新投影成系统内部表达需要的维度。



实现参见/models/common_attention.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/common_attention.py)中的 multihead_attention 函数。

Attention构成

Attention模块有两个输入：尺寸为 $L_q \times C$ 的 query，以及尺寸为 $L_m \times C$ 的 memory。



首先为查询添加时间信息。

左侧的第一个 Multi-Head Attention 模块施加在输入的查询上，在其 L_q 个元素之间建立关联。

右侧的第二个 Multi-Head Attention 综合当前查询的 L_q 个元素和原有记忆 L_m 个元素之间的关系，输出 L_q 个查询结果。

实现参见/models/slicenet.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/slicenet.py>)中的 attention 函数。

混合专家模块 Sparsely-Gated MOE

MOE类模块能够在不增加计算量的前提下，构造具有海量参数的模型，大幅提高模型表达能力。细节参看 Google Brain团队的Outrageously Large Neural Networks2。

整个模块包含若干并行的“专家” $E_i(x)$ 。它们的结构相同，参数不同。都是重复若干层的线性网络+激活函数。

实现参见/utis/expert_utils.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utis/expert_utils.py)中 FeedForwardExpert 函数。

另外有一个和输入有关的门函数 G 。其中 $G(x)$ 是一个系数的 n 维向量，如果 $G(x)_i = 0$ ，则不必计算 $E_i(x)$ 。

本文采用的门函数如下：

$$G(x) = Softmax(KeepTopK(H(x), k))$$

$$H(x)_i = (x \cdot W_g)_i + StandardNormal() \cdot Softplus((x \cdot W_{noise})_i)$$

$$KeepTopK(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v. \\ -\infty & \text{otherwise.} \end{cases}$$

其中添加的噪声 StandardNormal 相当于一个平滑项，其强度由 W_{noise} 控制。

实现参见 `/utils/expert_utils.py` (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utils/expert_utils.py) 中 `NoisyTopKGating` 函数。

输出 y_i 所有专家子模块通过门函数加权得到：

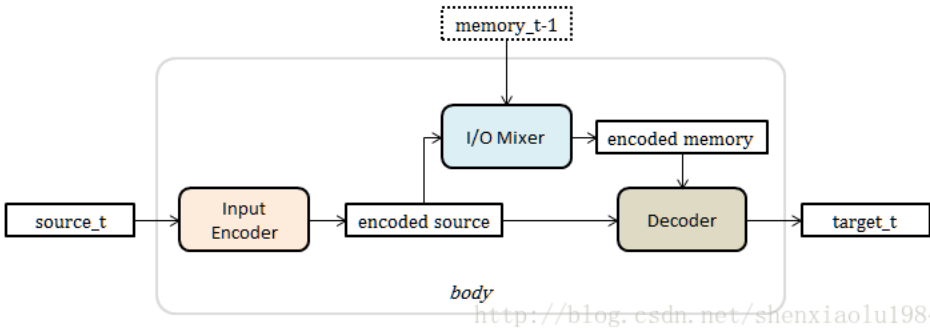
$$y = \sum_{i=1}^n G(x)_i E_i(x)$$

其中 $n = 280/60, k = 4$

实现参见 `/utils/expert_utils.py` (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utils/expert_utils.py) 中 `DistributedMixtureOfExperts` 函数。

body网络-构成

body 网络由如下三部分构成：



系统的输入和输出都是时间序列（非时间信号可以看做长度为1的特例）。

- Encoder 部分处理将 source 编码；
- Mixer 部分将编码后的 source 和系统此刻之前的记忆综合起来，生成编码后的 memory。
- Decoder 部分从编码后的源和记忆生成 target 表达。

Input Encoder

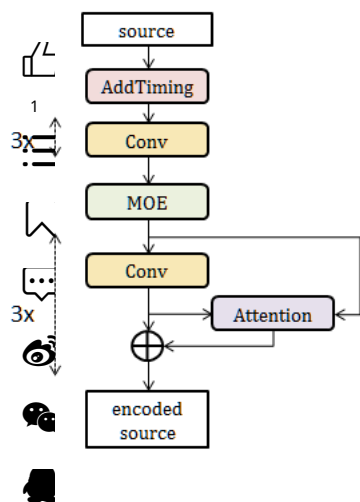
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

