CSCI 736 Neural Network

[toc]

Paper Time Line

Write 1 and 5

Paper Content	Time Schedule

Paper Math Equation

Learning Resource

李宏毅

文章理解

embedding(八 种 常 用 的　embedding)->rnn->lstm&gru->attention->seq2seq->self-attention->transformer->bert

embedding

RNN

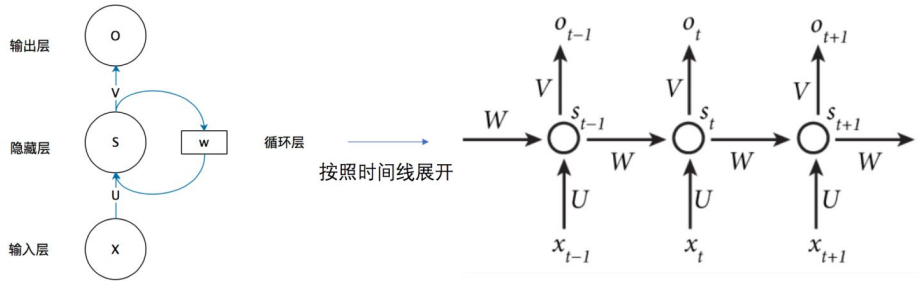


Figure 1: v2-b0175ebd3419f9a11a3d0d8b00e28675_1440w

LSTM

GRU

attention

参数少, 速度快, 效果好

优质教程

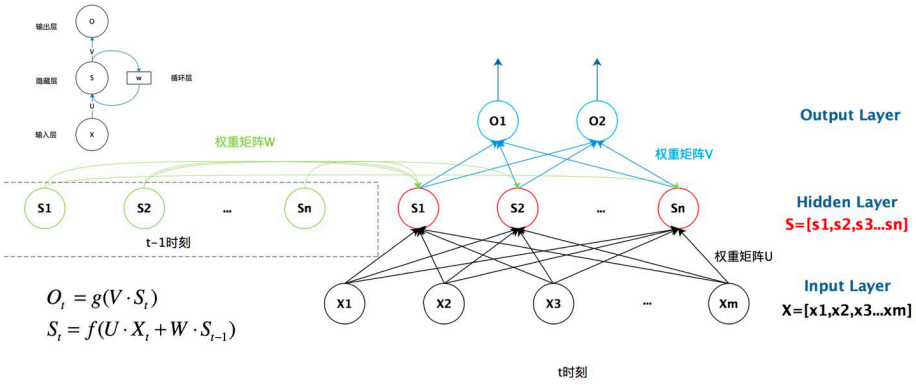


Figure 2: v2-9e50e23bd3dff0d91b0198d0e6b6429a_1440w

Encoder&Decoder

一类算法的统称

这类算法的统称:

1. 无论输入和输出的长度是什么, 中间的 向量 c 长度固定
2. 根据不同任务可以选择不同的编码器和解码器

缺点: 当输入信息太长时，会丢失掉一些信息.

seq2seq $\in Encoder\&Deconder$

一类算法的统称

这类算法的统称: 满足输入序列, 输出序列的目的

self-attention

transformer

bert

linear classifier: two vector

each vector dot product the embedding, then apply softmax, find maximum to get index

Deep Auto-encoder

Paper:Deep Auto-Encoder Neural Networks in Reinforcement Learning

PPT:Unsupervised Learning-Auto-encoder

Stating from PCA

可以把 PCA 的前半部分视为 encode, 后半部分 decode

重要特性: 增强 robust

对 CNN 建立 decoder, decoding 的过程实际还是在卷积

图一　图二

这一页 ppt 只是简单讲了下如何世纪一个 discrminator 来保证 auto-encoder decoder 效果好 (decoder 尽可能把 vector 还原为原始图像)

