Lecture 7 自监督学习

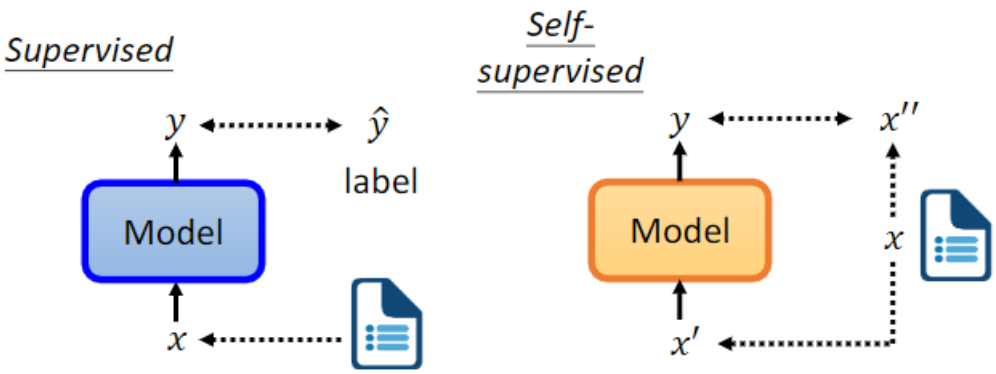
> Lectured by HUNG-YI LEE (李宏毅)
> Recorded by Yusheng zhao（yszhao0717@gmail.com）

> Self-Supervised Learning的许多模型都以芝麻街的人物命名。
>
> - ELMo（Embedding from Language Models）
> - BERT（Bidirectional Encoder Representation from Transformers）
> - ERNIE（Enhanced Representation through Knowledge Integration）
> - Big Bird（Transformers for Longer Sequence）
> - Cookie Monster

什么是self-supervised learning（自监督学习）

Supervised：需要成对的数据，资料（文章/图像）和labels

Self-supervised：资料没有labels，想办法把资料x分为两部分，一部分x'作为模型的输入，另一部分x''作为模型的标注。把x'输入到模型中，输出y，再与x''进行比对。*self-supervised*的方法可以看作是*unsupervised*的（对应的超集）

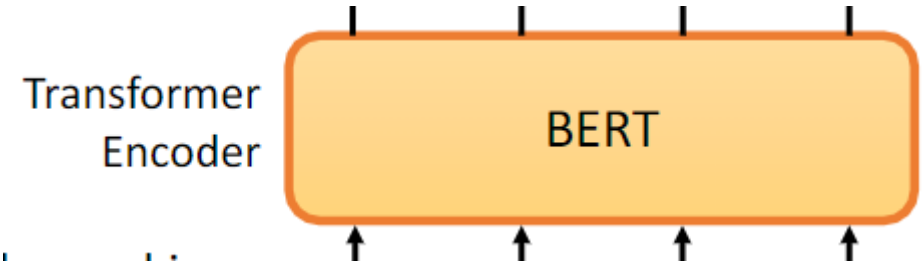


> “In self-supervised learning, the system learns to predict part of its input **from other parts of it input**. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.”
>
> ——By Yann LeCun 2019.4.30, facebook

BERT

> Bert很大，有340M个参数（parameters）
>
> Bert科普文：進擊的 BERT：NLP 界的巨人之力與遷移學習
>
> 目前的趋势：模型的规模越来越大，参数越来越多
>
> ELMo（94M）→ BERT（340M）→ GPT-2（1542M）→ Megatron（8B）→ T5（11B）→ Turing NLG（17B）→ **GPT-3**（比Turing NLG大10倍！！）→ Switch Transformers（1.6T）

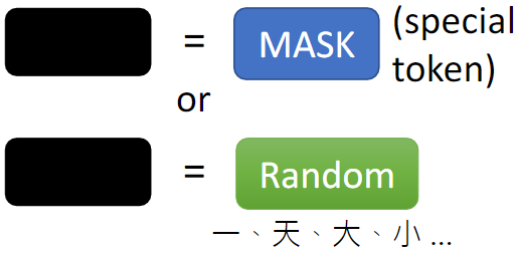
BERT是一个“Transformer-Encoder”的形式



能做的事情就是输入一排向量然后输出一排向量，BERT通常使用在NLP任务以及文字处理等等。文字、语音、甚至是图像都可以看作是一个sequence

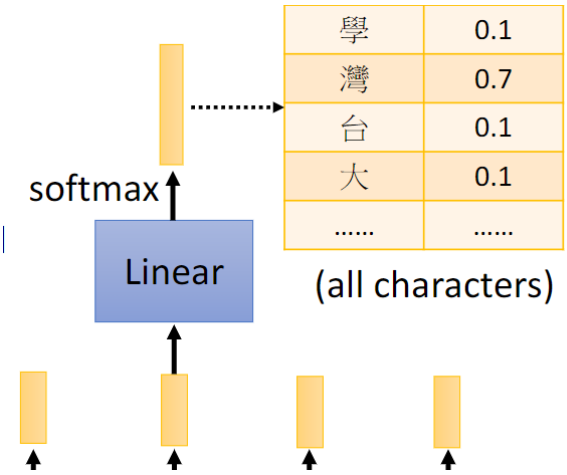
Masking Input（Masked token prediction）

对于BERT的输入，以一串文字为例，随机把其中一些的character掩盖住（Randomly masking some tokens）。所谓的"盖住"有两种方式：1、把某些字换成特殊的符号（MASK【special token】）；2、随机把把某个字换成另外的字（Random）



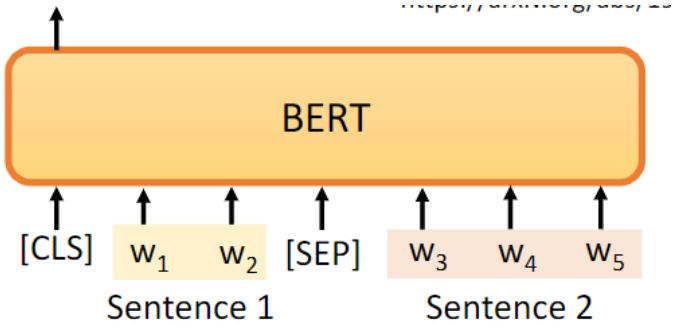
> token是处理一段文字的单位，它的尺度和大小由自己决定（比如中文中就是一个汉字（character），在英文单词中就是一个字母（1/26））

盖住部分所对应的输出向量，做一个Linear的transform（乘一个矩阵），然后再做一个$softmax$，得到一个输出，输出一个分布——预测任务。（这部分和Transformer是差不多的）



BERT学习的目标就是掩盖住的token要学习的输出（预测）应该和对应的ground truth越接近越好，近似于做一个分类任务（class数量和token数量一致）。所以最后的步骤中涉及到了minimize cross entropy

Next Sentence Prediction



[SEP]是一个分隔符，[CLS]是一个特别的符号。把整个句子输入进BERT里面。[CLS]的输出做一个Linear的transform，做一个二元判断问题（Yes/No）——判断这两个句子是否相接（Yes是/No否）。

这个方法似乎帮助不大。在一篇RoBERTa论文论证了这点。

另外一招（文献上似乎比Next Sentence Prediction有用）：SOP: Sentence order predictionUsed in ALBERT。它的方法：把两个sentence连一块（调换顺序），让BERT判断哪一个最佳。

Pre-training

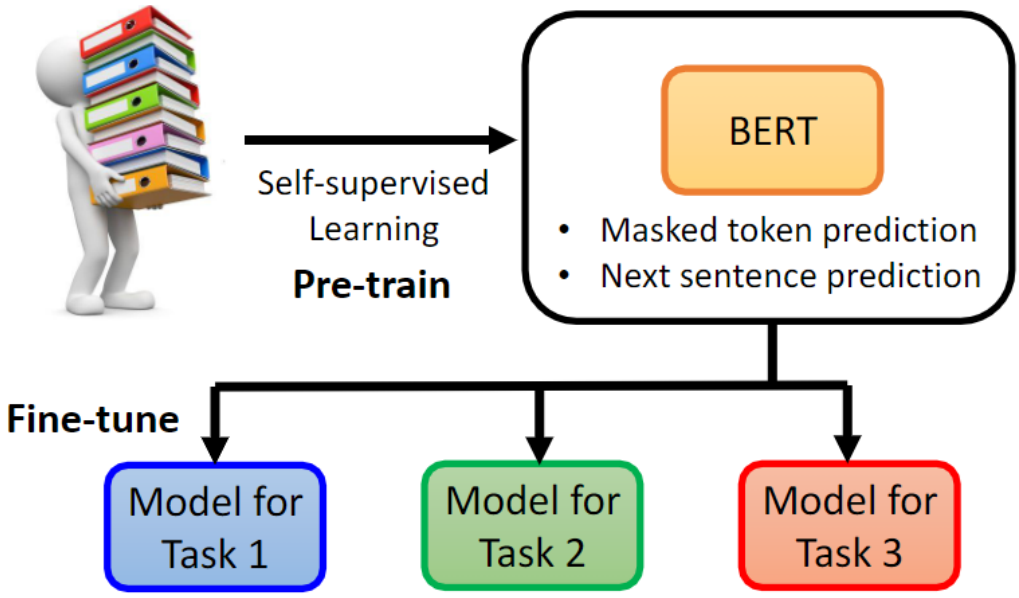
上述的BERT主要讲了两方面的应用：

- 填空题：随机盖住（masked）的输入token**预测**对应输出

  > E.g. Birds can ___（ground truth：fly）

- 判断题：两个句子（sentence）是否相连

除此之外，BERT可以其他任务上（*Downstream task*）



奇妙地比喻：像是胚胎里的干细胞，给一点特别的刺激（有标注的资料），就可以"分化"显著的完成各式各样的任务，**BERT分化成各式各样的任务**被称之为**Fine-tune**。而在Fine-tune之前，产生BERT的过程被称之为**Pre-train**。

