FOUNDATION

- Supervised Learning (Lecture 1-5)
  Not Efficient to collect data for each task
- Self-supervised Learning (Lecture 7)
  Pre-train (Develop general purpose knowledge)
  BERT GPT-3 GPT-2
  ELMO T5
- Generative Adversarial Network (Lecture 6)
- Reinforcement Learning (Lecture 12)

ADVANCED

- Anomaly Detection 异常检测 Lecture 8
- Explainable AI 准确并给出解释 Lecture 9
- Model Attack 模型攻击 Lecture 10
- Domain Adaption 训测分布不一致 Lecture 11
- Network Compression Lecture 13
- Life-long Learning Lecture 14
- Meta Learning Lecture 15
  Few-shot Learning Learn to learn

Model Training

Function with Unknown Parameters

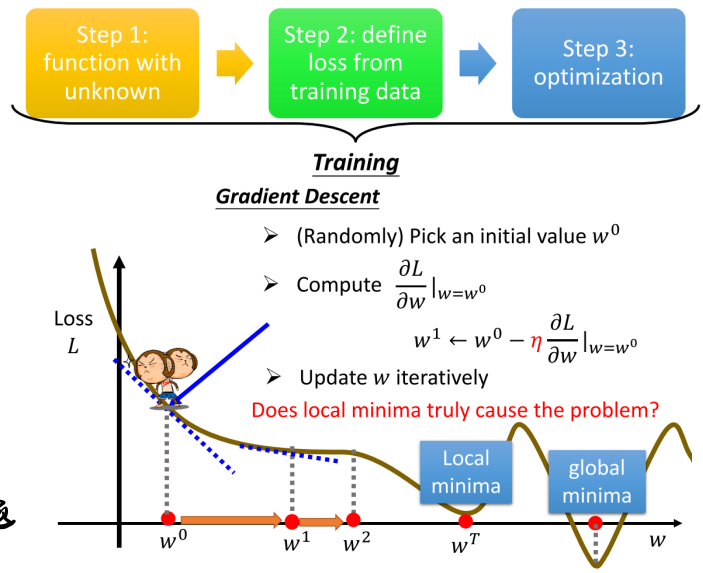
Domain Knowledge 定义模型知识

Define Loss from Training Data

- Loss 关于参数的函数 $L(b,w)$
  衡量一组参数的坏坏
- MAE (Mean Absolute Error) $e=|y-\hat{y}|$
- MSE (Mean Square Error) $e=(y-\hat{y})^2$
  y 和 $\hat{y}$ 都是概率 → Cross-entropy

Optimization $w^*, b^* = \arg\min_{w,b} L$

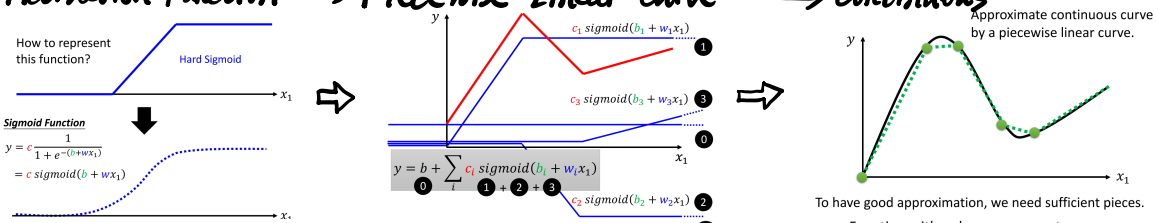
- Hyperparameters 自己定义的超参数 η 学习率
- Local minima 不是 Gradient decent 真正的问题



REGRESSION

Step 1: function with unknown

Activation Function → Piecewise Linear Curve (由线段构成的曲线) → Continuous



$y = b + wx_1$ Linear Function

Model Bias 由model本身导致的误差

$y = b + \sum_i c_i\,sigmoid(b_i + w_i x_1)$ More Flexible 单个 Feature

$y = b + \sum_j w_j x_j$ 多个 Feature

$y = b + \sum_i c_i\,sigmoid\left(b_i + \sum_j w_{ij} x_j\right)$

j: 1,2,3 no. of features

i: 1,2,3 no. of sigmoid

w_{ij}: weight for x_j for i-th sigmoid

$r_1 = b_1 + w_{11}x_1 + w_{12}x_2 + w_{13}x_3$

$r_2 = b_2 + w_{21}x_1 + w_{22}x_2 + w_{23}x_3$

$r_3 = b_3 + w_{31}x_1 + w_{32}x_2 + w_{33}x_3$

Function with unknown parameters

$y = b + c^T \sigma(b + Wx)$

x feature

Unknown parameters

W b c^T b

Rows of W

所有参数矩阵展开后拼接

$\theta = \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \vdots\end{bmatrix}$

Step 2: define loss from training data

- Loss is a function of parameters $L(\theta)$
- Loss means how good a set of values is

$y = b + c^T \sigma(b + Wx)$ feature

label $\hat{y}$ e

Given a set of values

Loss: $L = \frac{1}{N}\sum_n e_n$

Step 3: optimization

$\theta^* = \arg\min_\theta L$

- (Randomly) Pick initial values θ^0
- Compute gradient $g = \nabla L^1(\theta^0)$ L^1
  update $\theta^1 \leftarrow \theta^0 - \eta g$
- Compute gradient $g = \nabla L^2(\theta^1)$ L^2
  update $\theta^2 \leftarrow \theta^1 - \eta g$
- Compute gradient $g = \nabla L^3(\theta^2)$ L^3
  update $\theta^3 \leftarrow \theta^2 - \eta g$

1 epoch = see all the batches once

B batch batch batch batch N L

Example 1

- 10,000 examples (N = 10,000)
- Batch size is 10 (B = 10)

How many update in **1 epoch**?

1,000 updates

Example 2

- 1,000 examples (N = 1,000)
- Batch size is 100 (B = 100)

How many update in **1 epoch**?

10 updates

Regression

label $\hat{y} \leftrightarrow y = b + c^T \sigma(b + Wx)$ feature

Mean Square Error (MSE) large loss small loss stuck!

Cross-entropy large loss small loss

$MSE \equiv \sum_i (\hat{y}_i - y'_i)^2$ $CE = -\sum_i \hat{y}_i \ln y'_i$

换 Loss Function 可能改变 Optimization 更容易

最小化 Cross-entropy ⇔ 最大化 Likelihood

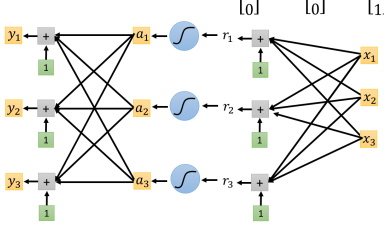
Cross-entropy 常与 Softmax 同时出现

CLASSIFICATION

Class as one-hot vector

Class 1 $\begin{bmatrix}1\\0\\0\end{bmatrix}$ Class 2 $\begin{bmatrix}0\\1\\0\end{bmatrix}$ Class 3 $\begin{bmatrix}0\\0\\1\end{bmatrix}$

与每个类别的距离相同



- 将 y 的值 Normalize 到 [0,1] 区间上，且和为1
- 扩大 max 和 min 间的差距
- Softmax 的输入称为 logit
- Binary Classification 中 Softmax ⇔ Sigmoid

Classification

feature $y = b' + W'\sigma(b + Wx)$

label $\hat{y}$ y' $y' = softmax(y)$

0 or 1 Make all values between 0 and 1 Can have any value

Step 1: function with unknown → Step 2: define loss → Step 3: optimization

- 参数越多，越容易 Overfit
- 未知参数的可能性(可选择的 Function 数量) → Complexity

We want $L(h^{train}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta$

What kind of $\mathcal{D}_{train}$ fulfill it? We Need

$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| \leq \delta/2$

$\mathcal{D}_{train}$ is a good proxy of $\mathcal{D}_{all}$ for evaluating loss L given any h. 使理想接近现实的 Dtrain

Formulation:

$L(h^{train}, \mathcal{D}_{all}) \leq L(h^{train}, \mathcal{D}_{train}) + \delta/2$

$\leq L(h^{all}, \mathcal{D}_{train}) + \delta/2$ $\quad h^{train} = arg\min_h L(h, \mathcal{D}_{train})$

$\leq L(h^{all}, \mathcal{D}_{all}) + \delta/2 + \delta/2 = L(h^{all}, \mathcal{D}_{all}) + \delta$

Dall Dtrain

- 从所有 Data 中采样 Training Data ← i.i.d (独立同分布 independently & identically)
- 希望 $L(h^{train}, Dall)$ 和 $L(h^{all}, Dall)$ 很接近
  - $L(h^{train}, Dtrain)$ 可能比 $L(h^{all}, Dall)$ 更低
  - 但 h^{all} 一定是 Dall 上 L 最小的
- 大部分情况 Dall 不可知 → 从 Dall 中采样中 Dtest 测试
  - 如果 Sample 较好 则训练结果可能会更好

→ 如何判断 Dtrain 的好坏

We want to sample **good** $\mathcal{D}_{train}$ $\quad \varepsilon = \delta/2$

$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| \leq \varepsilon$

$P(\mathcal{D}_{train}\ is\ \boldsymbol{bad}) \leq |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$ N 越大，|H| 越小 → Sample 到差 Dtrain 越难

N: Dtrain 中的样本数量 ⇒ 训练集规模↑，模型复杂度↓

|H|: 可选择的 Function 数量 (Model Complexity)

ε: 自定义参数 (ε 越小，好的 Dtrain 的标准越高) → 确定训练集规模

When parameters are continuous: VC-Dimension 判断 Complexity

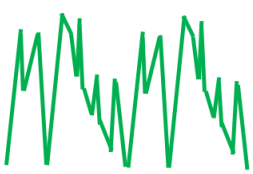
If we want $P(\mathcal{D}_{train}\ is\ bad) \leq \delta$

How many training examples do we need?

