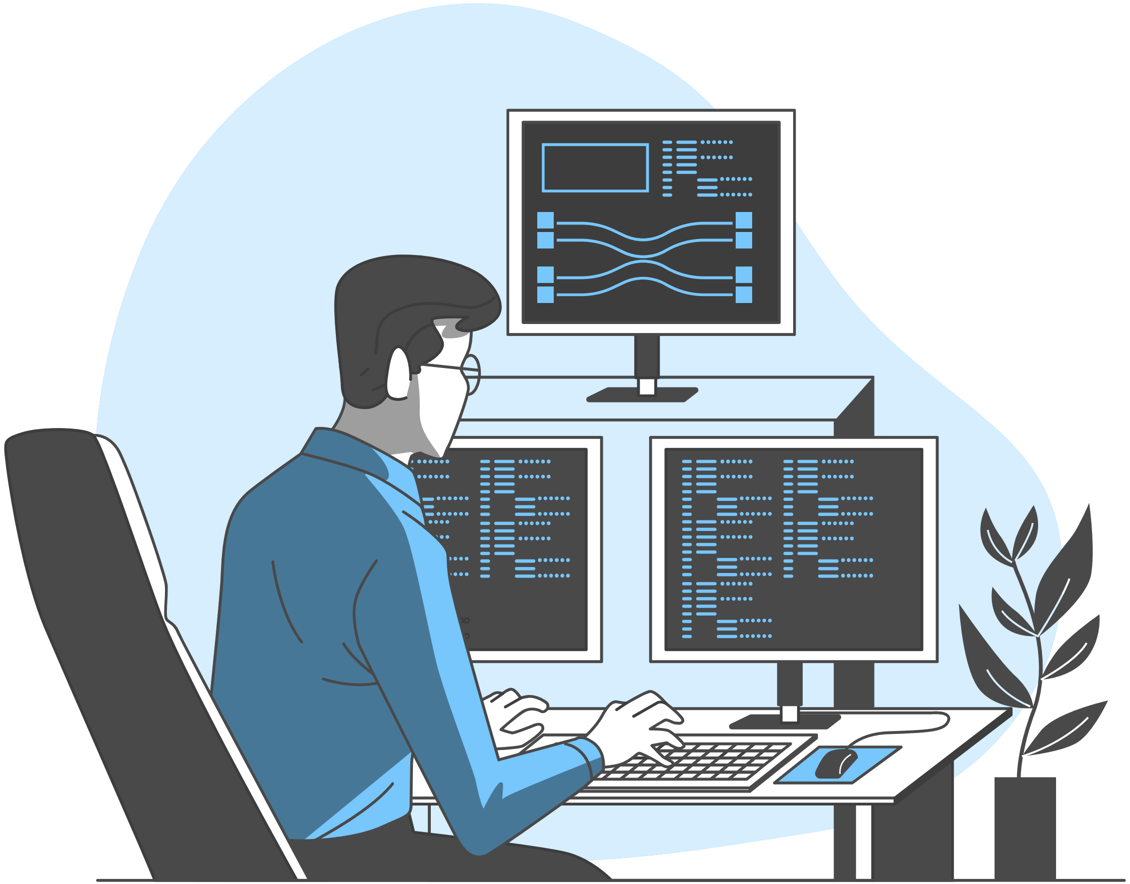
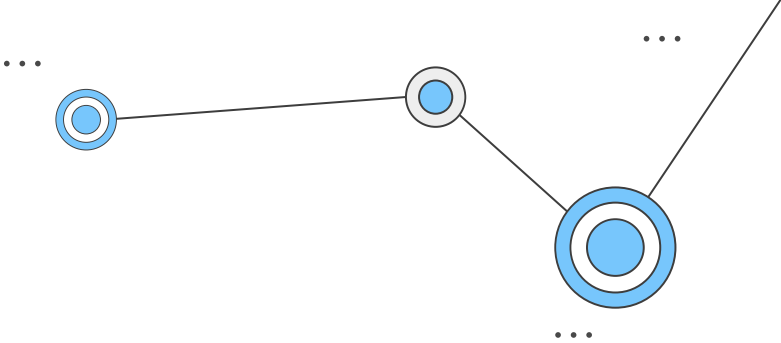




Bert

2023/04/27

Bidirectional Encoder Representations from Transformers

Transformer

Encoder

Word → Representation (vector)

E.g.

Education/Love/Job

↓

0/1/2

LabelEncoder()

Bidirectional Encoder Representations from Transformers

Transformer

Word

Representation (vector)

Task:cloze task, or masked language model, MLM

Bidirectional: jointly conditioning on both left and right context

I_____dogs and cats.

Bidirectional Encoder Representations from Transformers

Transformer

Word ⇄ Representation (vector)

input sequence: token by token (X)
input sequence: the entire sequence (O)
now the model can be accelerated by the GPUs
⇒ less time consuming

Bidirectional Encoder Representations from Transformers

Transformer

Word ⇄ Representation (vector)

We don’t need labeled data to pre-train these models.

Bidirectional Encoder Representations from Transformers

Transformer

Word

Representation (vector)

Task:cloze task, or masked language model, MLM

Bidirectional: jointly conditioning on both left and right context

I_____dogs and cats.