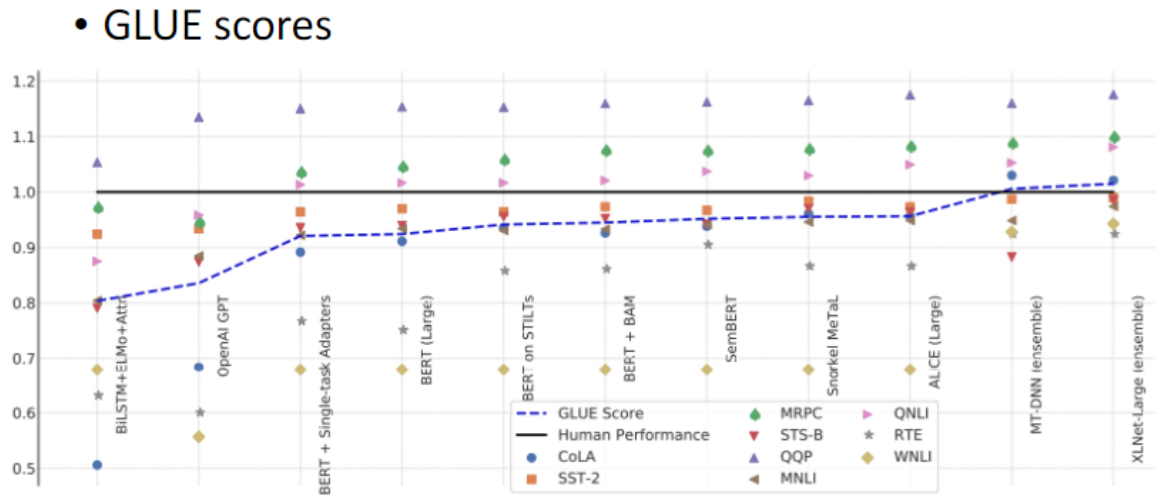
测试BERT能力的任务集：GLUE

> 标杆：General Language Understanding Evaluation (GLUE)https://gluebenchmark.com/GLUE
>
> GLUE also has Chinese version (https://www.cluebenchmarks.com/)

GLUE总共有九个任务，为了测试BERT模型的能力，拢共建立九个模型，每个模型测试完平均一下得到一个数值——代表了self-supervised model的好坏。

- Corpus of Linguistic Acceptability (CoLA)
- Stanford Sentiment Treebank (SST-2)
- Microsoft Research Paraphrase Corpus (MRPC)
- QuoraQuestion Pairs (QQP)
- Semantic Textual Similarity Benchmark (STS-B)
- Multi-Genre Natural Language Inference (MNLI)
- Question-answering NLI (QNLI)
- Recognizing Textual Entailment (RTE)
- WinogradNLI (WNLI)

E.g.

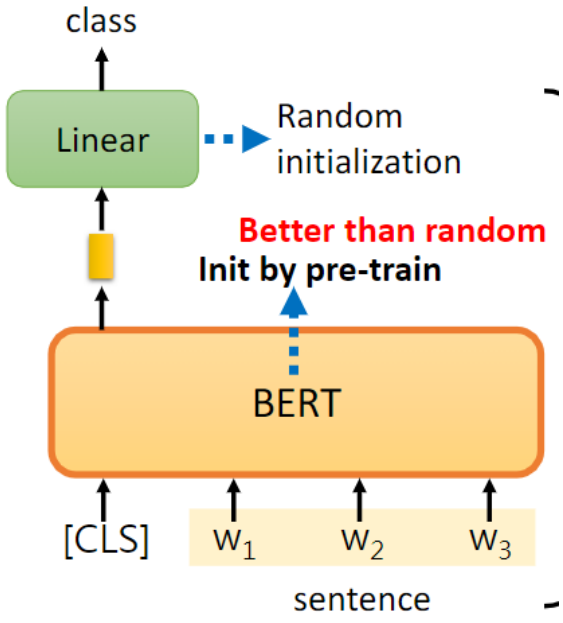


How to use BERT

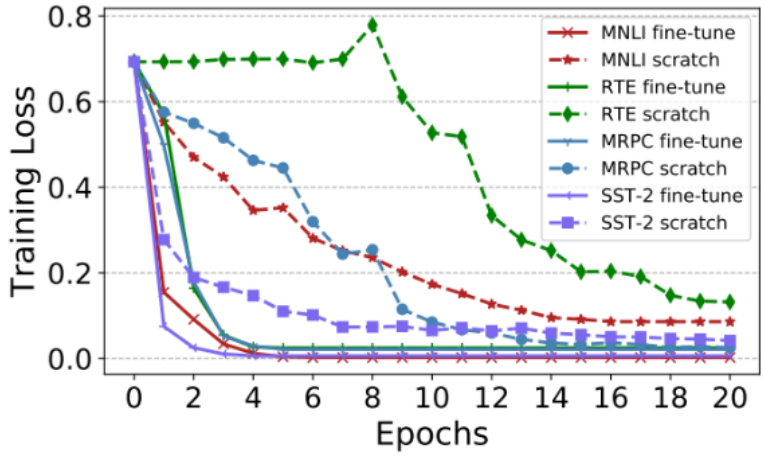
case 1

> Input：sequence
>
> output：class
>
> 例如：情感分析（Sentiment Analysis）....., So I am ___

Linear的参数是随机初始化的，而BERT的参数是来自已经学会了做填空题的模型。（Pre-training better than initiation randomly！！！）



> ### Pre-training v.s. Random Initiation
>
> （fine-tune）---（scratch），下图来自https://arxiv.org/abs/1908.05620



由上图，scratch的loss曲面下降的比较慢。

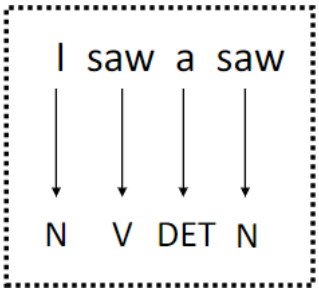
我们可以认为这时候的（带有pre-training）BERT既是unsupervised的也是semi-supervised。当BERT是学做填空题的阶段时是unsupervised的，而当BERT用在下游任务（downstream tasks）上时，由于存在大量无标注资料且存在少量有标注资料，则属于semi-supervised，上述pre-training+fine tune合起来就是semi-supervised。

case 2

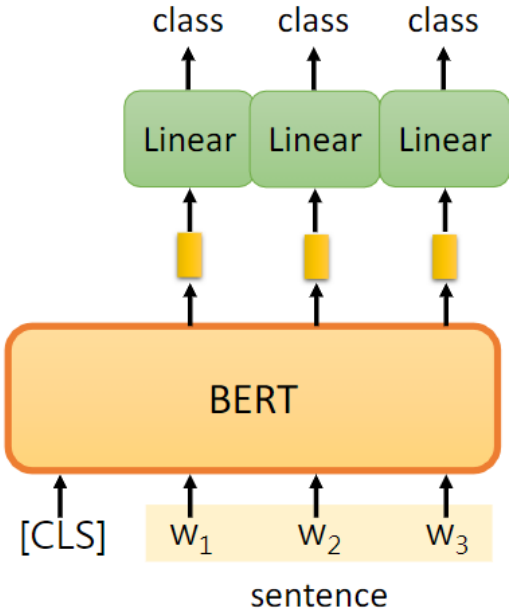
Input：sequence

output：same as input（输入和输出长度是一样的）

E.g. 词性标注（POS tagging）



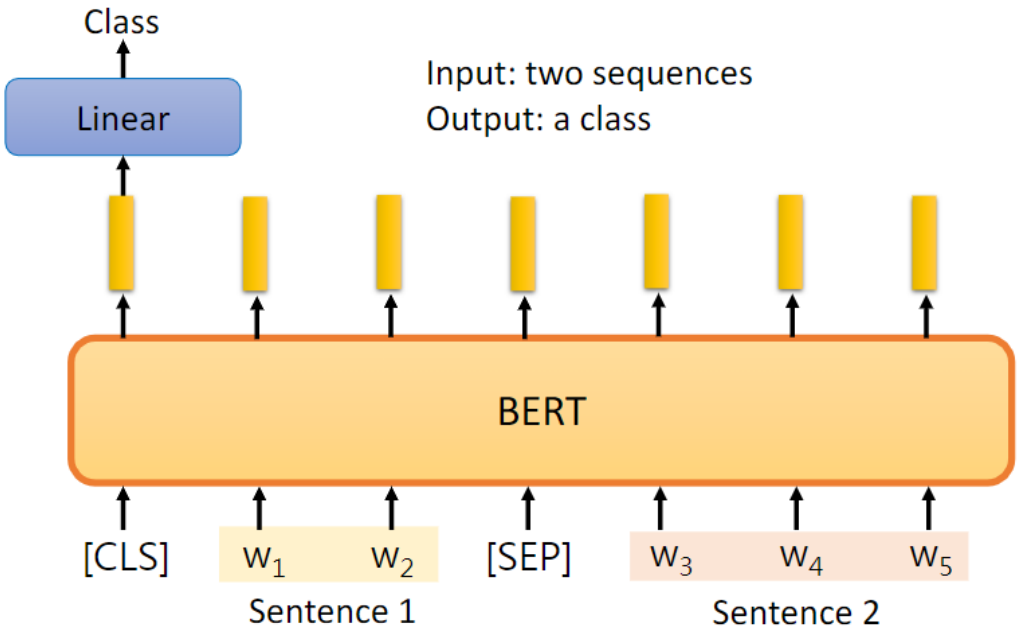
给BERT输入一个句子，句子里成分（token）对应的每一个向量，分别做一个Linear的transform，再过$softmax$，最后分类到一个class。和一般的分类问题相同：我们仍需要一些已有标注的资料——唯一不同的是在BERT的encoder部分其参数不是随机初始化的，而是从pre-train（找到一组表现较好的参数）中继承而来。