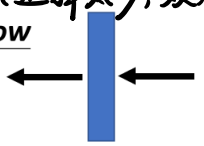
$|\mathcal{H}| \cdot 2exp(-2N\varepsilon^2) \leq \delta \Rightarrow N \geq \frac{log(2|\mathcal{H}|/\delta)}{2\varepsilon^2}$

Dilemma: 很难采集到数大的 N

|H| 过小，h 选择太少，效果差 → $L(h^{train}, Dall)$ 与 $L(h^{all}, Dall)$ 相差很小，但 $L(h^{all}, Dall)$ 本身很大

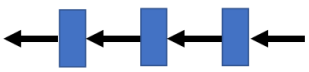


Shallow

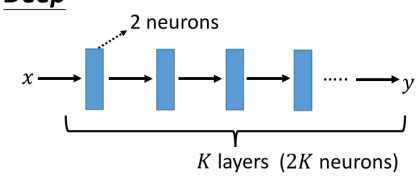


More parameters

Deep

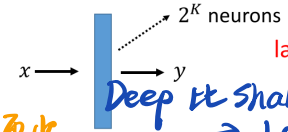


Deep

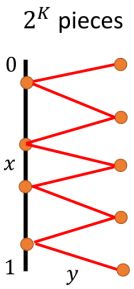


smaller |H|

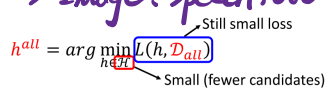
Shallow



larger |H|



- Loss 较低的 Function 是 Complex & Regular
- → Deep Network 更好
- → Image、Speech、etc



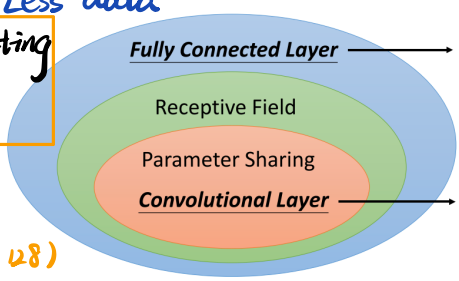
- Deep → 更低的 Loss → 更少的 Candidates

- 一层 Hidden Layer 可以表示任何 Function 但 Deep Structure 更有效 → 不易 Overfitting，需要的 Data 更少

Deep 比 Shallow 的 Parameters 少 → Less complex → Less data

全连接：Flexible 但易 Overfitting

CNN：Model Bias 大，但对于 image 来说效果更好



CNN

假设所有输入的 image 大小相同 → 先 Rescale 成一样大小

Receptive Field 感受野 → 每个 Neuron 只看局部的 Pattern

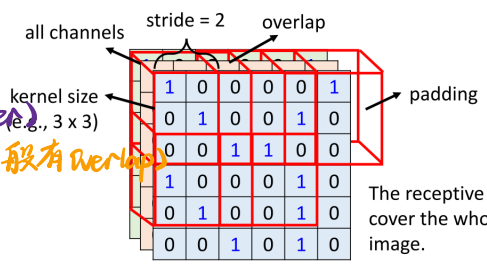
- 不同 Neuron 可以重叠，或 Receptive Field 相同
- 可以只看部分 Channel 每个 Receptive Field 一般会有多个 Neuron (64 or 128)
- 形状可以自定

Each receptive field has a set of neurons (e.g., 64 neurons).

Kernel 卷积核

Receptive Field 的长×宽

Stride 步长 (Hyper Parameter) → RF 移动的范围 (1 or 2，一般有 Overlap)



Padding 填充 → 超出范围后向 RF 中填充

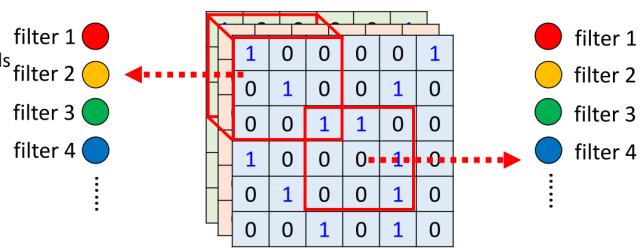
Parameter Sharing 参数共享

- Some patterns are much smaller than the whole image.
- The same patterns appear in different regions.

Each receptive field has a set of neurons (e.g., 64 neurons).

Each receptive field has the neurons with the same set of parameters.

- 相同的 Pattern 可能出现在图片的不同区域 (beak) → Parameter Sharing
- 两组 RF 不同的 Neuron 的 weight 相同 (权值共享) → RF 相同的 Neuron 不会共享参数 → 共用的参数 → Filter → 每个 RF 的 Neuron 参数相同



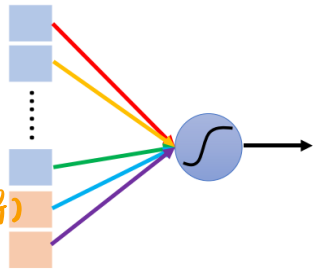
Filter 捕捉特征 → Feature Map

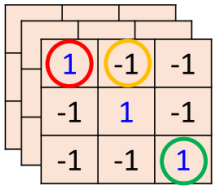
- 每个 Filter 检测一个 Pattern
  → 对整张图片做 Inner Product
- Filter 中的 value 是 Parameter
  → Unknown (Gradient Descent 学习所得)
- 一层卷积层会有多个 Filter (e.g. 64)
  所有 Filter 的结果 → Feature Map
- Filter 的高度 = Channel 数目
- Deeper Network (More Layers) → 侦测的 Pattern 越大

Feature Map 可以看做 # Channel = # Filter 的 image
Feature Map 的 Channel 数 = 上一层 image 的 Filter 数

Pooling 池化 通过 Subsampling 减小运算量

- 不是一个 Layer ← 没有需要学习的参数
  更类似 Active Function
- 精度较高的 image 可以取消 pooling 效果





Receptive field

Filter 中有 Bias
(ignore bias in this slide)

Neuron Version Story	Filter Version Story
Each neuron only considers a receptive field.	There are a set of filters detecting small patterns.
The neurons with different receptive fields share the parameters.	Each filter convolves over the input image.

不同的 Neuron 权值共享 观测不同的 Receptive Fields
⇩
每个 Filter 对整个 image 做卷积 → Convolution

WHY CNN?

- Image 中存在很小的局部特征 (Pattern)
- 相同 Pattern 可能出现在 Image 的不同区域
- Subsampling 不会改变物体

Disadvantage: 不能处理 Scaling & Rotation (缩放和旋转) → Data Augmentation

---

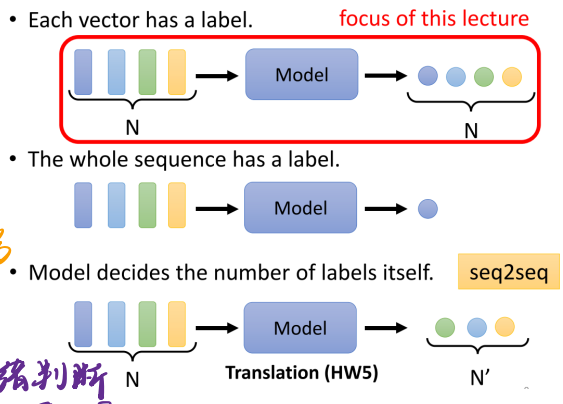
SELF-ATTENTION

应用场景 Sequence as Input

- 文本 One-hot Encoding Vector 间缺乏关联性，且过于冗长
  Word Embedding → 给每个 word 一个向量
  一个句子 ⇒ 一排长度不一的向量
- 语音 Frame (语音上的)向量 e.g. 用一个向量描述 25 ms 内的语音信号
- 图 Graph 中每个 Node 为一个向量

Output

- Each vector has a label POS Tagging 词性标注
- The whole sequence has a label Sentiment Analysis 情绪判断
- Model decides # labels Sequence to sequence Translation 翻译



Neighbor Information 上下文信息

全连接层若想 Neighbor (上下文) 需要很大的 window
⇩
→ 容易 Overfitting

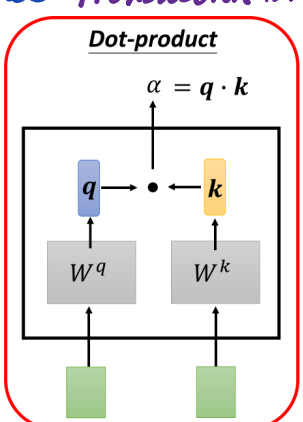
Self-Attention 可用作 Input 也可作 Hidden Layer