所以这启发我们: 我们需要一个足够好的能够衡量 encoder 与 decoder 的 discriminator/classifier 来监督 encoder 与 decoder 的训练与一个足够好的 encoder 来保证 vector 与原始图像能尽量一一对应 (每个原始数据尽量能有独一无二的 embedding)

参考文章: Deep InfoMax (DIM)

若训练集是 sequential, skip thought

skip thought->quick thought https://arxiv.org/pdf/1803.02893.pdf

quick thought 只认 encoder 不管 decoder, 每一个句子的 ebedding 跟他下一个句子的 embedding 越接近越好, 跟随机的句子的 emberdding 越不同越好

quick 设计的 classifier: 输入句子 A 应用 encoder 产生的 embedding, 句子 A 的下一句应用 encoder 产生的 embedding, 一对随机句子应用 encoder 产生的 embedding, classifier 需要能够认为句子 A 的下一句巨句子 A 的相似度最高

这样的 classifier 与产生这个 embedding 的 encoder 同时训练

Feature Disentangle

- -

假设 encoder 返回的前 100 个 embedding 放入 speaker classifier 中进行训练, 直到 speaker classifier 无法区分出那种音色, 这时候就认为前 100 已经没有了音色信息, 音色信息跑到了后 100 个中

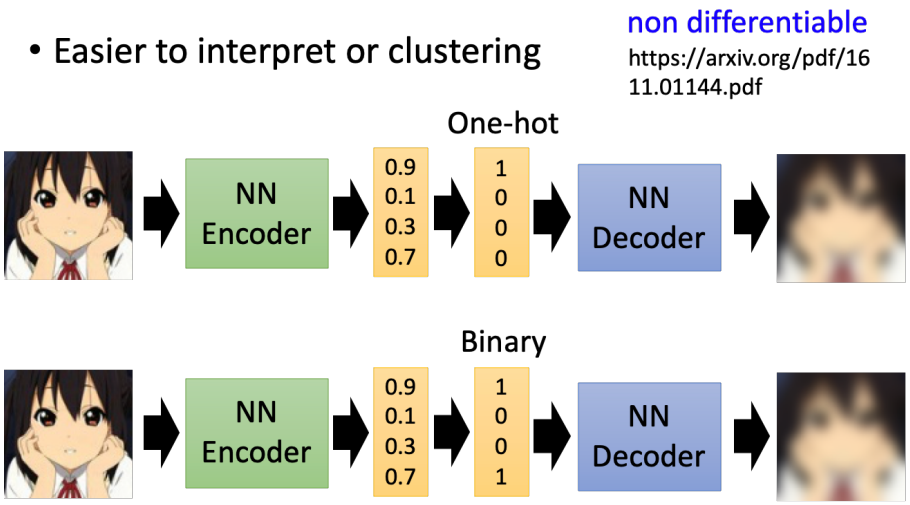
Instance normalization: 一种特逼得 layer, 可以抹掉不想要的信息

比如全部抹掉音色信息, 那么剩下的就是纯正的语音信息, 具体方案依赖 Gan 实现

Vector Quantized Variational Auto-encoder (VQVAE)

-

-



返回的 vector 对其内容做 one-hot(最大的变 1 其余 0) 转换或者 binary 转换 (设定 threshold, 大于的变为 1 其余 0), 推荐 binary, 这样可以意外的发现训练集中原本不存在的 cluster

假设 codebook 中只有 5 个 vector, 则 encoding 返回的 vector 与这五个做相似度比较, 与 codebook 中哪一个相似就把 codebook 中的哪个返回给 decoder

seq2seq2seq

Gan

generator+ranfdom vector-> target high dimensional vector

discriminator: 像二次元则高分, 否则低分

不仅训练 gnererator 还需要 discriminator

Conditional Gan (Supervised)

generator 有可能会发现某一种一旦可以骗过 discriminator 后, 就不再改变自己, 无视输入, 时钟输出同一个东西来欺骗 discirminator