上述例子都是文字任务，事实上我们也可以用在语音任务、影像任务等，把影响或语音看作是一排向量

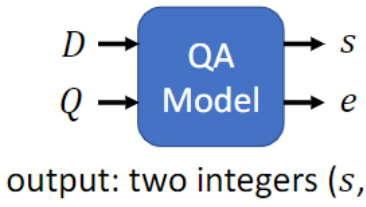
case 3

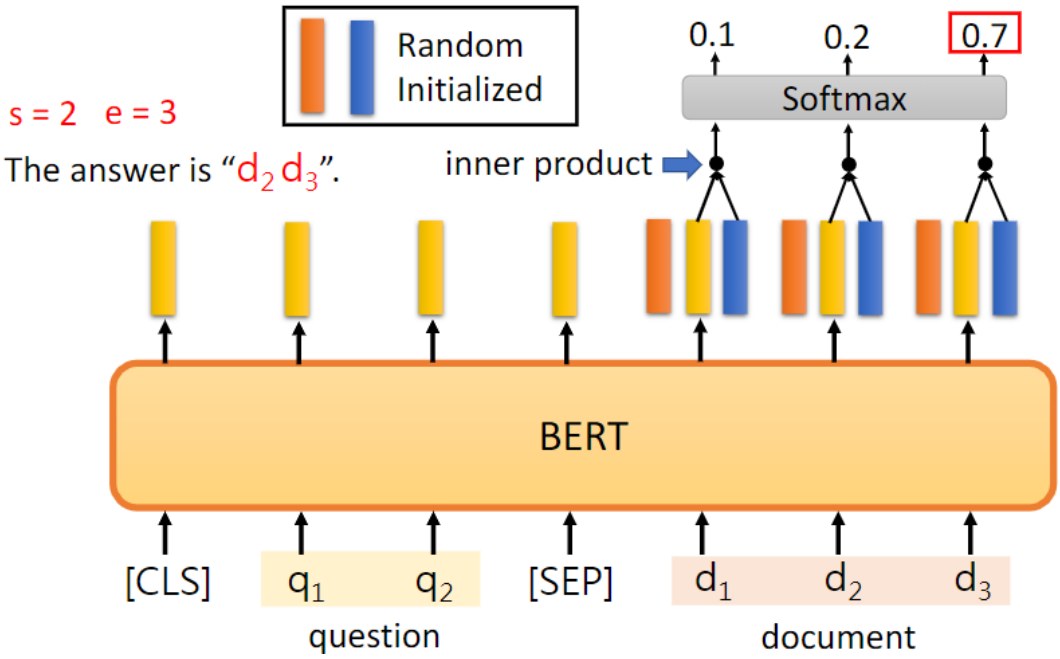
> Input：two sequences
>
> Output：a class（输入两个句子，输出一个类别）
>
> E.g. Natural Language Inferencee：
>
> 给机器一个（已知的）前提（premise）和假设（hypothesis）；让model判断这个前提和假设是否相符（吐出这两个句子的关系）。譬如在立场分析（谁赞成？谁反对？contradiction、entailment、neutral）应用。



只取CLS[Classification]的部分的代表向量，做transform-->$softmax$-->分类，判断这两个句子是否矛盾。

case 4

> 非开放的问答系统：Extraction-based Question Answering（QA）——问题的答案包含在文章中
>
> ***Document***:$D = \{d_1, d_2, \ldots, d_N\}$ ***Query***: $Q = \{q_1, q_2, \ldots, q_M\}$
>
> 
>
> output: two integers (s, e)
>
> 输出两个正整数，代表答案所在的字符index范围（s:start；e:end）
>
> ***Answer***: $A = \{d_s, \ldots, d_e\}$



document：读文章；question：看问题。[CLS]和[SEP]的token和一般的BERT一样

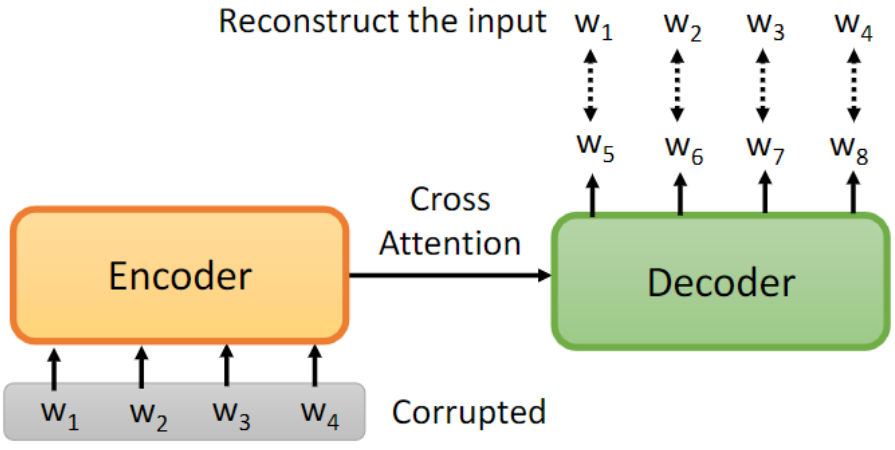
需要随机初始化只有两个向量，这两个向量的长度和BERT输出的长度是一样的，其中橘色代表答案开始的位置，蓝色代表答案结束的位置。先把橘色的拿出来和文章对应的单位（token）做一个inner product；算出数值-->过$softmax$看哪里分数最高，那么s（起始位置）就是这个位置的编号；同理，蓝色的向量也拿出来和文章对应的单位（token）做一个inner product；算出数值-->过$softmax$看哪里分数最高，那么e（终止位置）就是这个位置的编号。

为了训练这个模型，我们也需要训练资料。BERT理论上没有输入长度的限制，实作上有限制，所以需要把整篇文章拆成小部分分别做任务。

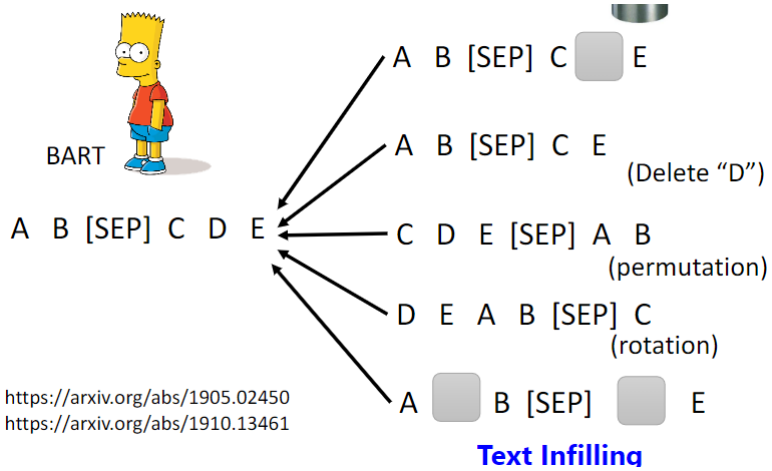
重头训练BERT往往是大公司才能做，即便是高校的资源也很难训动。

BERT Embryology

Pre-training a seq2seq model



为了增强鲁棒性，给Encoder输入的句子做一些"污染"，要求model还原输入产生输出。参考的工作：MASS以及BART

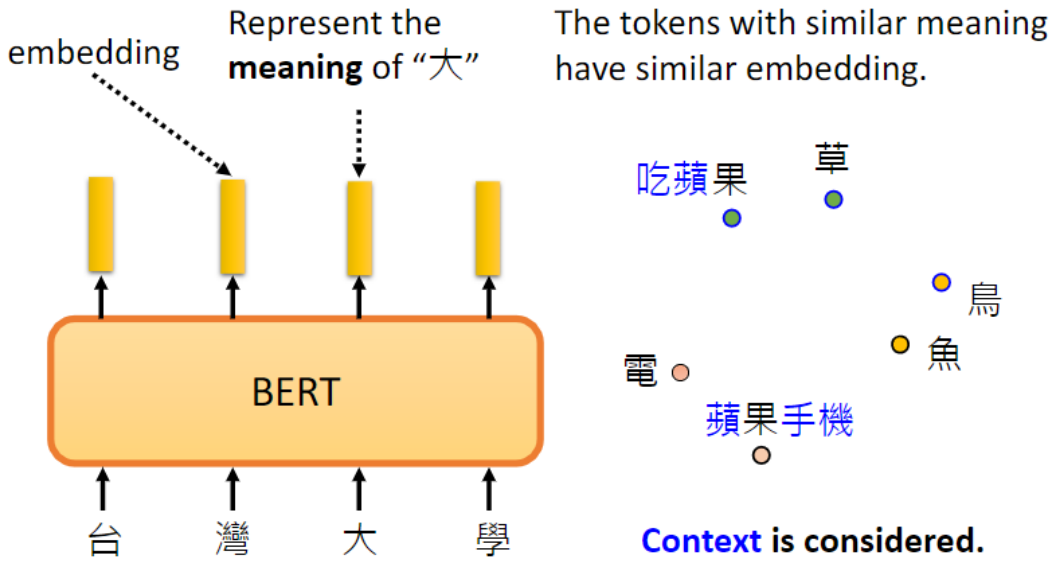


在BERT上做一些改良的工作Google做了完善的整理

- Transfer Text-to-Text Transformer (T5)
- Colossal Clean Crawled Corpus (C4)　——公开的数据集