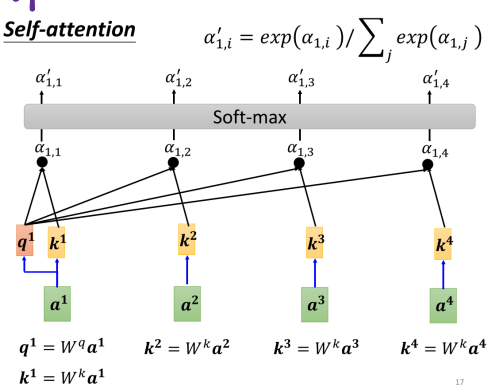
Structure 结构

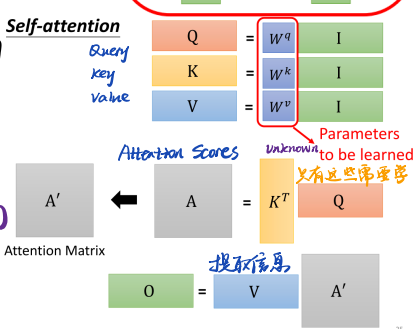
- 计算 query, key, value
  分别对 a 乘上 W^q, W^k, W^v
- Input vector 间的关联性
  α → vector 间的关联性 (关联程度) Attention Score
  - 通过 Dot-product 计算 Attention Score α
  - 使用 Softmax (or Relu) 对 α 作 Normalization
- 基于 Attention Scores 提取信息
  α 最大的 a 对应的 v 会主导 Output vector b

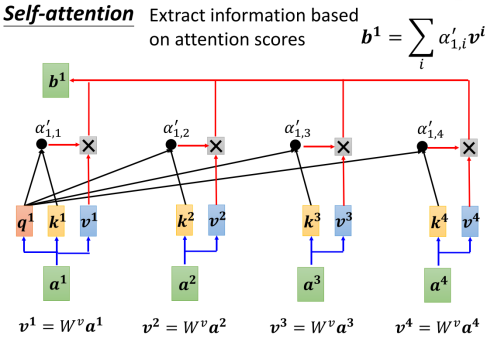
Features 特性

- 从一组 Input Vector (a) 得到一组 Output Vector (b)
- Output vectors ($b_1 \cdots b_4$) 同时被计算出来
  (Parallel)
- 只有 W^k, W^q, W^v 是需要学习的参数 (未知)







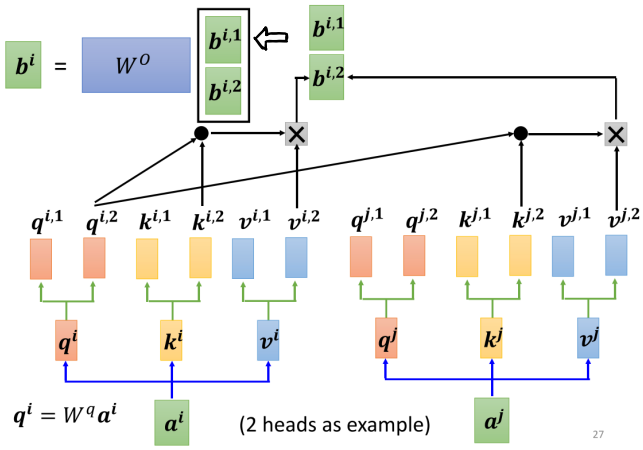


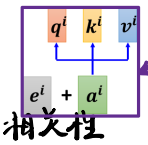
Multi-head Self-Attention

相关 → 用 q 去匹配 K (Dot-product)

Multi-head → 不同的 q 表示不同种类的相关性

Multi-head Self-attention Different types of relevance





Positional Encoding

Self-attention 中没有对 vector 在 sequence 中的位置信息。

→ 每个位置都有唯一的 Positional Vector e^i ← Hand-Crafted

← 从 Data 学得

Application

- Speech → 只看一小段 Sequence Truncated Self-Attention
- Image Image 看作一个 Vector Set 每个 Pixel 看作一个 3 维 Vector

CNN v.s. Self-Attention

- CNN 只关注 Receptive Field 中的特征

→ Self-Attention 关注整个 Image (Sequence) 的信息

- CNN 的 Receptive Field 是人为设定 (Fixed)

→ Self-Attention 的 Receptive Field 是学习所得

- Self-Attention 是更 Flexible 的 CNN

通过对 Self-attention 的设计和限制

可以实现 CNN 的功能

{ Less Data → CNN 更好 Self-Attention Overfitting

More Data → Self-Attention 更好

CNN: self-attention that can only attends in a receptive field

➢ CNN is simplified self-attention.

Self-attention: CNN with learnable receptive field

➢ Self-attention is the complex version of CNN.

RNN v.s. Self-Attention

- 相邻较远的 Vector, SA 更容易从中提取关联

RNN 需要一直存在 Memory 中

- RNN 不可并行，SA 可以并行 (parallel)

Reference

Long Range Arena: A Benchmark for Efficient Transformers

Efficient Transformers: A Survey

加速 Attention 机制的计算 N×N 计算量可能会非常大

Self-Attention 只是一个巨大网络中的一个 module

→ 当 N 非常大的时候，提高 S-A 效率影响较大

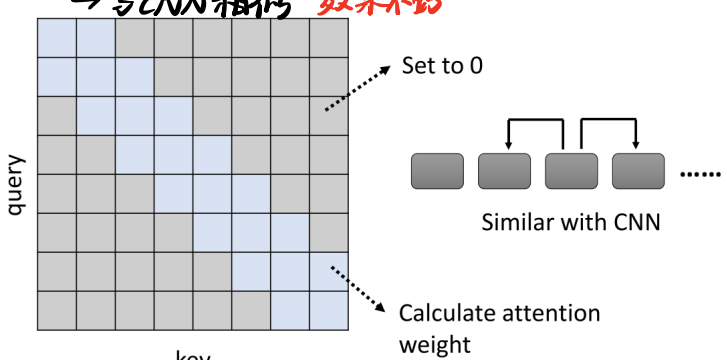
→ 通常用于图像处理 (e.g. N = 256 × 256)

Skip Calculations (with Human Knowledge)

- Local Attention (Truncated Attention)

→ 只看相邻的 Vector

→ 与 CNN 相似 效果不好



- Stride Attention → 跳过几个 Vectors

从某个位置为中心，看左右两边发生的事情

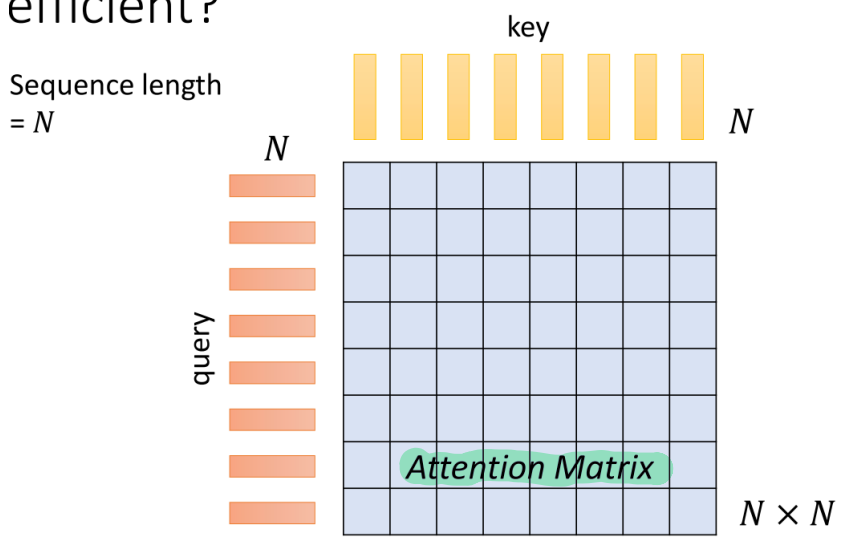
- Global Attention → 在原 Sequence 中加入 Special Token

{ Input Sequence 中原有的 Token (e.g. CLS) 作为 Special Token

向 Input Sequence 中加入额外的 Special Token

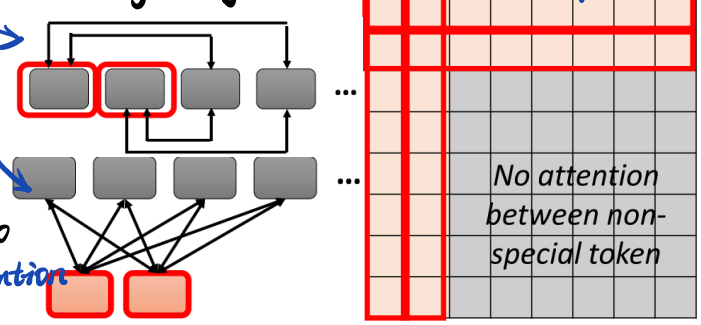
→ 除了 Special Token，其他 Token 不作 Attention

How to make self-attention efficient?