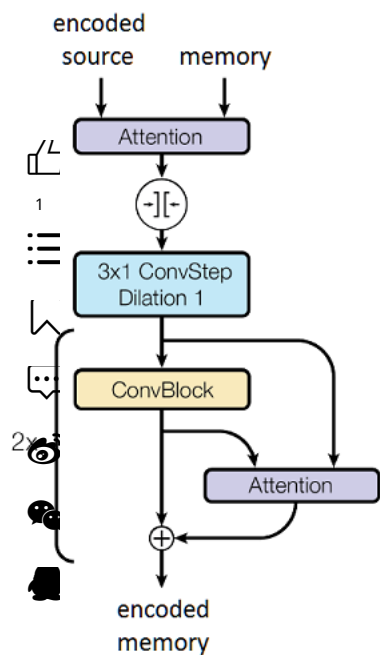
源信号 source 首先添加时间信息，通过3次卷积，并通过一个MOE模块。
之后，与自身重复进行3次 Attention，相当于充分关联输入序列。
最后得到编码后的源信息 encoded source。



实现参见/models/multimodel.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/multimodel.py>), MultiModel 类 model_fn_body_sharded 函数84-106行。源码和论文无法一一对应。

I/O Mixer

首先将编码后的源信息和记忆信息通过 Attention 进行混合。
之后经过concat操作压缩一维。(此处不详)
最后将混合信息通过与 Encoder 类似的2次自身 Attention 操作，获得编码后的记忆 encoded memory。



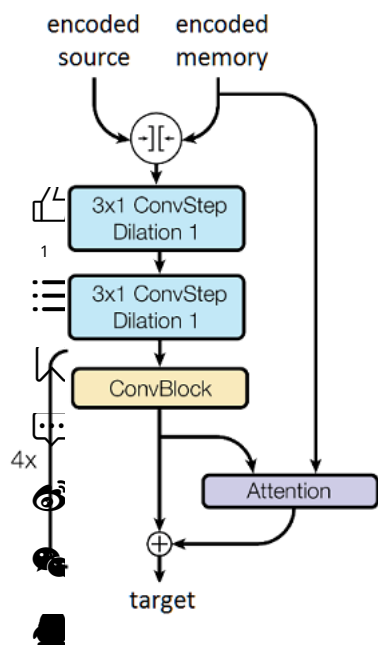
实现参见/models/multimodel.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/multimodel.py>), MultiModel 类 model_fn_body_sharded 函数119-142行。源码和论文无法一一对应。

Decoder

首先将编码后的源信息和记忆信息串接起来。

而后经过两个卷积模块。

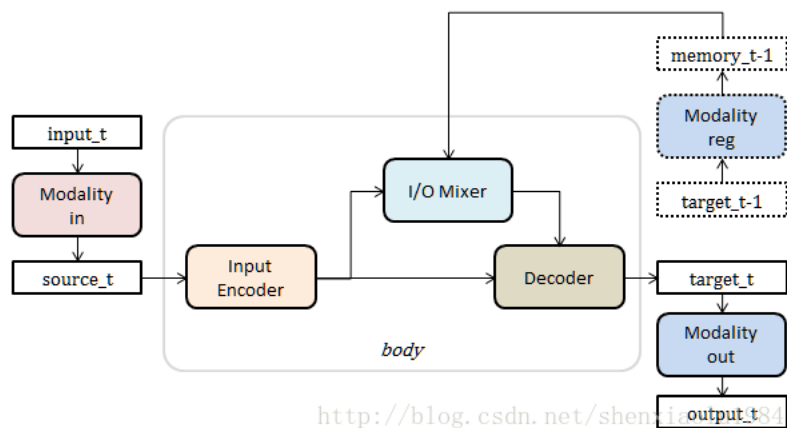
最后通过与 Encoder 类似的4次自身 Attention 操作获得目标信息 target。



实现参见/models/multimodel.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/multimodel.py>), MultiModel 类 model_fn_body_sharded 函数108-116行。
具体实现：/models/slicenet.py (<https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/slicenet.py>)中 slicenet_middle 函数。源码和论文无法一一对应。

modality网络

对于同一形态的信息，modality网络有三种作用：



in 网络：把原始输入 input 转化为源信息 source

out 网络：把目标信息 target 转化为输出 output

regress 网络：把前时刻的目标信息转化成记忆 memory

论文中没有提到 regress 网络，直接用 out + in 代替，但在源码中有所体现。

基类实现参见/utils/modality.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utils/modality.py)中的 Modality 类。bottom，targets_bottom，top 函数分别对应上述三个网络。

不同形态信息的具体实现参见/models/modalities.py (https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/modalities.py)

实验与总结

略去细节，只说值得注意的现象和结论。

- 在没有仔细tune的前提下，本文结论只比state-of-art稍逊色。
- 同时在不同领域的多个任务上训练，几乎不会损害单个任务的精度。
- 对于小数据集任务，同时训练其他任务甚至能够提升本任务的表现。即使是毫不相关形态之下的问题。
- 传统上用于某种形态问题的模块（例如用于语言的attention机制和MOE）能够对其他形态的问题有所帮助。