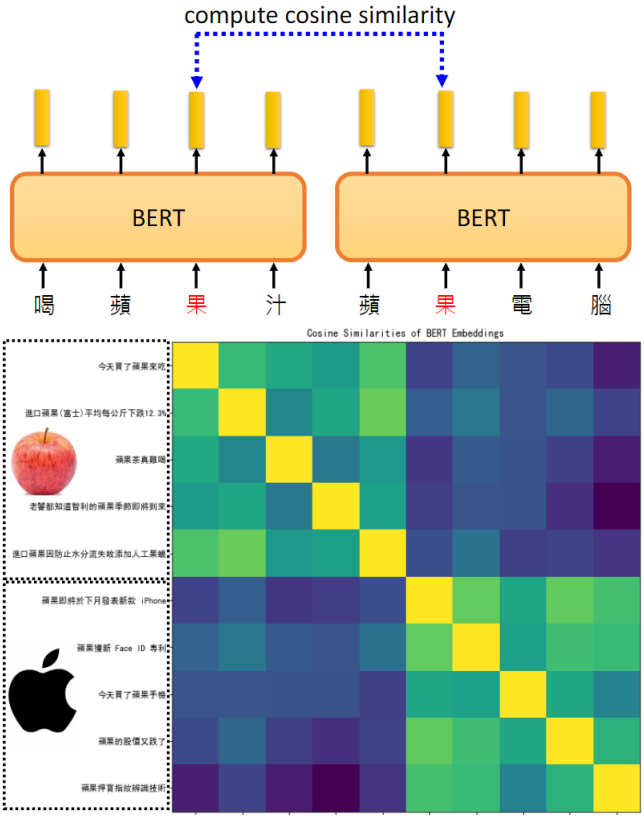
Why does BERT work?

以文字处理为例，以一排文字输入BERT，产生出一排向量（称之为**embedding**），每个这样的向量代表着对应的文字序列的单位token。如果我们把每个embedding向量计算两两之间的距离，我们会发现意思越相近的字（token）代表的embedding距离越小（越靠近）

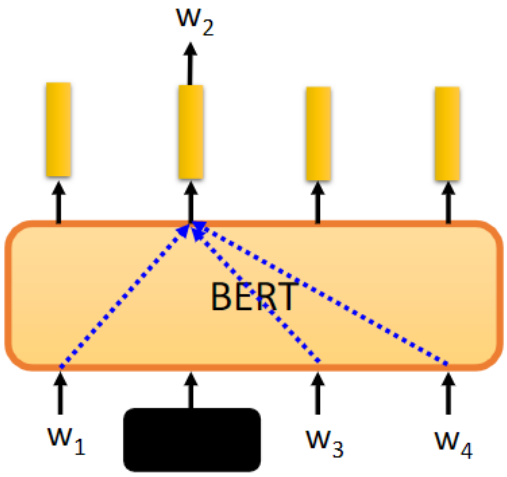


考虑到”一词（字）多义“问题，由于BERT会考虑到上下文，所以同一个字（词）在不同的上下文中的embedding都是不一样的。

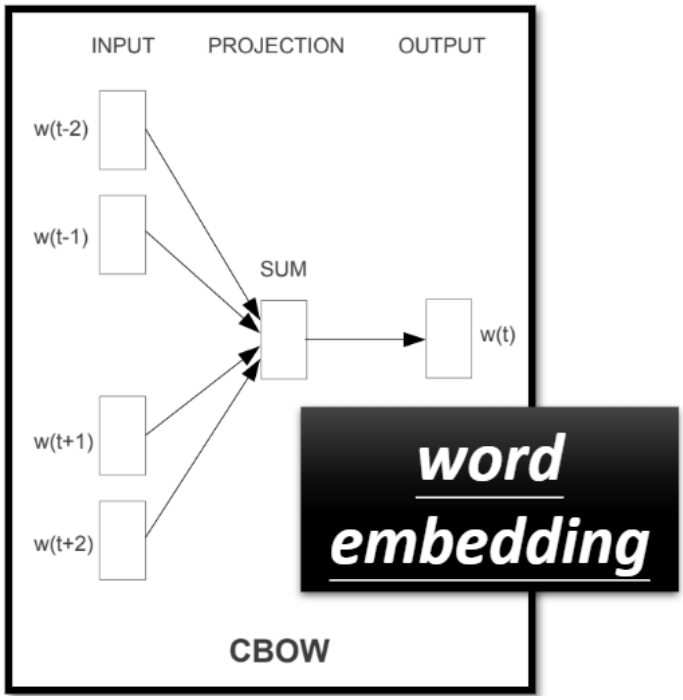
以”果“为例，我们需要收集很多包含这个字的句子，然后把这些句子都丢进BERT里面，各自得到自己的**embedding vector**，然后计算关于这个字的vector集的相互之间的**cosine similarity（相似度）** 👇



相似度图中，偏黄色值较大（表明相似度较大），所以BERT处理后的embedding vector相似度表达了原有输入字符（含义）的相似度。一个词汇的意思可以从上下文推断出来，而BERT所做的事情就是抽取每个token上下文的资讯：举个栗子——如下图，把w_2掩盖起来，让BERT完成预测w_2的任务，而依靠的资讯就是被掩盖的w_2一定范围内上下文的信息。



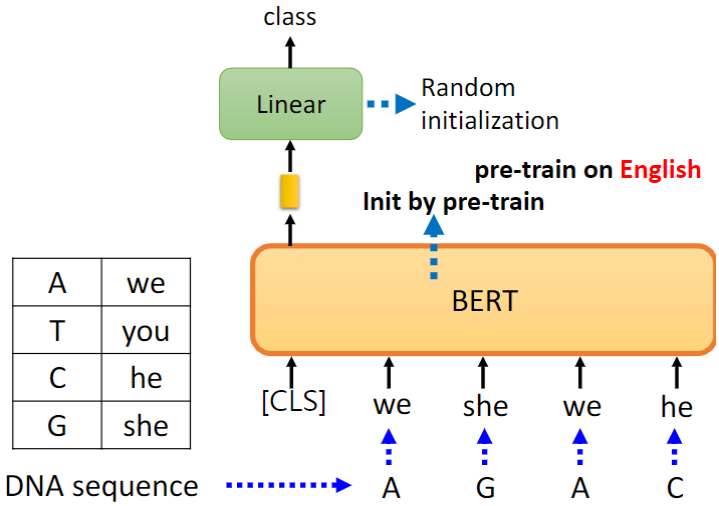
我们可以认为BERT就是一个self-attention的集合体，通过训练得到好的参数后，就可以用上下文来表示词单位的信息，这个就是**representation（文本表示）**。这样的想法，在BERT之前就有了——**Word embedding（词嵌入）**中的一个技术**CBOW**



> Word2vec 是 Word Embedding 方式之一（将文本转换成可计算的向量）。其中一个模型称之为连续词袋模型（continues bag of words，CBOW）。这是一个非常简单的模型，就用了两个transform的一个linear model。BERT其实就是deep learning版的CBOW。lol

所以这个想法认为BERT是CBOW的一个“进阶版”，因此BERT从文本信息中抽取出来的向量（embeddings）又称之为**Contextualized word embedding。**

> 另外一个关于BERT的想法来自于李宏毅老师介绍学生的一项“莫名其妙”的工作：https://arxiv.org/abs/2103.07162
>
> 这个工作介绍一个把BERT应用在蛋白质、DNA的分类任务上。由于DNA由脱氧核苷酸（A、G、C、T）双螺旋组成，把AGCT分别对应到任意的四个英语词汇，将这个句子sequence输入进BERT，然后如上述我们讲的做一个文本分类任务（有木有感觉xjb做？？？），然而得到了很好的实验结果。



实验结果👇

• Applying BERT to **protein, DNA, music classification**

	Protein			DNA				Music
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice	composer
specific	69.0	76.0	63.0	87.3	87.3	79.1	94.1	-
BERT	64.8	74.5	63.7	83.0	86.2	78.3	97.5	55.2
re-emb	63.3	75.4	37.3	78.5	83.7	76.3	95.6	55.2
rand	58.6	65.8	27.5	75.6	66.5	72.8	95	36

BERT到底为什么会好？这里面有很多值得研究探讨的问题。这里面给与BERT完全乱七八糟的文字（譬如DNA所映射的），但是BERT却得到比较不错的分类结果，说明BERT不单单是能够对于文字的含义有一定理解，还有其他因素存在。

还有许许多多模型莫名其妙的work了...但是为什么？还需要我们追寻...