⇨ 不同 Head 使用不同 Attention e.g. Longformer, BigBird

- Attend to Every Token 收集整个 Sequence 的信息
- Attended by Every Token 了解 Global Information



Critical Parts

对 Attention Matrix 中 small value 直接置零

→ Clustering 根据 Query 和 Key 的相似程度做 Cluster

同一个 Cluster 的 Query 和 Key 做 Attention，其他为 0

Learnable Patterns

使用另一个 Network 学习哪些位置做 Attention

Sinkhorn Sorting Network 参数和 SA Layer 同时学得

(Jointly Learned)

Representative Key 去除 Attention Matrix 中的冗余信息

Redundant Columns (Low Rank) → 信息冗余

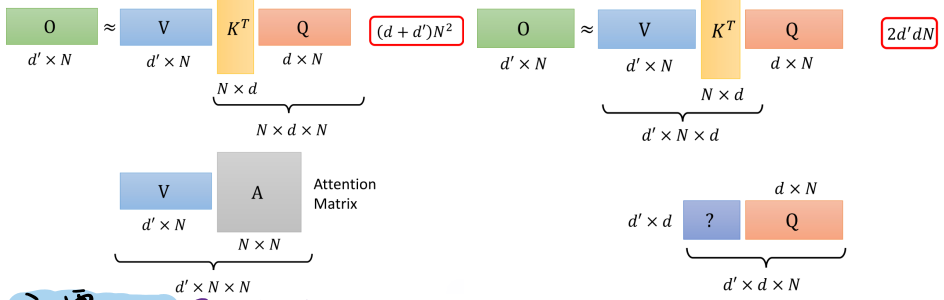
→ 从N个 key 中选取K个有代表性的 key & value (key & value 一对应)

改变 Query 数目 → 改变了 Output Sequence 的长度

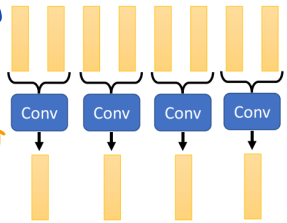
Compressed Attention 用 CNN 压缩 Sequence (K个代表性的 key)

Linformer 将 Sequence 看作 $d \times N$ 矩阵，乘上 $N \times K$ 矩阵 → $d \times K$ 矩阵

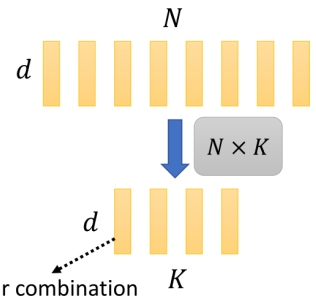
K,q first → v,k first 三个矩阵相乘 <⇒ Self-Attention



Compressed Attention
https://arxiv.org/abs/1801.10198

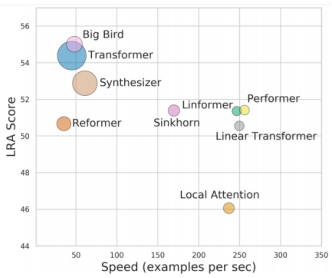


Linformer
https://arxiv.org/abs/2006.04768



CONCLUSIONS

- Human knowledge
  - Local Attention, Big Bird
- Clustering
  - Reformer
- Learnable Pattern
  - Sinkforn
- Representative key
  - Linformer
- k,q first → v,k first
  - Linear Transformer, Performer
- New framework
  - Synthesizer



- 圈越大，占用空间越大
- 越右越快
- 越上越准

无需 q、k Synthesizer

→ 将 Attention Matrix 作为网络参数学习 不会使表现变差

→ Input 不同的 Sequence，Attention Weight 都相同

Attention-free 无需 Attention 处理 Sequence

- Fnet: Mixing tokens with fourier transforms
- Pay Attention to MLPs
- MLP-Mixer: An all-MLP Architecture for Vision

TRANSFORMER

Sequence to Sequence Output Sequence 的长度由模型决定

大部分 NLP 任务都可以看作 Question Answering (QA)

→ 使用 Seq2seq 模型解决 但不一定效果最好

e.g. Google Pixel 4 使用 RNN Transducer

Syntactic Parsing 文法剖析 树状结构 → Sequence

e.g. Grammar as a Foreign Language

Multi-label Classification 多标签分类

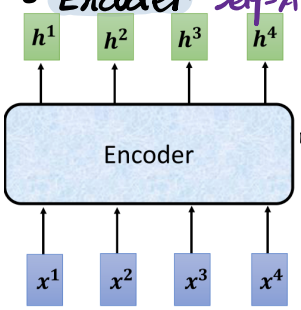
https://arxiv.org/abs/1909.03434
https://arxiv.org/abs/1707.05495

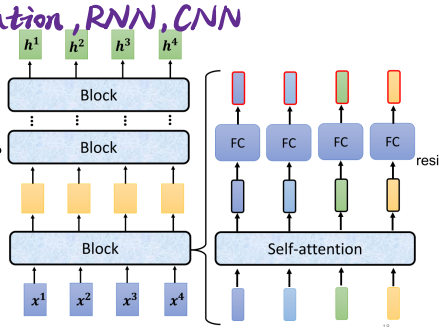
- 同时属于多个 Class
- 由 Seq2seq 模型决定

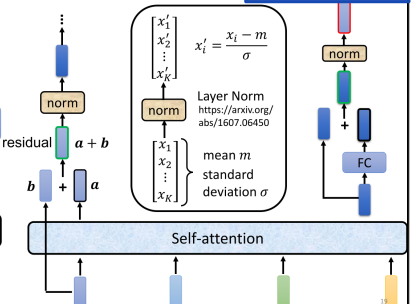
Object Detection 目标检测 https://arxiv.org/abs/2005.12872

Structure 结构

- Encoder Self-Attention, RNN, CNN







- On Layer Normalization in the Transformer Architecture
  https://arxiv.org/abs/2002.04745
- PowerNorm: Rethinking Batch Normalization in Transformers
  https://arxiv.org/abs/2003.07845

→ 输入一个 Vector 序列，Output 一个 Vector Sequence

→ BERT 和 Transformer 使用了相同的 Encoder

- Decoder Autoregressive (AT)

→ 读取 Encoder 的 Output 作为 Decoder 的 Input

Bos Begin of Sentence

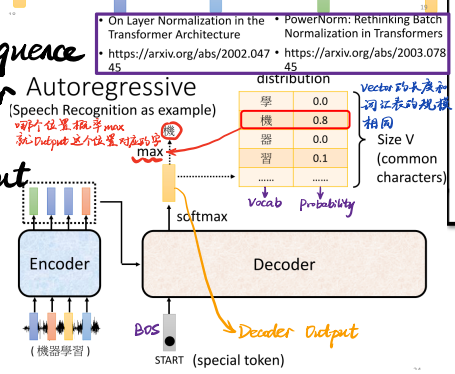
用 One-hot Vector 表示，1 就是 1，其他是 0

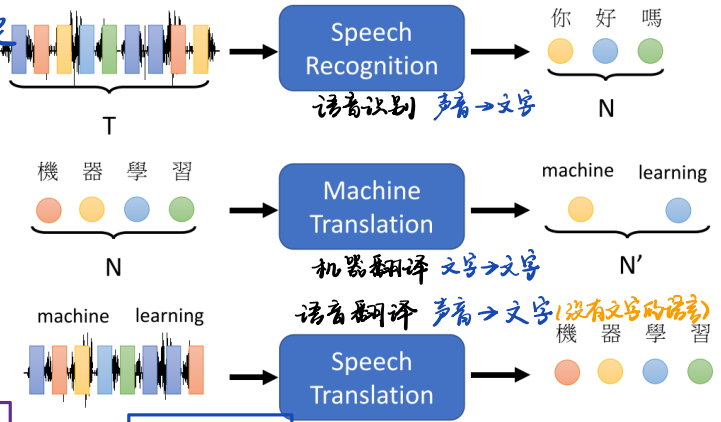
→ Decoder 决定 Sequence 的长度

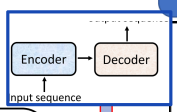
加入 Stop Token Output END 的概率最大

Autoregressive

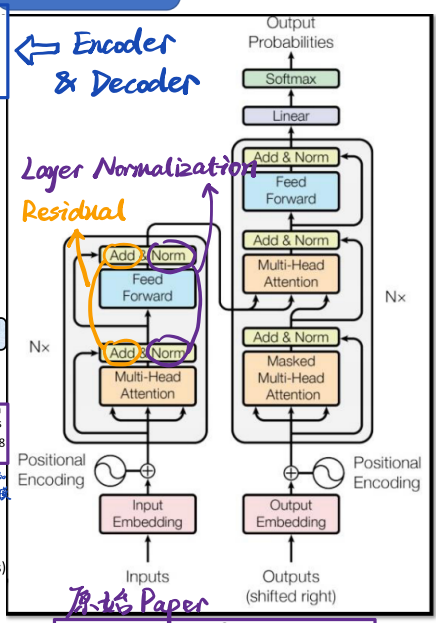
(Speech Recognition as example)