总体来说，本文在大一统模型的道路上又前进了一步。反观本文的三大设计理念，会发现其更接近人的行为方式：


本文	人类	实例
绝大部分计算量都集中在 body 网络中，modality 网络设计尽量精简。	复杂的思维组件负责处理不考虑形态的抽象概念；简单的输入输出组件负责处理和表达不同形态的具体信号	脑补很强大，眼耳口鼻很粗糙
系统内部的表达尺寸相同，但不固定。	不同复杂程度的抽象概念使用不同长度的信息量来存储。	越常用的概念表达越简单
对于相同形态的不同问题，使用相同的 modality 网络	同类的不同任务使用相同的输入输出组件	用同样的耳朵听不同的语言

本文的源码部分还不完善，有待观望。

1. Vaswani, Ashish, et al. "Attention Is All You Need." arXiv preprint arXiv:1706.03762 (2017). ↩
2. Shazeer, Noam, et al. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." arXiv preprint arXiv:1701.06538 (2017). ↩

版权声明：本文为博主原创文章，未经博主允许不得转载。



 sinat_31188625 (/sinat_31188625) 2017-09-06 15:14

1楼


(/sinat_31188625)问你在总结的表格中，提到的
“系统内部的表达尺寸相同，但不固定。”
是什么意思，在论文中通过什么方式体现。

回复 1条回复



一个模型库学习所有：谷歌开源模块化深度学习系统Tensor2Tensor (<http://blog.csdn.n...>

选自Google.research 机器之心编译参与：黄小天、李泽南在谷歌提交热点论文《Attention Is All You Need》和《One Model To Learn...

 AMDS123 (<http://blog.csdn.net/AMDS123>) 2017年06月20日 11:57 1899



One Model To Learn Them All原文谷歌翻译版本 (<http://blog.csdn.net/jingkebiao4847/ar...>



One Model To Learn Them All Łukasz Kaiser Google Brain lukaszkaizer@google.com Aidan N. Gomez * ...

 jingkebiao4847 (<http://blog.csdn.net/jingkebiao4847>) 2017年11月28日 13:54 87




One Model To Learn Them AI (<http://download.csdn.net/download/lokib...>



<http://download.csdn.net/download/lokib...> 2017年10月29日 12:38 1.16MB [下载\(\)](#)


One function to run them all... Or just eval (<http://blog.csdn.net/u014032673/article/detai...>

When I was writing a sysBio package, I needed a function that can be customized based on the user ...

 u014032673 (<http://blog.csdn.net/u014032673>) 2016年05月17日 09:00 632


[深度学习论文笔记][ICLRW 17] Learning What Data to Learn (<http://blog.csdn.net/u010158659>

这篇文章属于使用Bootstrap提升模型训练性能、加快模型训练速度的研究范畴。相类似的比较出名的工作有Curriculum Learning (ICML 09), self-paced learn...

 u010158659 (<http://blog.csdn.net/u010158659>) 2017年05月08日 17:46 628


[翻译]斯坦福CS 20SI:基于Tensorflow的深度学习研究课程笔记,Lecture note 4: How to ...

"CS 20SI: TensorFlow for Deep Learning Research" Prepared by Chip Huyen Reviewed by Danijar Hafner...

 wanguyuehx (<http://blog.csdn.net/wanguyuehx>) 2017年03月05日 23:24 3144


[深度学习论文笔记][Image Reconstruction] Understanding Deep Image Representation...

Mahendran, Aravindh, and Andrea Vedaldi. "Understanding deep image representations by inverting them..."

 Hao_Zhang_Vision (http://blog.csdn.net/Hao_Zhang_Vision) 2016年10月31日 10:07 1225

【深度学习】聚焦机制DRAM(Deep Recurrent Attention Model)算法详解 (<http://blog.cs...>

Visual Attention基础，Multiple object recognition with visual attention算法解读。

 shenxiaolu1984 (<http://blog.csdn.net/shenxiaolu1984>) 2016年06月28日 22:14 6614

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



登录

注册



泛函编程 (14) — try to map them all (<http://blog.csdn.net/f9db33t79p/article/details/72...>)

虽然明白泛函编程风格中最重要的就是对一个管子里的元素进行操作。这个管子就是这么一个东西： $F[A]$ ，我们说 F 是一个针对元素 A 的高阶类型，其实 F 就是一个装载 A 类型元素的管子， A 类型是相对低阶，或者说是基础...

 f9db33t79p (<http://blog.csdn.net/f9db33t79p>) 2017年05月18日 21:38  42



泛函编程 (14) — try to map them all (http://blog.csdn.net/TIGER_XC/article/details/445...)