Model Fine-Tuning

The process that trains the pre-trained model (trained on a huge dataset) on our relatively smaller dataset.

Train the entire architecture
Feed the output to a softmax layer:
The error is back-propagated through the entire architecture and the pre-trained weights of the model are updated based on the new dataset.

Model Fine-Tuning

The process that trains the pre-trained model (trained on a huge dataset) on our relatively smaller dataset.

Train partially:

Keep the weights of initial layers of the model frozen while we retrain only the higher layers. (test and try)

Model Fine-Tuning

The process that trains the pre-trained model (trained on a huge dataset) on our relatively smaller dataset.

<u>Train the new ones:</u>
Freeze all the layers of the model and attach a few neural network layers of our own.
Weights updated: the attached layers

Pre-processing

Bidirectional Encoder Representations from Transformers

Transformer

Encoder

Word → Representation (vector)

E.g.

tokenizer(“睡覺要廢”)

tokenizer()

Tokenization unit: character

{'attention_mask':[1,1,1,1,1,1],

CLS SEP

'input_ids':[101,3152,3315,2968,1242,102],

‘Token_type_ids: [0,0,0,0,0,0]}

Bidirectional Encoder Representations from Transformers

Pre-processing

Transformer

Encoder

Word → Representation (vector)

E.g.
tokenizer(“魑魅魍魎”)

tokenizer()

Out of Vocabulary(OOV)

CLS UNK UNK SEP

'input_ids':[101, 100, 7791, 7793, 100, 102]

Bidirectional Encoder Representations from Transformers

Pre-processing

Transformer

Word → Encoder → Representation (vector)

E.g.

tokenizer(['貓追狗', '貓追老鼠'])

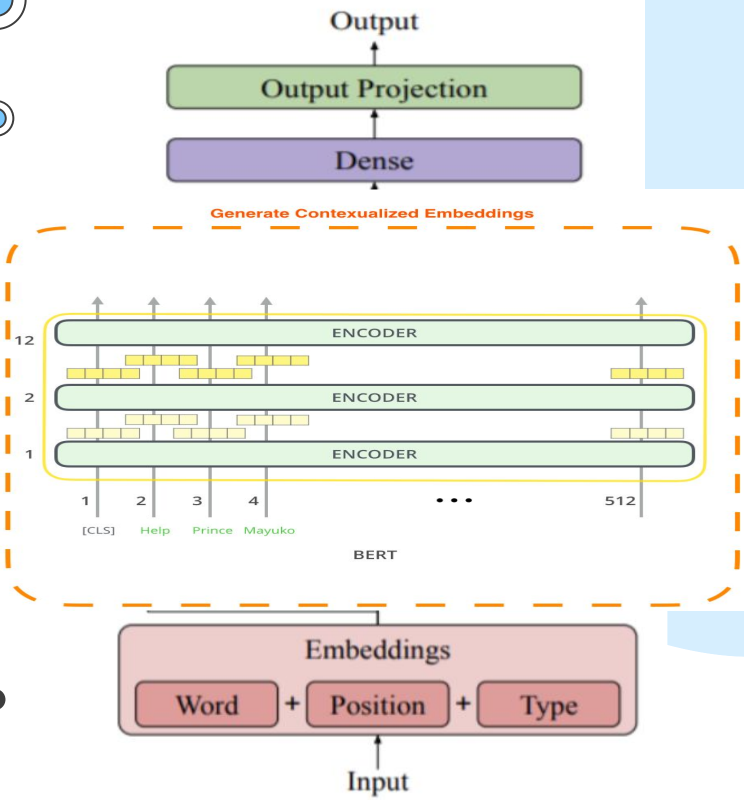
tokenizer()

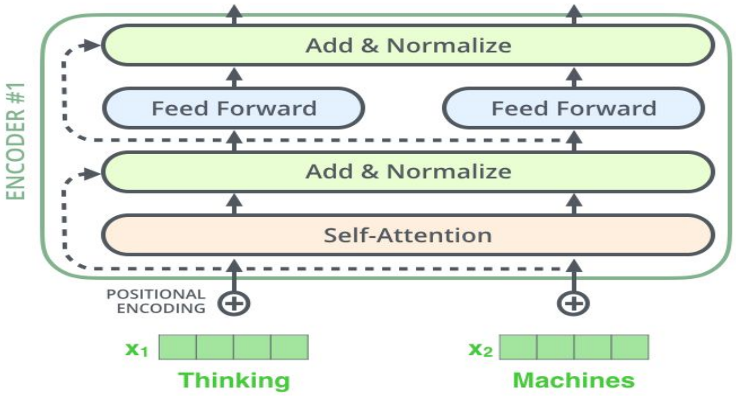
padding

PAD

'input_ids':[101, 6506, 6841, 4318, 102, 0],
[101, 6506, 6841, 5439, 7962, 102]

Bidirectional Encoder Representations from Transformers





Bidirectional Encoder Representations from Transformers

Model training

Training Arguments:

- learning_rate (LR):
  最重要的參數，通常在BERT裡是1e-5~1e-4左右。可以想成模型在更新參數時有多「衝動」

- batch_size: 每次模型要處理幾句，愈多句速度愈快，訓練效果也可能比較好。但愈多會耗愈多記憶體。

- num_train_epochs: 要把整個資料走過幾次。