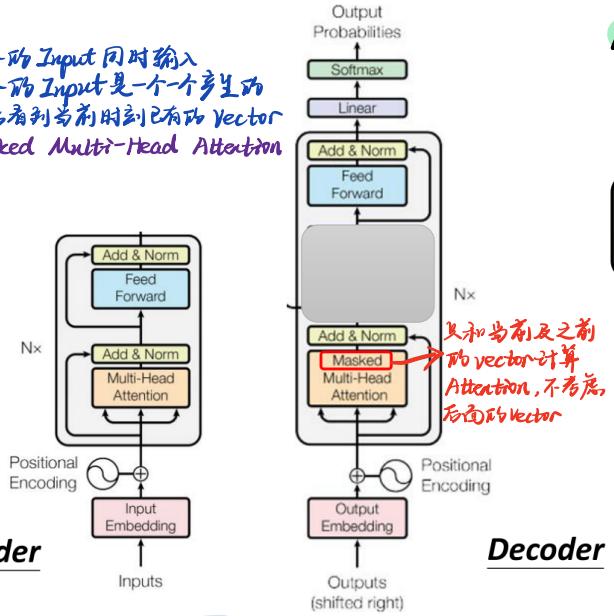
⇦ Encoder & Decoder



原始 Paper

Transformer
https://arxiv.org/abs/1706.03762

Encoder的Input同时输入
Decoder的Input是一个一个产生的
→ 只能看到当前时刻已有的Vector
→ Masked Multi-Head Attention



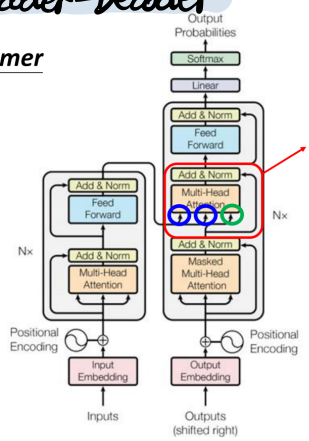
算和当前及之前的vector计算Attention，不看后面的vector

Encoder

Decoder

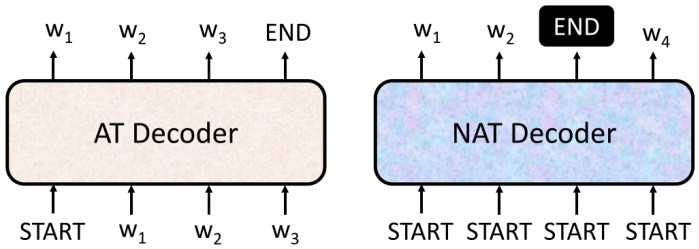
- Encoder-Decoder

Transformer



Non-autoregressive (NAT) 逐字 Output → Sequence Output

AT v.s. NAT

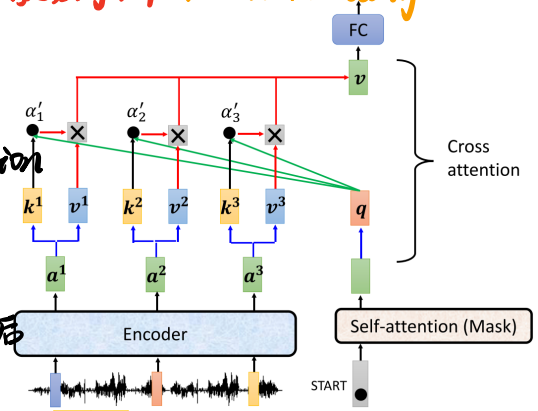


- AT一个字一个字产生 每次一个Token
  NAT一词放入一排BOS，同时产生整个Sequence 每次一个Sequence
- Output Length → 使用另一个Classifier输出Sequence长度
  → 输出一个长序列，忽略END后的Token
- Advantage • 并行运算 运行更快
  • 便于控制Output Sequence长度 稳定生成
- NAT的Performance一般差于AT Multi-modality

Cross-Attention

- Encoder提供key和vector
  Decoder提供query
- 先有Cross-Attention后有Self-attention

Listen, attend and spell: A neural network for large vocabulary conversational speech recognition
https://ieeexplore.ieee.org/document/7472621

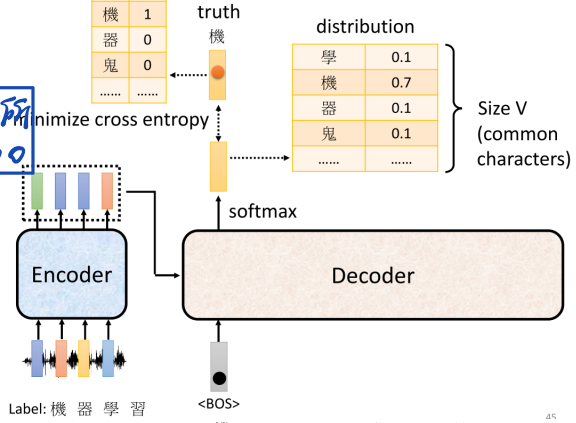


- 每一层Decoder都使用Encoder最后一层的Output (原始Transformer)
  → 可以设计不同的C-A方式

- Training

→ Ground Truth是一个长度为vocab大小的One-hot Vector
  Output是一个Probability Distribution
→ 计算每个Vector的GT与Output的Cross Entropy
  所有Output的Cross-Entropy总和越小越好
→ 训练时，在Decoder的Input阶段给Ground Truth
  → Teacher Forcing使用Ground Truth作Input

这个字在vocab中所在位置为1，其余为0

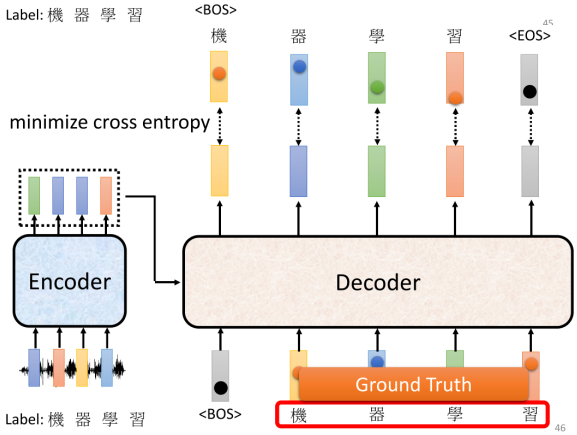


- Tips

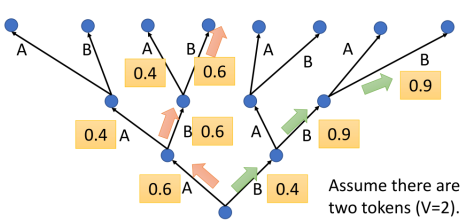
Copy Mechanism Create → Copy
有时不需要Decoder产生全部的Output
只需要从Input中复制部分作为Output → e.g. Chatbot 生僻字 Summarization 摘要
→ Pointer Network, Copying Network

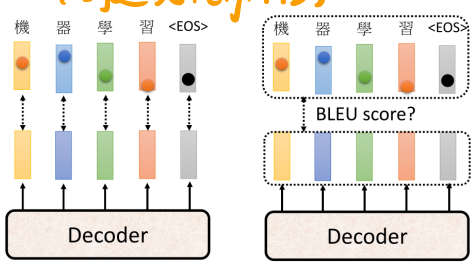
Guided Attention 低资源误(e.g. TTS漏读字)
引导机器的Attention过程，做Attention时设定规则
→ Monotonic Attention, Location-aware Attention

Beam Search 估测出一个最好的Solution
→ 对有固定答案的任务效果比较好，对需要随机性的任务效果不好 (创造类 e.g. TTS)



The red path is **Greedy Decoding**.
The green path is the best one.
Not possible to check all the paths ... → Beam Search



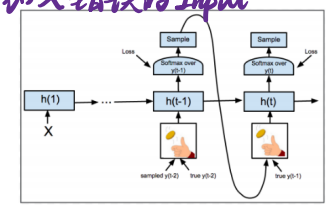


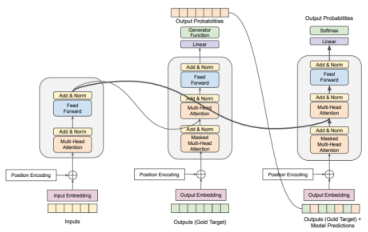
Optimization Evaluation Metrics
训练用Cross-Entropy，Validation用BLEU
→ 将BLEU Score作为RL Agent的Reward
https://arxiv.org/abs/1511.06732

Scheduled Sampling

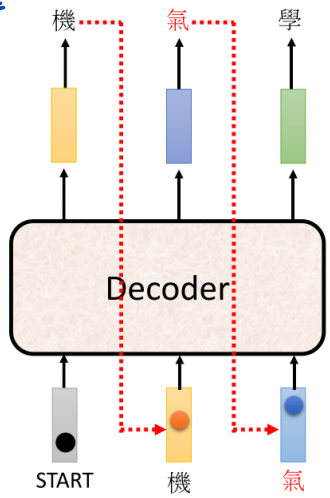
Exposure Bias (Mismatch) 训练用的都是 Ground Truth，Test 时会有错误的 Input 影响准确率
→ 在训练时加入错误的 Input