虽然明白泛函编程风格中最重要的就是对一个管子里的元素进行操作。这个管子就是这么一个东西： $F[A]$ ，我们说 F 是一个针对元素 A 的高阶类型， F 就是一个装载 A 类型元素的管子。泛函编程风格就是在 F 内部用对付 A 类...

 TIGER_XC (http://blog.csdn.net/TIGER_XC) 2015年03月23日 19:41  480



从头实现一个深度学习对话系统--Seq-to-Seq模型详解 (<http://blog.csdn.net/liuchonge/a...>)



上一篇文章已经介绍了几篇关于Seq-to-Seq模型的论文和应用，这里就主要从具体的模型细节、公式推导、结构图以及变形等几个方向详细介绍一下Seq-to-Seq模型。这里我们主要从下面几个层次来进行介...

 liuchonge (<http://blog.csdn.net/liuchonge>) 2017年12月17日 13:03  196



learn opencv- 深度学习使用Keras - 基础知识 (<http://blog.csdn.net/wc781708249/article/...>)

参考： 1、<https://github.com/spmallick/learnopencv> 2、<https://keras.io> 3、<https://github.com/jolij/Cas...>

 wc781708249 (<http://blog.csdn.net/wc781708249>) 2017年11月09日 19:18  132



Tensorflow实战学习(四十二)【TF.Learn、分布式Estimator、深度学习Estimator】 (<http...>)

TF.Learn，TensorFlow重要模块，各种类型深度学习及流行机器学习算法。TensorFlow官方Scikit Flow项目迁移，谷歌员工Illia Polosukhin、唐源发起。Scik...

 WuLex (<http://blog.csdn.net/WuLex>) 2017年11月22日 11:08  153



learn opencv-Ubuntu(cuda)上安装深度学习框架 (<http://blog.csdn.net/wc781708249/arti...>)

参考：<https://github.com/spmallick/learnopencv>在带有CUDA支持的Ubuntu上安装深度学习框架在本文中，我们将学习如何在具有NVIDIA图形卡的机器上安装...

 wc781708249 (<http://blog.csdn.net/wc781708249>) 2017年11月10日 14:34  141



深度学习常用的Data Set数据集和CNN Model总结 (http://blog.csdn.net/qq_17448289/art...)

这是博主我在刚接触Faster rcnn之后，对CNN的网络模型以及深度学习数据集的归纳整理。...

 qq_17448289 (http://blog.csdn.net/qq_17448289) 2016年10月18日 17:02  6893

【神经网络与深度学习】Caffe Model Zoo许多训练好的caffemodel (<http://blog.csdn.ne...>)

Caffe Model Zoo 许多的研究者和工程师已经创建了Caffe模型，用于不同的任务，使用各种种类的框架和数据。这些模型被学习和应用到许多问题上，从简单的回归到大规模的视觉分类，到Siame...

 LG1259156776 (<http://blog.csdn.net/LG1259156776>) 2016年09月28日 09:23  4097

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



深度学习常用CNN Model总结(alex googlenet等) (<http://blog.csdn.net/wonengguwozai/...>)

登录

注册




原文: http://blog.csdn.net/qq_17448289/article/details/52850223 【数据库】FaceDataset常用的人脸数据库 ...

 wonengguwozai (<http://blog.csdn.net/wonengguwozai>) 2016年12月08日 12:55  782



python机器学习系列教程——深度学习框架比较TensorFlow、Theano、Caffe、SciKit-L...

全栈工程师开发手册（作者：栾鹏） python教程全解 TheanoTheano在深度学习框架中是祖师级的存在。Theano基于Python语言开发的，是一个擅长处理多维数组的库，这一点...

 luanpeng825485697 (<http://blog.csdn.net/luanpeng825485697>) 2017年12月28日 15:01  293



【神经网络与深度学习】如何将别人训练好的model用到自己的数据上 (<http://blog.csdn.net/luanpeng825485697/article/details/78111444>)

caffe团队用imagenet图片进行训练，迭代30多万次，训练出来一个model。这个model将图片分为1000类，应该是目前为止最好的图片分类model了。假设我现在有一些自己的图片想...

 LG1259156776 (<http://blog.csdn.net/LG1259156776>) 2016年09月15日 21:37  5642

深度学习笔记——Attention Model（注意力模型）学习总结 (http://blog.csdn.net/mpk_no1/article/details/78111444)

Attention Model（注意力模型）学习总结，包括soft Attention Model，Global Attention Model和Local Attention Model，静态AM，...

 mpk_no1 (http://blog.csdn.net/mpk_no1) 2017年08月06日 21:49  8559