Self-supervised Learning

Unsupervised Learning: there is no labels y, the targets are missing

- in classical ML, used for density estimation, dimensionality reduction (e.g. PCA, t-SNE), and clustering (K-means)...
  Self-supervised Learning: there is still no labels y, but the model replace the targets with the input x
- used for pre-training

Two NLP pre-training models: BERT and GPT

BERT

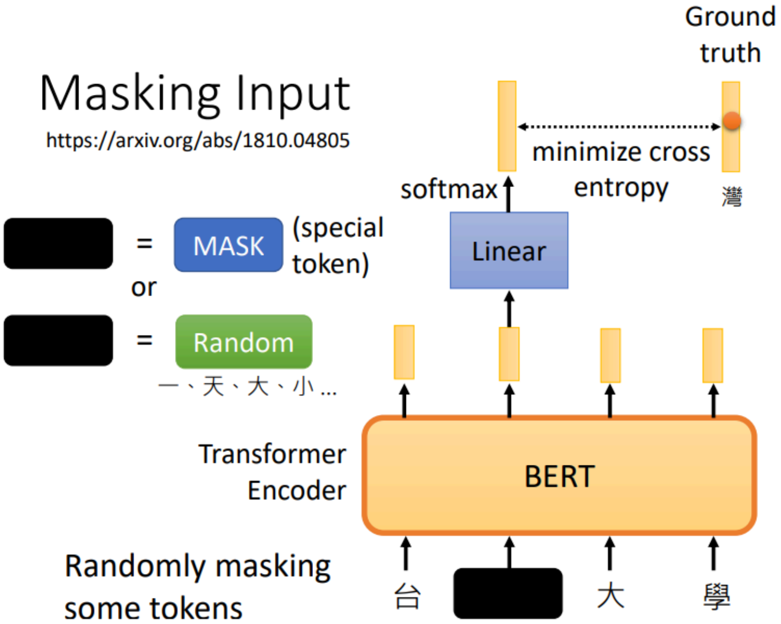
Bidirectional encoder representations from transformers
Basically, BERT is a transformer encoder, so the length of input sequence is equal to the length of output sequence.

Pre-train (self-supervised) + Fine-tune(supervised) --> BERT (semi-supervised)

Pre-train

BERT was pre-trained in 2 tasks:

- Masking Language Model: predict the randomly masked token given its context



  the way of masking is also ramdomly chosen:

    - replace the token with a special token [Mask]
    - replace the token with another random token
- Next Sentence Prediction: predict if the two sentence appeared sequentially in the training corpus
  2 special tokens in NSP
    - [CLS] (for classify) to mark the beginning of the sentence (in pre-train) and represent the sentiment information (in fine-tuning)
    - [SEP] (for separate) to separate sentences

Fine-tune

We can find-tune the pre-trained BERT to gain the capability in **downstream tasks**

Task sets: a collection of well-defined tasks used to evaluate the performance of pre-trained models
e.g. General Language Understanding Evaluation (GLUE)

How to fine-tune

1. Pre-processs the input seq
    1. add [CLS] at the very beginning
    2. add [SEP] between the seqs if there are more than 1 seq
2. BERT encodes tokens into embeddings which contains information extracted form the context
    - **e.g. the token embedding of [CLS] represents the information of the whole seqeunce, including the sentiment information**
3. **Select embeddings according to your understanding of the downstream task.
    - e.g. select the embedding fo [CLS] in sentiment analysis task
4. Apply a linear (random initialized) followed by a Softmax layer on the selected embedding
    - The linear layer functions as the Fully Connect Layer
    - Other activation functions are rarely used

Finetuned tasks for BERT:

- Sentiment analysis (Seq2Class)
    - Apply a linear (random initialized) and a Softmax to the embedding of [CLS]
- POS tagging (Seq2Seq)
    - Apply a linear (random initialized) and a Softmax to the embeddings of the sentence
- Natural Language Inferencee (NLI) (two Seqs --> a class)
  Given a premise and a hypothesis, determine whether it is a contradiction, entailment and neutral

    - Apply a linear (random initialized) and a Softmax to the embedding of [CLS] that encodes the information of two sentences

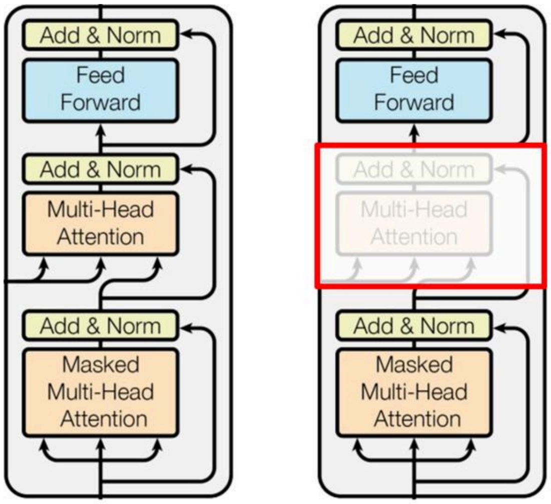
- Extraction-based Question Answering (two Seqs --> two integers)
  Given the document (context) and the question related to this context, determine the start and end positions of the answer in the document
  1. Random initialize two vectors
  2. Compute the inner products between these vector and the document embeddings
  3. Apply a Softmax Layer to get probablity distribution for the start and end positions respectively

Why BERT works

The conventional explanation: A word's meaning is determined by the company it keeps (the context). BERT learn the word's meaning in Masking Language Model task, so it can understand each word well.
Continous Bag of Words (CBOW): a ML model that predicts the masked token by the context. BERT can be seen as the DL version of CBOW.

GPT

Generative Pre-training Transformer
Basically, GPT is a transformer decoder.
The second Multi-Head Attention layer was used for learning from the encoder, so in GPT, this layer was removed.
The Masked Multi-Head Attentions layer was retained, so GPT is also autoregressive.



Transformer Decoder

GPT Decoder
去掉第二个 Multi-Head

Pre-train

Task: predict next token by the previous tokens' embeddings and positional embeddings
GPT is pre-trained in a huge unlabelled dataset (self-supervised).
After pre-training, GPT build a generative model of the language.

Fine-tuning

supervised discriminative fine-tuning on spacific tasks
training data:

- task description
- examples

SSL for speech and image

SSL for speech Task sets: Speech processing Universal Performance Benchmark (SUPERB)

1. Generative Approach
   replace the seq input of pre-training
   - BERT series for speech
     - Masking strategies should be altered since the speech seq is continuous
       - masking consecutive features
       - masking specific dimensions of all tokens (learn more speaker information)
     - representative model: Mockingjay
   - GPT series for speech
     - Predicting strategies for speech
       - predict the 3rd token after
     - Autoregressive Predictive Coding (APC)
2. Predictive Approach
   speech and images contain many details that are difficult to generate, so we alter the pre-training tasks to predictive ones
   - Image: Rotation Prediction, Context Prediction
   - Speech: HuBERT
3. Contrastive Learning
   alter the pre-training tasks to constrasting
   Image:
   - SimCLR, use Data Augmentation to obtain the positive pairs
   - MoCo
   - MoCo v2
     Speech:
   - CPC

    - VQ-wav2vec + BERT
    - Wav2vec 2.0

4. Bootstrapping
    - Image: Bootstrap your own latent (BYOL), Simple Siamese (SimSiam)
    - Speech: Data2vec
5. Simply Extra Regularization
    - Image: Barlow Twins, VICReg
    - Speech: DeLoRes