现在我们修改 discirminator, 不仅判断 generator 的结果有多好, 还判定 generator 的输入与输出有多匹配

1. text->image

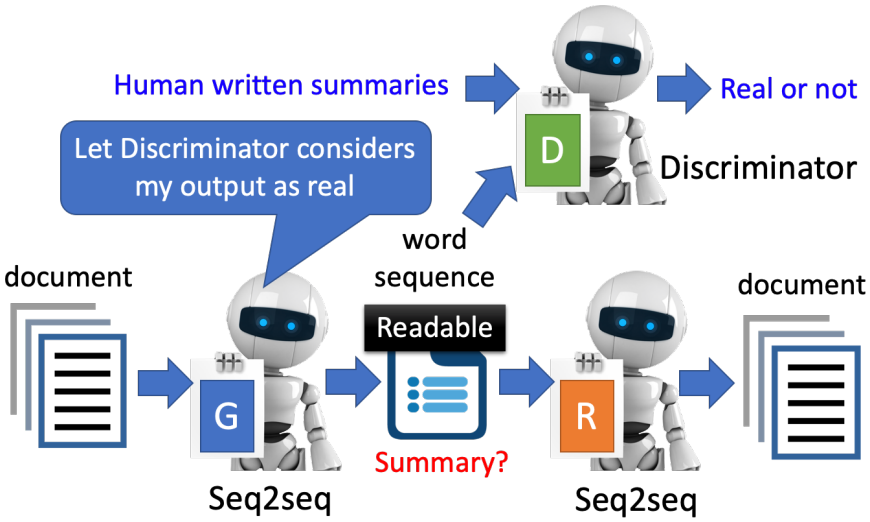


Figure 3: image-20210310002011618

好图片 + 好 text=1

好图片 + 烂 text=0= 兰图片 + 好 text

2. sound to image

e.g. 电视雪花声-> 瀑布, 声音越大, 瀑布越猛

类似直升机的声音-> 快艇海上行走, 声音越大, 快艇引起的水花越大

3. image->text

   e.g. image-> multi label

Unsupervised Gan Cycle Gan

- P72Generative Adversarial Network(1_10)

  1:33:15

- P73Generative Adversarial Network(2_10)

  26:19

- P74Generative Adversarial Network(3_10)

  38:59

- P75Generative Adversarial Network(4_10)

  1:20:19

- P76Generative Adversarial Network(5_10)

  25:04

- P77Generative Adversarial Network(6_10)

  50:07

- P78Generative Adversarial Network(7_10)

  46:03

- P79Generative Adversarial Network(8_10)

  22:46

- P80Generative Adversarial Network(9_10)

  1:27:23

- P81Generative Adversarial Network(10_10)

  30:09

Presentation

1. 自我介绍, 标题页
2. 第二页!!

   the thoughts of our algorithm is that you input the question and document and it return subspan of the documents as the answer.

   Here is an example of the model

3. As we choose bert as baseline, we inputs tokens and bert return the answer's start and end index.

   So let's assume that our question has n tokens and documents have m tokens, we inputs their concatenation into bert.

   最后一页!!!

   The bert will return vector C which has the same dismension as inputs but we only need the document part, because the answer is just the sub span of input document.

   As we need to find the start and end index, we prepare two linear classifiers

   We use them take dot product with C's document part and apply softmax to get p{start,i }and p{end,j} probabilty distribution

   Assume the correct start and end index named "I" and "J", then we could get the p_I from p_startdistri, p_J from p_end_j distri ,then the loss fucntion for this sample is

   should have correspond equation

Next, we apply backpropagation.

we reapeat these processes until finish all training part.

Beyond the basic QA system, our goal is to make it robust on unseen domain, to achieve such goal,we will try some existing strategies which will be discussed in related work and see if we can make any improvements.

CLS: the key word of the classifier

SEP: separate question tokens and document tokens

Proposal Feedback