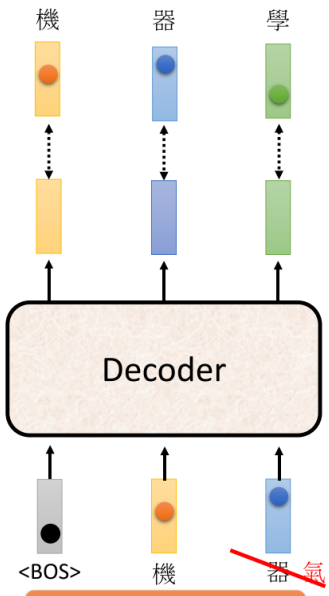
- Original Scheduled Sampling
  https://arxiv.org/abs/1506.03099
- Scheduled Sampling for Transformer
  https://arxiv.org/abs/1906.07651
- Parallel Scheduled Sampling
  https://arxiv.org/abs/1906.04331





There is a mismatch! ☹
exposure bias



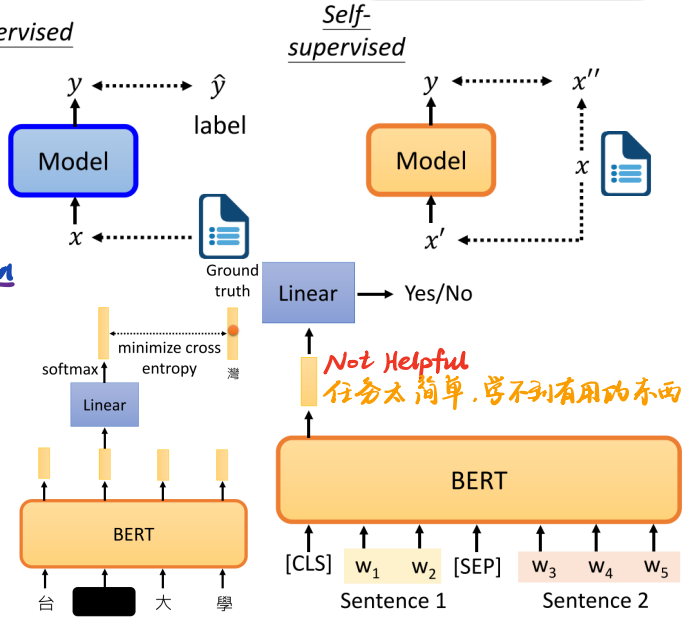


56

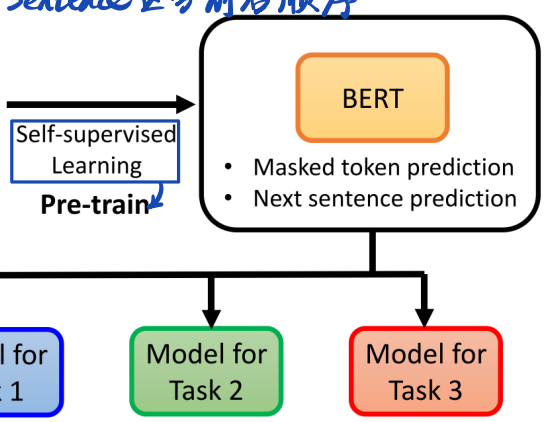
BERT

Self-supervised Learning → Pretrain 产生 BERT 的过程
→ 属于 Unsupervised Learning 的一种 unsupervised 指代过于泛
- 没有标注 → 取数据的一部分作 Input (x')，一部分作为标注 (x'')
- 目标：令 Output 和标注越接近越好（y 和 x''）

Masking Input https://arxiv.org/abs/1810.04805 → 学做填空
→ BERT 结构与 Transformer 的 Encoder 相同
- 随机遮住 Sequence 中的一部分 Token → 替换成 Special Token (MASK)
  → 随机换成另一个字
- Output 是一个概率分布 猜测 被 Mask 的 Token
  分类问题 → 类别与字典规模相同



Next Sentence Prediction
- Input 两个句子，取 CLS 处的 Output 判断两句是否相接
  → CLS Sequence 开头 SEP 两个句子之间的分隔符
- SOP: Sentence Order Prediction ALBERT
  → 两个 Sentence 互分前后顺序



评判 BERT 性能：GLUE General Language Understanding Evaluation
→ 在每个任务上微调，取平均值
- Corpus of Linguistic Acceptability (CoLA)
- Stanford Sentiment Treebank (SST-2)
- Microsoft Research Paraphrase Corpus (MRPC)
- Quora Question Pairs (QQP)
- Semantic Textual Similarity Benchmark (STS-B)
- Multi-Genre Natural Language Inference (MNLI)
- Question-answering NLI (QNLI)
- Recognizing Textual Entailment (RTE)
- Winograd NLI (WNLI)

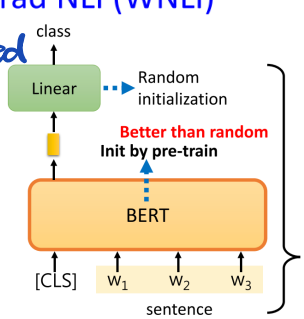
Super Glue 更难的任务集

How to use BERT

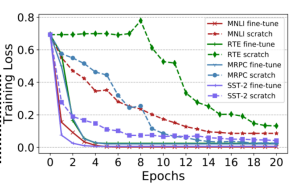
(self) Pretrain + (supervised) Finetune semi-supervised

Case 1：Classification sequence → class
- 取 CLS 对应的 Output 做 Linear 变换后进行分类
- 训练时，Linear 和 BERT 都会进行 Update
- 初始化，Linear 随机，BERT 由 Pre-train 初始化

Scratch 整个 Model (BERT & Linear) 都是随机初始化
→ Loss 下降慢且最小值更大



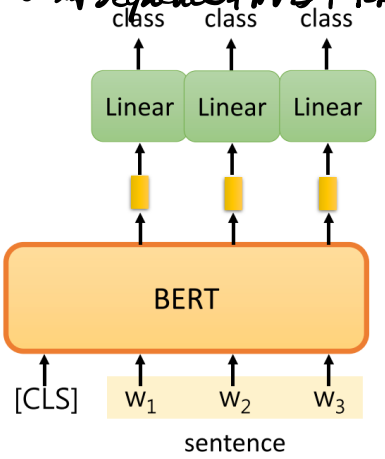
Input: sequence
output: class
Example: 情感分析
Sentiment analysis
this is good
positive
This is the model to be learned.



实线：Pre-train
虚线：Scratch

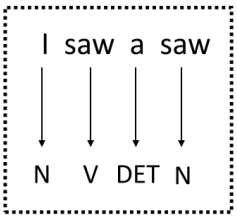
Case 2：Seq2Seq　Sequence → Sequence

- Input & Output 都是 Sequence，且长度相同
- 对 Sequence 中的每个 Token 的 Output 做 Classification



Input: sequence
output: same as input

Example:
POS tagging 词性标注



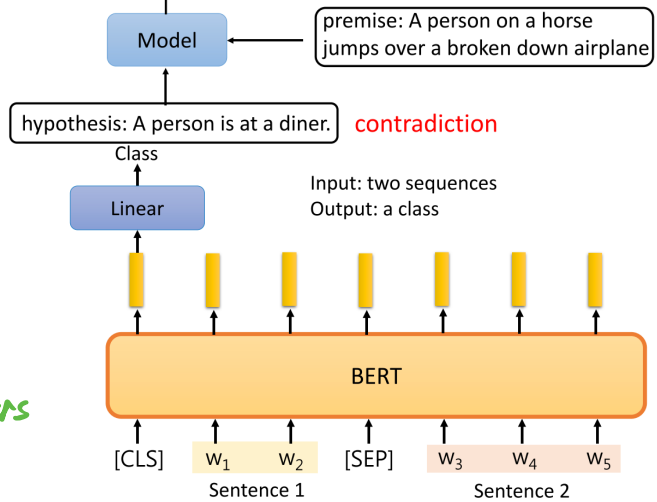
Case 3：多 Sequence 分类　2 sequences → 1 class

判断从前提能否推出假设
Premise　Hypothesis
e.g. 立场分析

contradiction
entailment
neutral

Input: two sequences
Output: a class

Example:
Natural Language Inferencee (NLI)



Input: two sequences
Output: a class

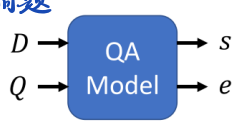
Case 4：Question Answering　2 Sequences → 2 integers

→ 给机器一篇文章，提问 & 回答 Answer 一定在文章中

- Extraction-based Question Answering (QA)

Document: $D = \{d_1, d_2, \cdots, d_N\}$　文章

Query: $Q = \{q_1, q_2, \cdots, q_M\}$　问题



output: two integers (s, e)

Answer: $A = \{d_s, \cdots, d_e\}$　Answer 第一个字　最后一个字

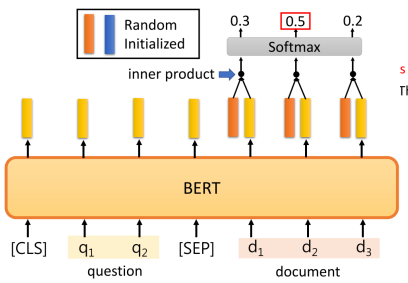
In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain 77 atte 79 cations are called "showers".