Multi-lingual BERT（多语言BERT）

> ——Training a BERT model by many different languages
>
> Training on the sentences of 104 languages 这玩意儿会做104种语言的填空题。

更神奇的是，如果我们拿英语问题（QA）做训练集，它就可以在中文问题（QA）的测试集上表现优异。

- English: SQuAD, Chinese: DRCD

Model	Pre-train	Fine-tune	Test	EM	F1
QANet	none	Chinese	Chinese	66.1	78.1
BERT	Chinese	Chinese		82.0	89.1
	104 languages	Chinese		81.2	88.7
		English		63.3	78.8
		Chinese + English		82.6	90.1

F1 score of Human performance is 93.30%

上述难道是曲线机器学习吗？？教BERT英文填空，然后就会做中文题（裸考）？？——Cross-lingual Alignment

有一种解释就是：对于lingual BERT来说，不同语言对于相同词（或者词汇意思相似）的embedding很接近，所以处理来说就达到了这样一个神奇的结果。

> Mean Reciprocal Rank（MRR）用来评估不同语言的对齐程度。Higher MRR，better alignment（对齐）

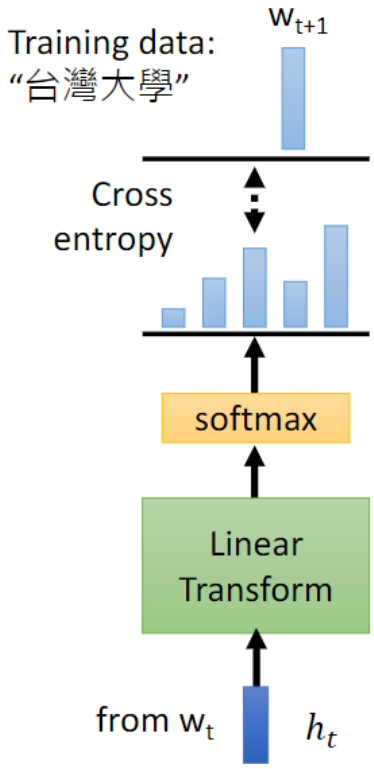
只要数据多，算力够，大力出奇迹！

一个可能是解决Unsupervised token-level translation的思路。具体见老师PPT和学生文章：https://arxiv.org/abs/2010.10041

GPT

> 和BERT不同，GPT的任务是预测接下来会出现的Token（Predict Next Token）

E.g.给GPT model一个token，然后模型处理得到一个embedding记为h，然后模型用这个embedding来预测下一个token是什么



通过$softmax$得到一个distribution，然后做交叉熵。

GPT具备“生成”的能力；比方说，输入一个残缺的句子/文章，让富有“想象力”的GPT把其余的部分补完。GPT用的想法和BERT不太一样，但是它也可以用和BERT一样的方式。

经过pre-train后的BERT，我们使用者只需要做一些微调（fine-tune）即可。

Few-shot Learning：少样本学习

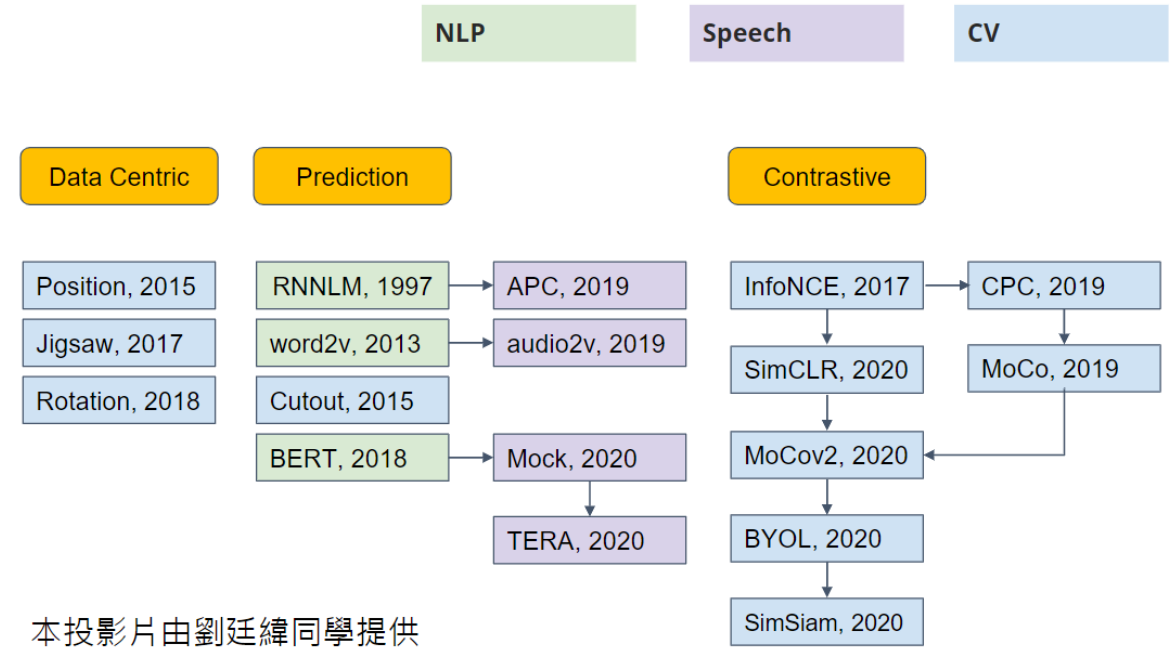
> no gradient descent

One-shot Learning

Zero-shot Learning

Self-supervised Learning for application beyond Text

> 流水账式讲述下相关内容，如果感兴趣自行了解细节



本投影片由劉廷緯同學提供

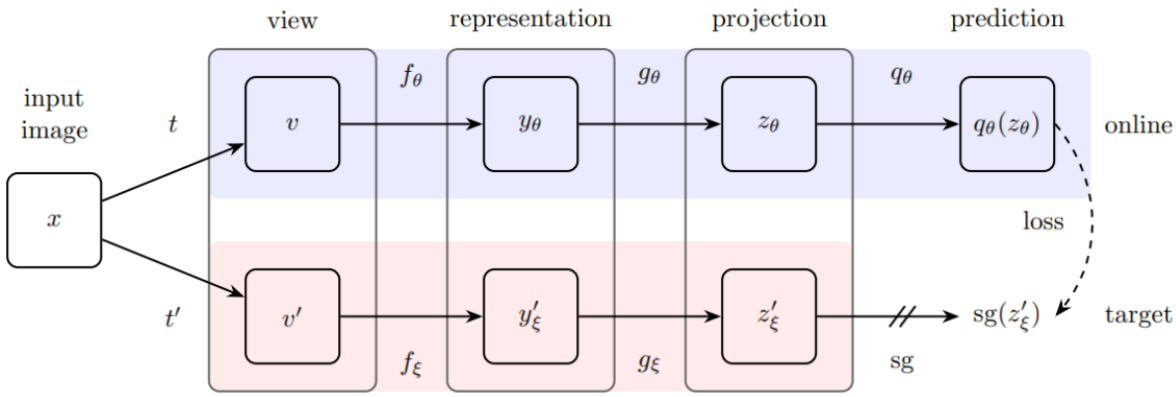
- SimCLR

  > https://arxiv.org/abs/2002.05709
  >
  > https://github.com/google-research/simclr

- BYOL

  Bootstrap your own latent: A new approach to self-supervised Learning

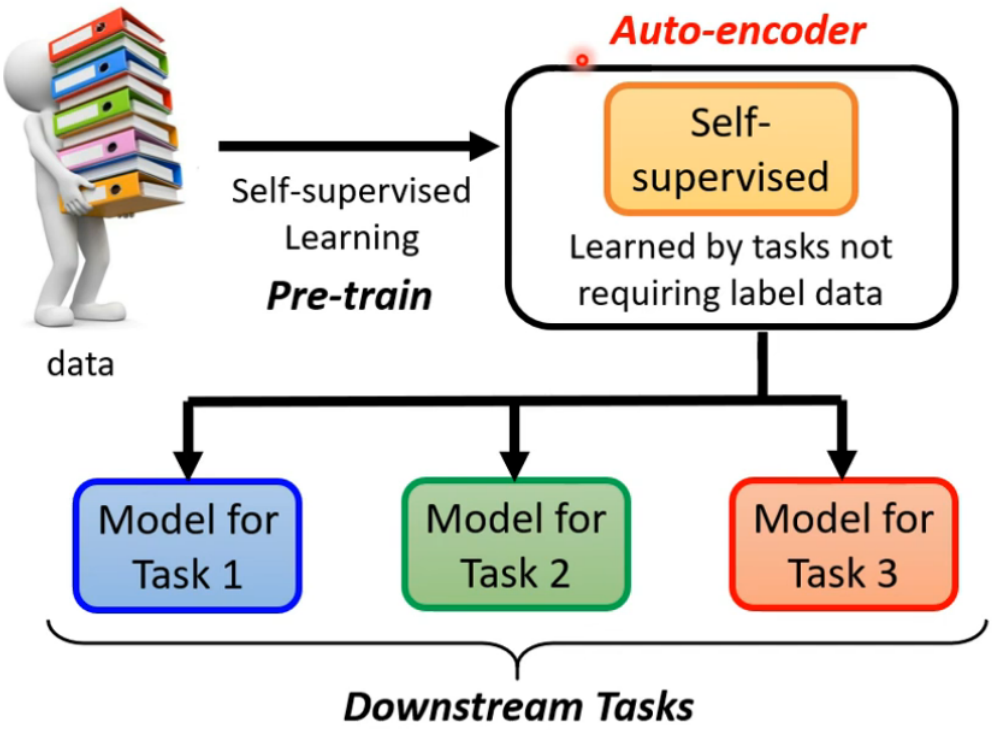
  > https://arxiv.org/abs/2006.07733



Auto-Encoder

Auto-Encoder也可以看作是self-supervised learning的一环。

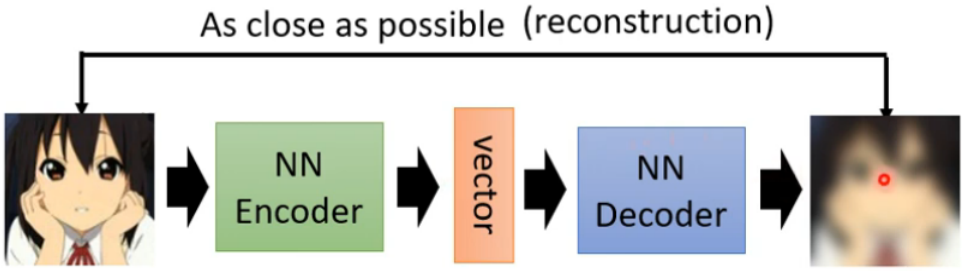
> 从self supervised learning的框架说起——Auto-Encoder的前世今生
>
> - 大量的没有标注（label）的资料（data）
> - 用这些资料训练一个模型，发明一些不需要标注资料的任务，e.g.做填空题（BERT）、预测下一个token（GPT）等等。用这些任务来给模型进行学习，这样的学习就叫做自监督学习（有人也称之为**pre-train**），这样子得到的预训练模型经过**微调（fine tune）**就可以用于其他**下游任务（downstream task）**中。



- 在有预训练（自监督学习）出现之前，存在的更古老的无需标注资料的学习任务，称之为**Auto-encoder**，（老师觉得）auto-encoder也可以看作是自监督学习的pre-train的一种方法。

Auto-encoder 如何运作?

- 大量的未标记的训练资料（课程以图像为例）
- 两个network：**Encoder**和**Decoder**
  - 输入一张图片，Encoder把输入编码，输出一个向量
  - 这个向量成为Decoder的输入，解码后输出一张图片（类似于GAN里头的Generator）
  - 两者都是多层的Network



- 训练目标：Decoder输出的图片和Encoder输入的图片越像越好（把图片看作向量，那么希望Decoder输出的向量和Encoder输入的向量距离越接近越好）
- 以上这个过程，有人也称之为reconstruction（重建）。和Cycle GAN思路几乎一模一样。
- Encoder的输出（vector）有时候我们叫它Embedding或是Representation或是code。

Auto-encoder如何用于下游任务?

- 输入的图片可以看作是一个很长的vector，Encoder的作用：降维（Dimension Reduction）、压缩为低维度的向量。

  > 降维（Dimension Reduction）技术：（not nased ML）PCA、t-SNE

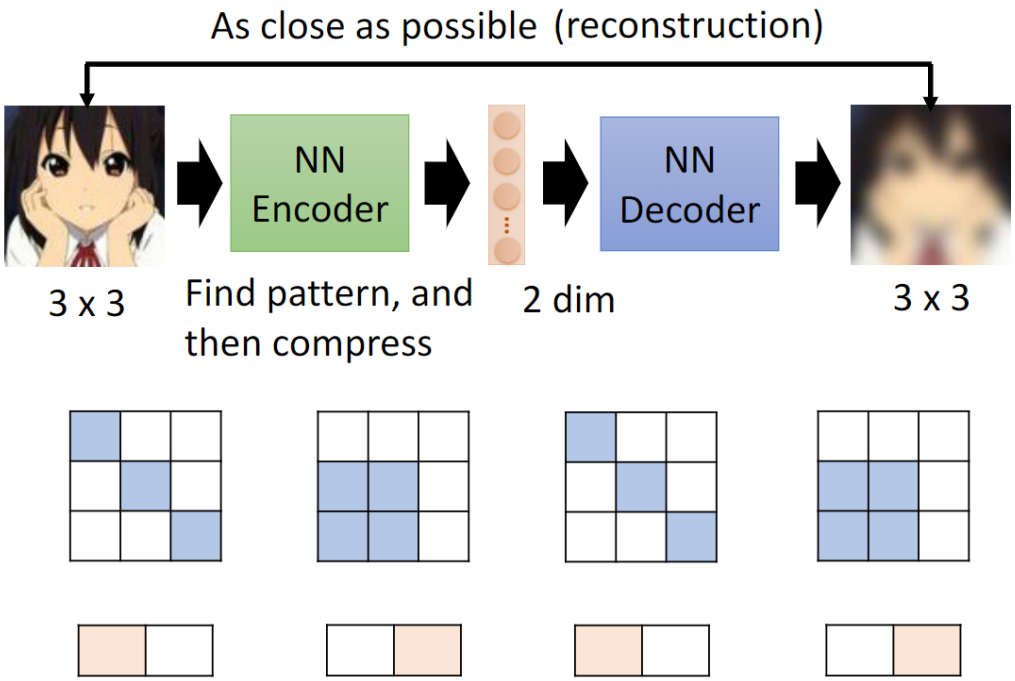
- 得到Encoder的特征提取，Embedding会是一个low dim vector；由于输出图像也是一个高维向量。所以embedding这部分也被称之为bottleneck。

WHY Auto-encoder?

> 引子——神雕侠侣：杨过和樊一翁

auto-encoder所做的就是把一张图片压缩然后又还原回来。思考这样一个问题：以一张3×3图片（9个数值）为例，如果encoder将该图片压缩到2维（两个数值），那么decoder如何从2维的low dim embedding中还原出3×3的输出图像？

原因在于：图片的变化/特征是有限的表达的，对于3×3的9个数值，并不是所有数据都表征了该图片。



Encoder做到了化繁为简，找出复杂的东西（本质的）有限的变化，找到简单的模式，那么就可以用比较少的训练资料完成机器学习的任务。

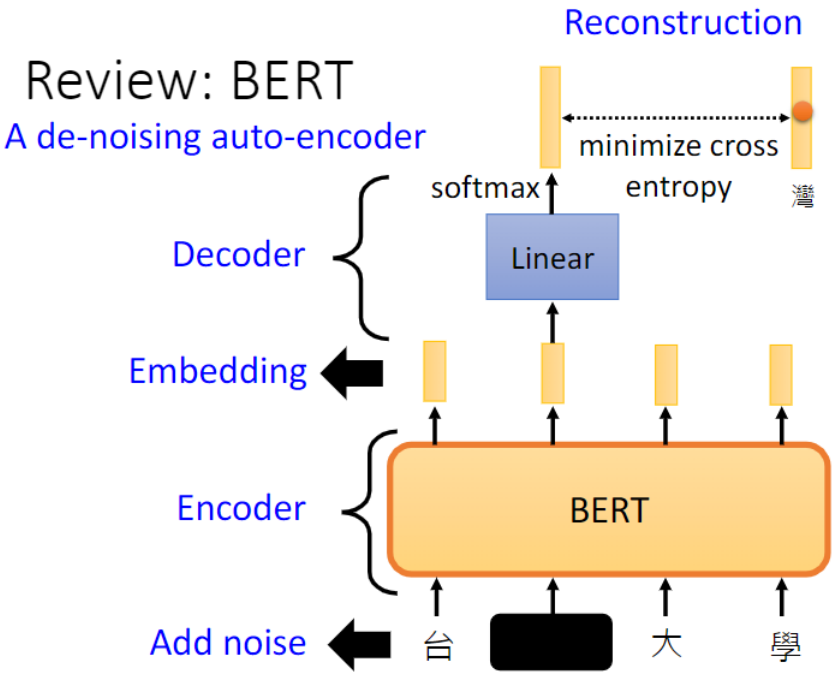
（Auto-encoder is not a new idea……）Hinton在2006年发表在Science中的文章用的RBM技术来处理编码器的“pretraining”；分层来train，而不是一起deep train（过去觉得train不起来）。

> 受限玻尔兹曼机（英语：restricted Boltzmann machine, RBM）是一种可通过输入数据集学习概率分布的随机生成神经网络。

Auto-encoder的一种变形：De-noising Auto-encoder

> 在以上讲述的auto-encoder的步骤 + 原来要输进encoder的图片加上一些噪声 + 还原加入噪声之前的图片

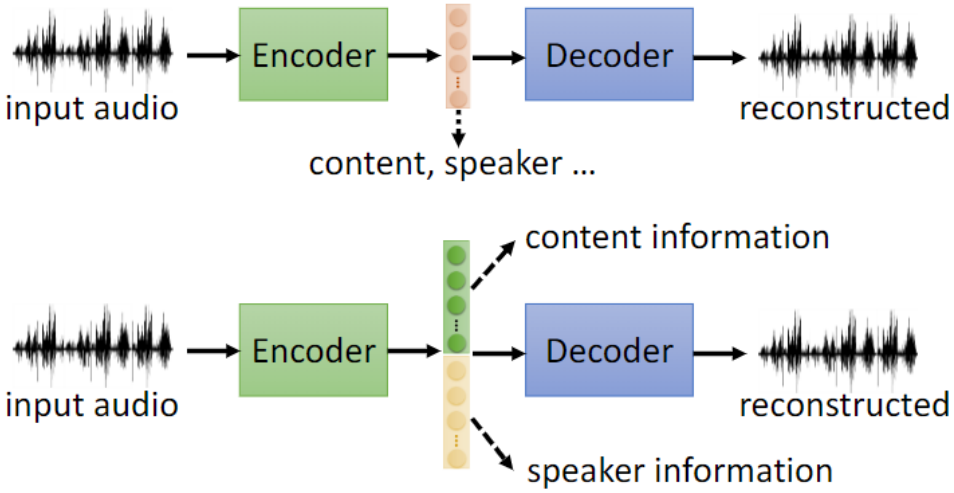
- 联手学会去掉噪声
- BERT很类似：这个decoder不一定必须是linear的。对于整个bert而言，如果中间比方说第六层输出是embedding，那么前六层就当作encoder，后几层就是decoder