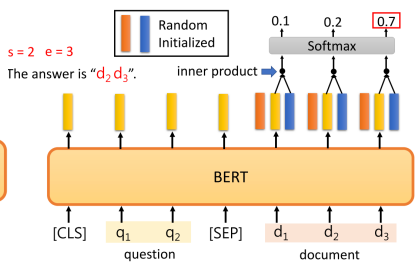
What causes precipitation to fall?
gravity　$s = 17, e = 17$

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud　$s = 77, e = 79$

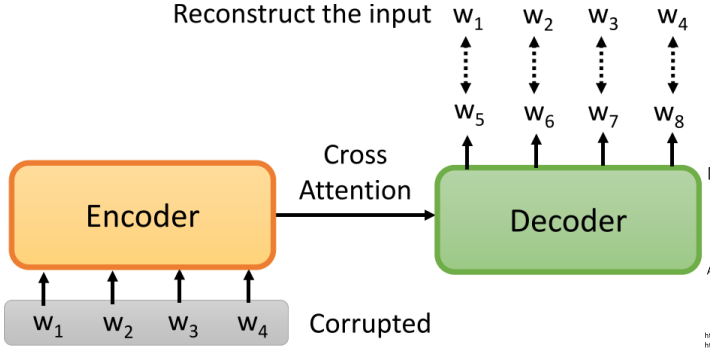
- Random Initialize (从头开始训练) 两个向量
- 预测 Answer 在文档中的开始和结束位置
- Sentence 长度有限制　不超过 512

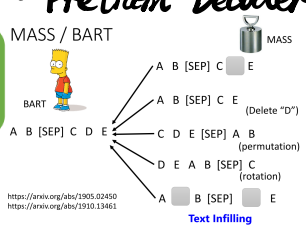




Pretrain Seq2Seq Model　Encoder + Decoder

- Bert 又 pretrain 了 Encoder
- Pretrain Decoder → 破坏 Input 再重建



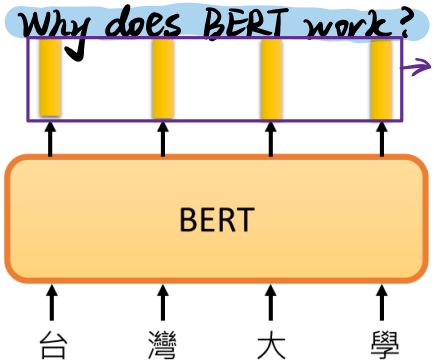


e.g. Mass, BART

比较各种破坏方法效果：T5

- Transfer Text-to-Text Transformer (T5)
- Colossal Clean Crawled Corpus (C4)

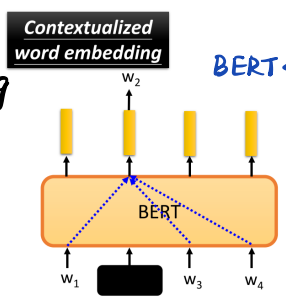
Why does BERT work?



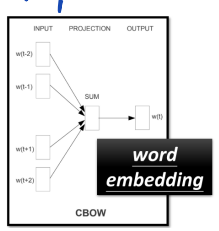
→ Embedding　Token 的向量化表示

- 意思相近的 token 具有相似的 Embedding
- 考虑上下文　Output 表达 Input 的意思

Contextualized Word Embedding
从上下文有的提取这个 Token 的特征
训练 BERT 做填空题 (Pre-train)
使其具有生成这个位置的表示向量的能力



BERT ⇔ Deep CBOW



BERT 用于 DNA 分类　效果非常好　可能 BERT 是一组效果很好的初始化参数

Multi-lingual BERT

在多种语言的语料集上训练 BERT

- English: SQuAD, Chinese: DRCD

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

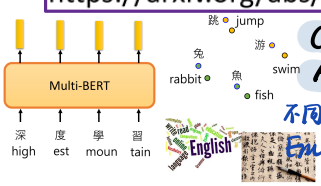
F1 score of Human performance is 93.30%

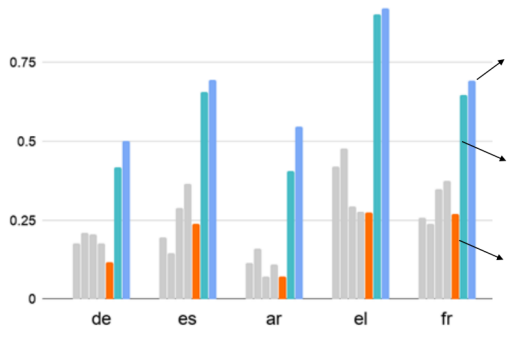
Zero-shot Reading Comprehension

Pre-train 使用多语言，Fine-tune 使用 English Test 使用中文，效果很好

→ Training：多语言填空，English 做 QA

https://arxiv.org/abs/2103.07162

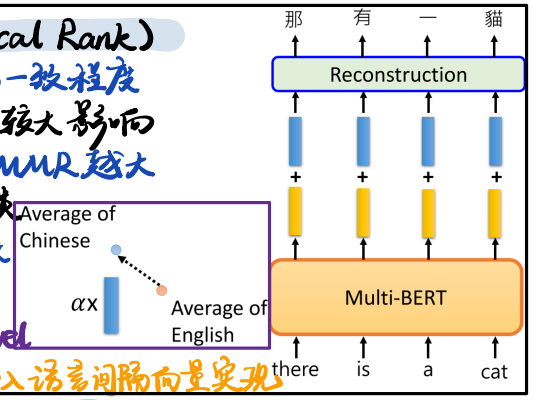




MMR (Mean Reciprocal Rank)

不同语义相同意思的一致程度

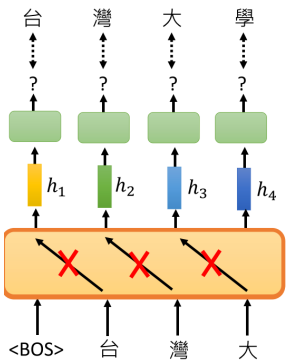
- 训练数据规模有较大影响
  Training Data 越多，MMR 越大
- 能够分辨语言的种类
  中文 → 中文　英文 → 英文
- 不同语言可相互转换
  → Unsupervised Token-level Translation 通过加入语言间隔向量实现

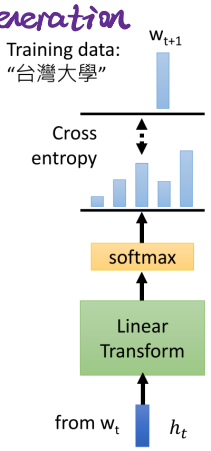


GPT 与 BERT 在 Pre-training 阶段的任务不同

Predict Next Token

- 用现有 Token 的 Embedding 预测下一个会出现的 Token
- 与 Decoder 相似，但是 Masked 不能看到后面会出现的 Token
- 可以用作生成 Generation





Beyond Text

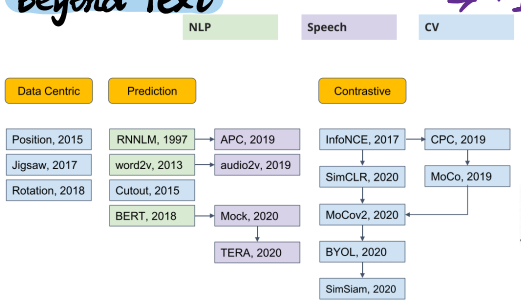
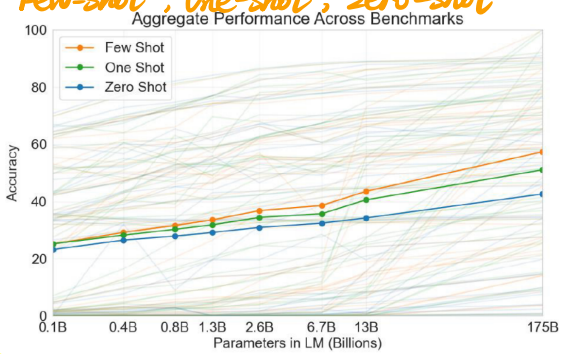


Image SimCLR、BYOL
https://arxiv.org/abs/2002.05709
https://github.com/google-research/simclr

How to use GPT?