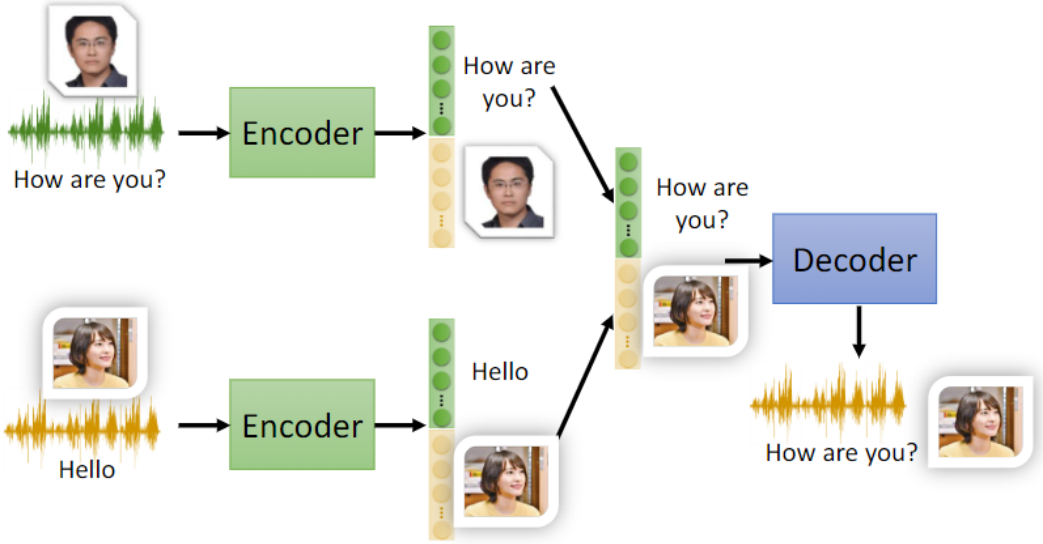
Auto-encoder：Feature Disentanglement

> 除了下游任务，auto-encoder的其他有趣的应用；disentanglement：纠缠的东西解/分离开。文章如下
>
> https://arxiv.org/abs/1904.05742；https://arxiv.org/abs/1804.02812；https://arxiv.org/abs/1905.05879

其目的：了解在train一个Auto-encoder时，在encoder产出的embedding中每个维度都代表了哪些资讯。



- 应用1：Voice Conversion：柯南的领结变声器
  - 如果supervised learning：需要对称的训练数据，如果我想变声新垣结衣呢——数据很难收集。
  - 用Feature Disentanglement，知道embedding的维度的表征含义，我们就可以——



- 以上这件事情居然是可以办得到的。效果有点……
- 影像上，nlp上Feature Disentanglement都可以有相应的应用

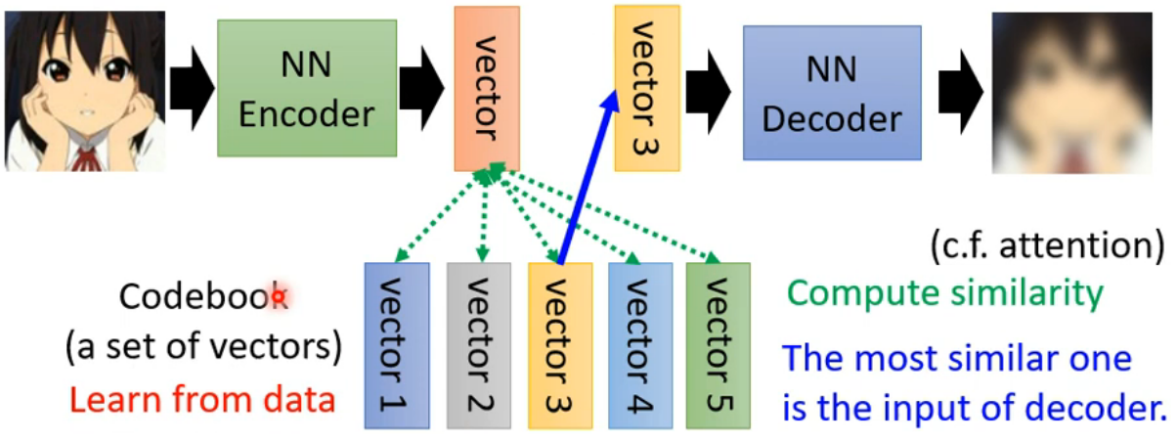
Discrete Latent Representation

目前为止，我们都假设embedding是一个向量；那抹，如果是一串binary呢？如果是一个one-hot呢？

- binary：判断某些特征是否有无
- one-hot：做到unsupervised的分类，这时候指的就是特征了，例如在手写数字识别任务中。这使得在non-label data训练情况下让机器自动学会分类。

最知名的：Vector Quantized Variational Auto-encoder (VQVAE)，流程如下

- 输入一张图片，Encoder输出一个normal的vector，它是连续的（continuous）
- 预先有一个codeBook（a set of vectors），把Encoder的向量和依次和codebook中的每个向量计算相似度（老师表示很类似attention，key和value共用了一个vector）
- 找到codebook中相似度中最大的那个vector，丢进Decoder中，输出一张图片。
- 接下来的training就是使输入输出越接近越好。Encoder、Decoder以及Codebook都是一起从资料中被学出来的。

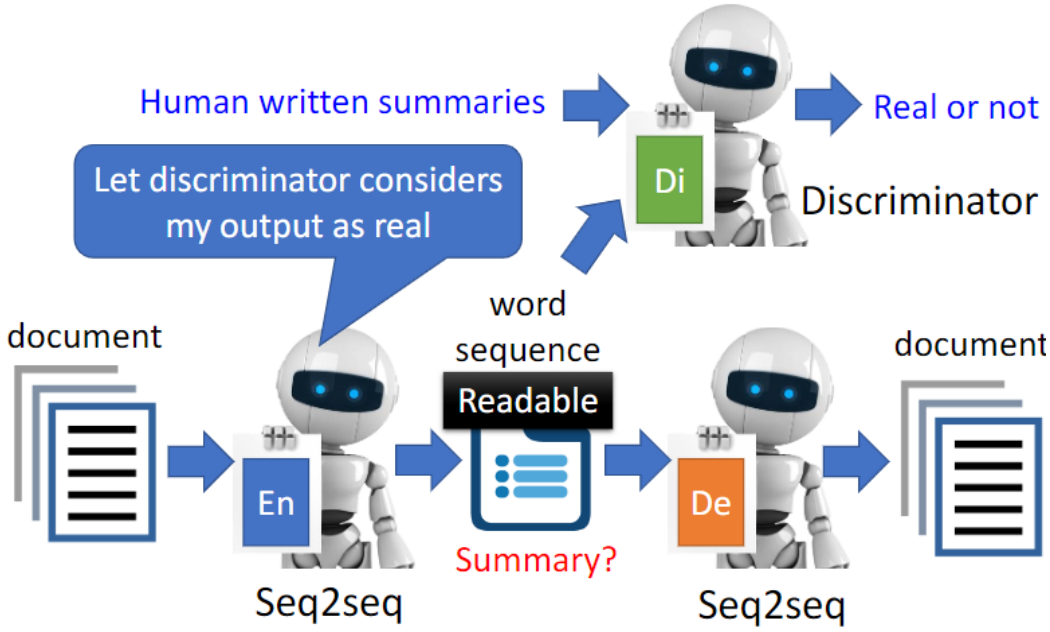


- 好处：Latent Representation被离散了（discreted）。所有Decoder的输入只能存在在codebook中，等于让这个embedding是离散的，其可能 取值是有限的。
- 有意思的是，当这种idea应用到语音的时候，codebook可以学到最基本的发音单位（phoneme），里边的每一个vector就对应着训练资料总体声音集的某一个基本发音。https://arxiv.org/pdf/1901.08810.pdf

Text as Representation（embedding）

> crazier idea：Representation（embedding）只能是一段向量吗？如果是一段文字呢？当然可以——

- 一篇文章丢进Encoder，产生一个word sequence，把这个文字序列丢进Decoder还原一篇文章。这串word sequence仿佛就是**这篇文章的摘要。**
- 显然地，Encoder和Decoder必须是Seq2seq模型，比方说transformer
- 这样的整体就是**seq2seq2seq auto-encoder**，把长的sequence压缩成短的sequence，再把短的sequence还原为长的sequence，只需要大量没有标注的资料（文章），理论上讲这就是一个unsupervised的summarization；然而按照这样的简单逻辑实际上根本train不起来，原因在于实际train了后Encoder和Decoder之间会发明自己的"暗号"，产生的embedding基本上是unreadable的...（当然Decoder是看得懂的）
- 对以上的**seq2seq2seq auto-encoder**进行改进使其work：（参考GAN）加上一个Discriminator，Discriminator看过人写的文章（摘要），知道人写的句子长什么样子，可以判断Encoder的输出是否像是人写的句子。



- **看起来没办法train的问题，RL硬做。**（硬train一发）
- Text as Representation结果：（fail or success）
  - ***Document***:澳大利亞今天與13個國家簽署了反興奮劑雙邊協議,旨在加強體育競賽之外的藥品檢查並共享研究成果 .....
  - ***Summary***:
    - Human:澳大利亞與13國簽署反興奮劑協議
    - Unsupervised:澳大利亞加強體育競賽之外的藥品檢查
  - ***Document***:中華民國奧林匹克委員會今天接到一九九二年冬季奧運會邀請函,由於主席張豐緒目前正在中南美洲進行友好訪問,因此尚未決定是否派隊赴賽 .....
  - ***Summary***:
    - Human:一九九二年冬季奧運會函邀我參加
    - Unsupervised:奧委會接獲冬季奧運會邀請函

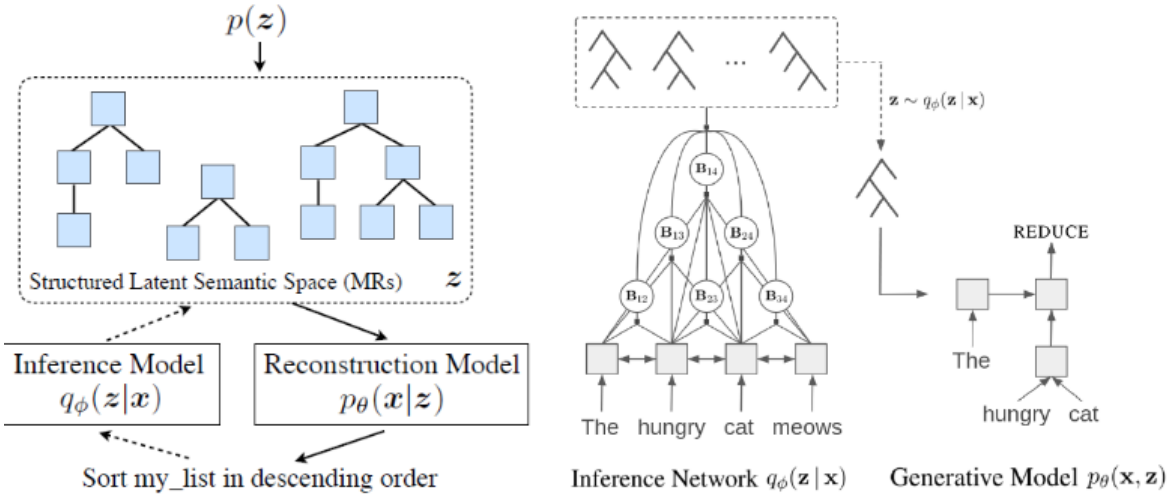
机器学会主动将奥林匹克运动会缩写为奥运会

- ***Document***:據此間媒體27日報道,印度尼西亞蘇門答臘島的兩個省近日來連降暴雨,洪水泛濫導致塌方,到26日為止至少已有60人喪生,100多人失蹤 .....
- ***Summary***:
  - Human:印尼水災造成60人死亡
  - Unsupervised:印尼門洪水泛濫導致塌雨
- ***Document***:安徽省合肥市最近為領導幹部下基層做了新規定:一律輕車簡從,不準搞迎來送往、不準搞層層陪同 .....
- ***Summary***:
  - Human:合肥規定領導幹部下基層活動從簡
  - Unsupervised:合肥領導幹部下基層做搞迎來送往規定:一律簡

Tree Structure as Embedding

一段文字转换成tree structure，再把tree还原为一段文字。参考论文如下：

> https://arxiv.org/abs/1904.03746；https://arxiv.org/abs/1806.07832

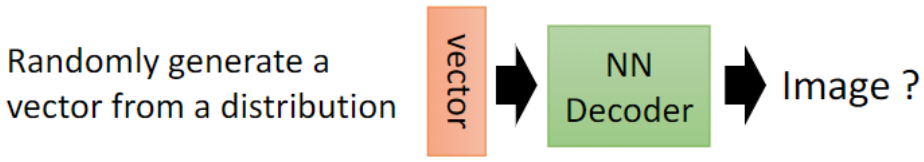


MORE Applications

Auto-encoder更多的应用，举例来说

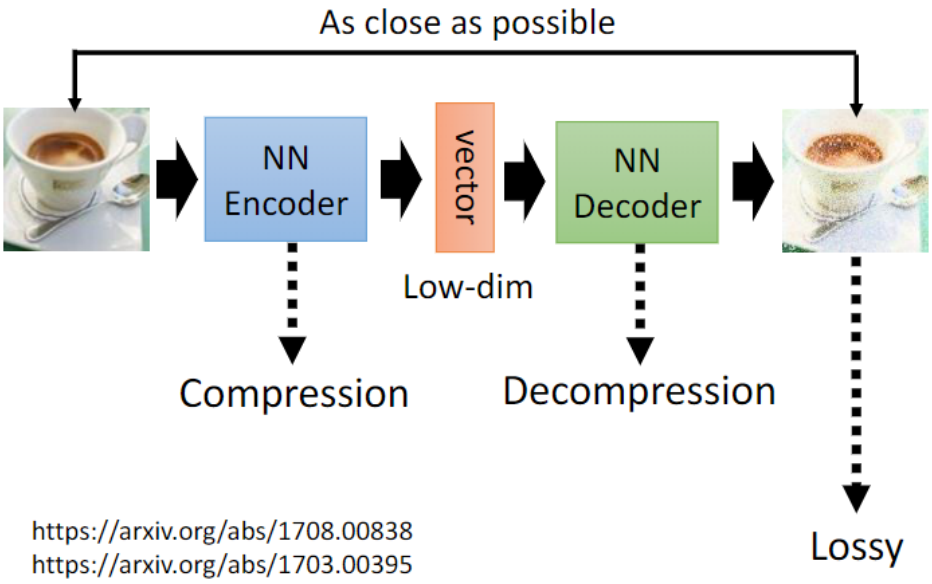
- 把Decoder及其输入（embedding）输出拿出来，就是一个Generator

  从一个已知的distribution，sample出一个vector，丢进Decoder里边，看是否生出图。上文对GAN的笔记里也提到了除了GAN以外的Generator譬如Variational Auto-encoder（VAE），其基本原理类似，VAE还做了其他事情（改进或变化）。



- Auto-encoder拿来做压缩。由于Encoder的输入是高维的向量，而输出embedding一定是低维的。将Encoder的输出（embedding）完全可以当作是压缩的结果。而Decoder拿来做解压缩。

  这个压缩是会失真的（lossy）。论文：https://arxiv.org/abs/1708.00838；https://arxiv.org/abs/1703.00395



https://arxiv.org/abs/1708.00838
https://arxiv.org/abs/1703.00395

- Auto-encoder来做**异常检测（Anomaly Detection）**

> 介绍**异常检测（Anomaly Detection）**：
>
> - Given a set of training data $\{x^1, x^2, \ldots, x^N\}$
> - Detecting input x is similar to training data or not（找到离群点）
>
> normal <----> anomaly（outlier，novelty，exceptions）

  - “相似”这件事并没有清晰的绝对的具体定义，通常根据情景而表现不同，换言之“相似”是相对的，取决于训练资料的成分。
  - 欺诈侦测（Fraud Detection）

> • Training data: credit card transactions, x: fraud or not
> • Ref: https://www.kaggle.com/ntnu-testimon/paysim1/home
> • Ref: https://www.kaggle.com/mlg-ulb/creditcardfraud/home

  - 网络入侵检测（Network Intrusion Detection）

> • Training data: connection, x: attack or not
> • Ref: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

  - 医学上的影像检测（分类），比如说Cancer Detection

> • Training data: normal cells, x: cancer or not?
> • Ref: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

我们能不能把这种异常检测任务来当作二元分类（Binary Classification）？的确两个任务非常相像，但是，其问题在于数据的收集中，异常检测中的负样本相对来说非常的少，这种存在的样本不平衡在实际的数据集中往往体现：绝绝大多数是正样本（正常的）资料，而几乎没有异常的资料（统计上讲）。因此，异常检测是不同与一般的分类任务的，这类分类问题被称之为单类分类任务（**One Class Classification**）

我们只有一个类别的资料，如何训练我们的分类器？——Auto-encoder派上用场↓

利用Auto-encoder这样一种特性：

当完成整个Encoder-Decoder在单类训练集上的训练后，对于正样本（正常数据）的输入，由于Auto-Encoder在训练集中看过类似的图/文本，因此经过Encoder的编码，Decoder的解码能完成这个样本数据的重建（reconstruction）；换言之，对于normal的数据，Auto-encoder的输入和输出是趋于一致的，或者是相似的。

但是，对于异常数据的输入，由于Auto-Encoder在训练集中未接触到类似的图/文本，Decoder的重建无法完成；换言之，对于 anomaly的数据，Auto-encoder的输出和输入差异会很大。

通过Auto-encoder这样的特性，我们可以完成一个单分类的分类器。