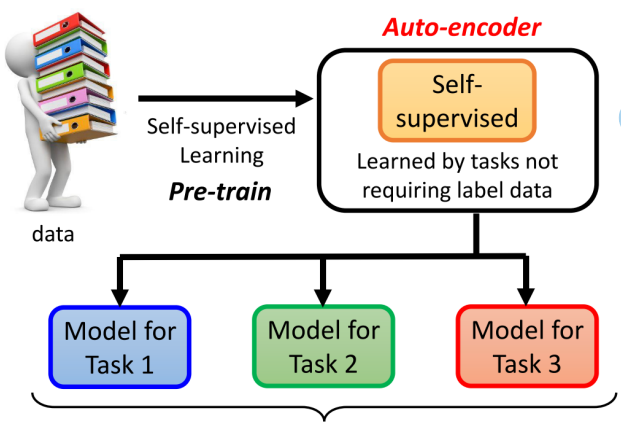
GPT 规模过大，Fine-tune 都较为困难
→ 给出任务的 Description 和少量 example
No Gradient Descent 没有调 GPT 参数
→ "In-context" Learning
Few-shot，One-shot，zero-shot



Average of **42** tasks

AUTO-ENCODER

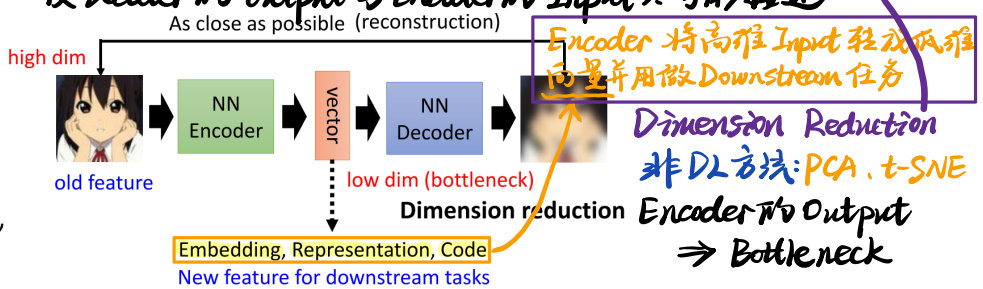
Self-supervised Learning Framework



Self-supervised Learning ⇔ Pretrain

- 不需要标注 Data 就可以进行训练，自行设计训练任务
- Auto-encoder 可以看作 Self-supervised Learning 的一种

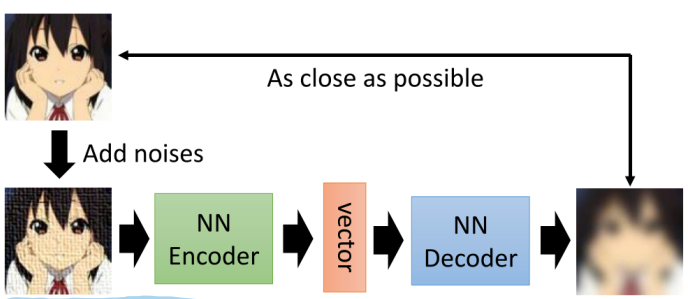
Reconstruction 重建 用简单方法表示复杂的图片
使 Decoder 的 Output 与 Encoder 的 Input 尽可能接近



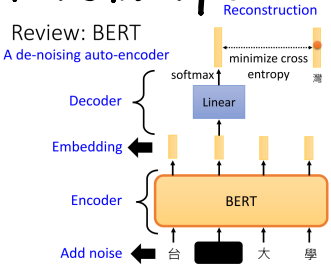
Encoder 将高维 Input 转成低维向量并用做 Downstream 任务

Dimension Reduction
非 DL 方法：PCA、t-SNE
Encoder 的 Output
⇒ Bottleneck

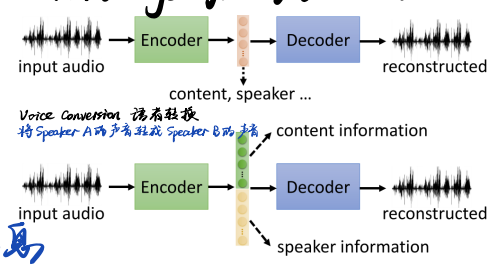
De-noising Auto-encoder



给 Encoder 的 Input 加上 Noise，令 Decoder 恢复无 Noise 的 Input → 与 BERT 的 mask 机制类似
Input Sequence 中有的 Token 被遮
⇒ 在原 Sequence 的基础上加 Noise



Linear 层 ⇒ Decoder Restruction
以取 BERT 前几层的 Output
⇒ 后面的 Layer 可视为 Decoder

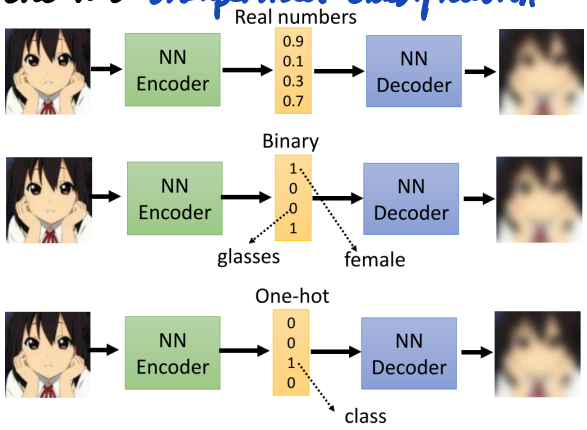


Feature Disentanglement
Encoder 的 Output (Representation) 中包含了 Input 的所有信息
由于高维的 Input 被转成了低维的 Representation 无法分辨 Representation 的各维度具体表示了 Input 的哪些信息
→ 在 Auto-encoder 训练时同时知道 Representation 各维度代表了什么信息

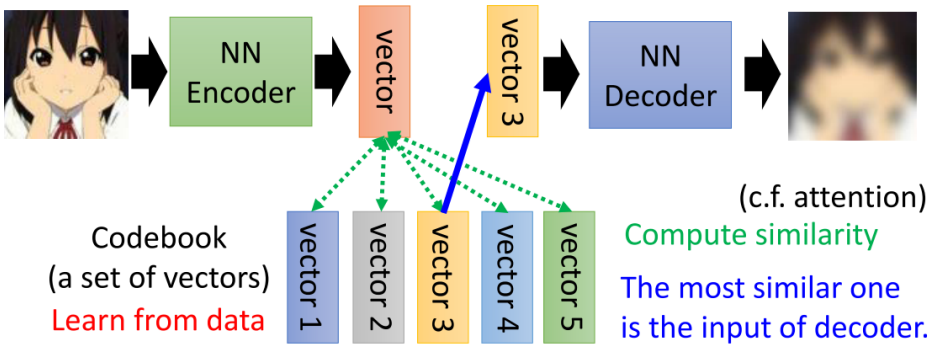
Discrete Latent Representation

Binary 更好地解释 Encoder 的 Output

One-hot Unsupervised Classification



• Vector Quantized Variational Auto-encoder (VQVAE)



VQVAE：无限的可能离散化 Codebook 中的 Vector

Phonetic Information

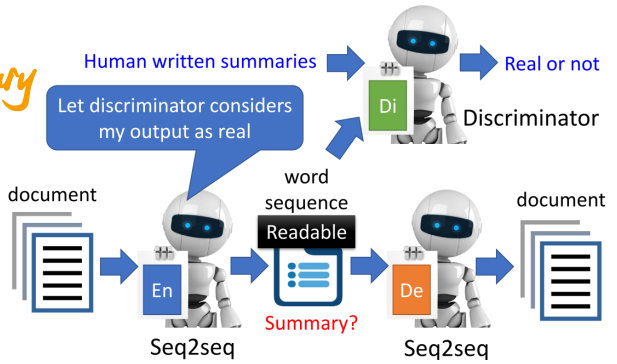
https://arxiv.org/pdf/1901.08810.pdf

Text as Representation

使用文字代替 Embedding 文章的 Summary

seq2seq2seq auto-encoder → Unsupervised Summarization

中间的 Summary not readable → 加入 Discriminator



Tree as Embedding

https://arxiv.org/abs/1806.07832

https://arxiv.org/abs/1904.03746

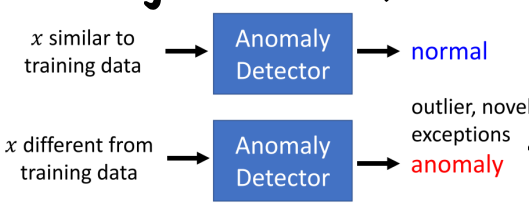
Applications

Generator 从已知 Distribution 随机生成 vector 给 Decoder

产生图片 → VAE (Variational Auto-encoder)

Compression Encoder 压缩，Decoder 解压缩 Lossy 图片失真

Anomaly Detection 异常检测 • 检测 Input x 是否与训练数据相似



→ Fraud Detection 信用卡欺诈

→ Network Intrusion Detection 网络安全

→ Cancer Detection

- 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
- Network Intrusion Detection
  - Training data: connection, x: attack or not
  - Ref: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- Cancer Detection
  - Training data: normal cells, x: cancer or not?
  - Ref: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

• One-class Classification Reconstruction Loss 判断 anomaly

绝大部分 Training Data 都是 Positive，负样本很难收集