

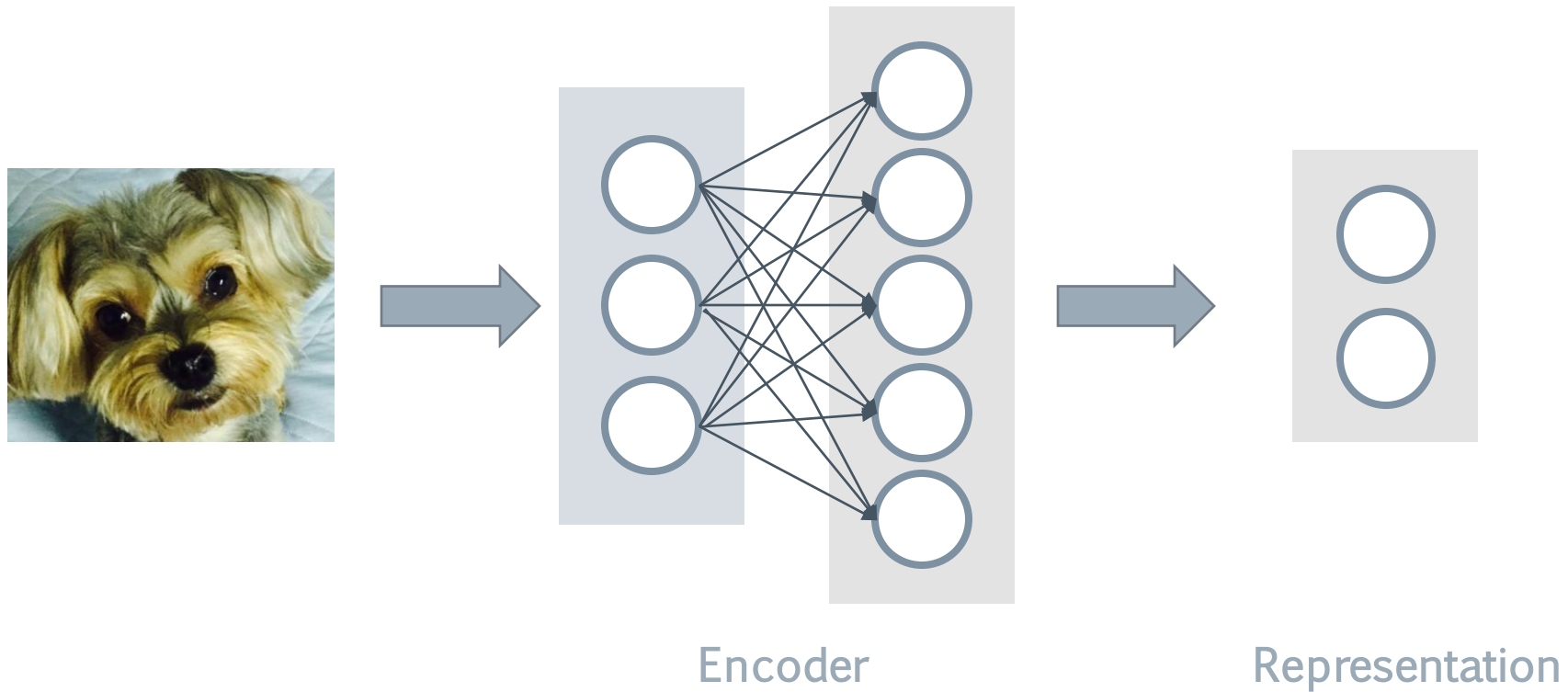
自然語言處理

Seq2Seq model

Instructor: 馬豪尚

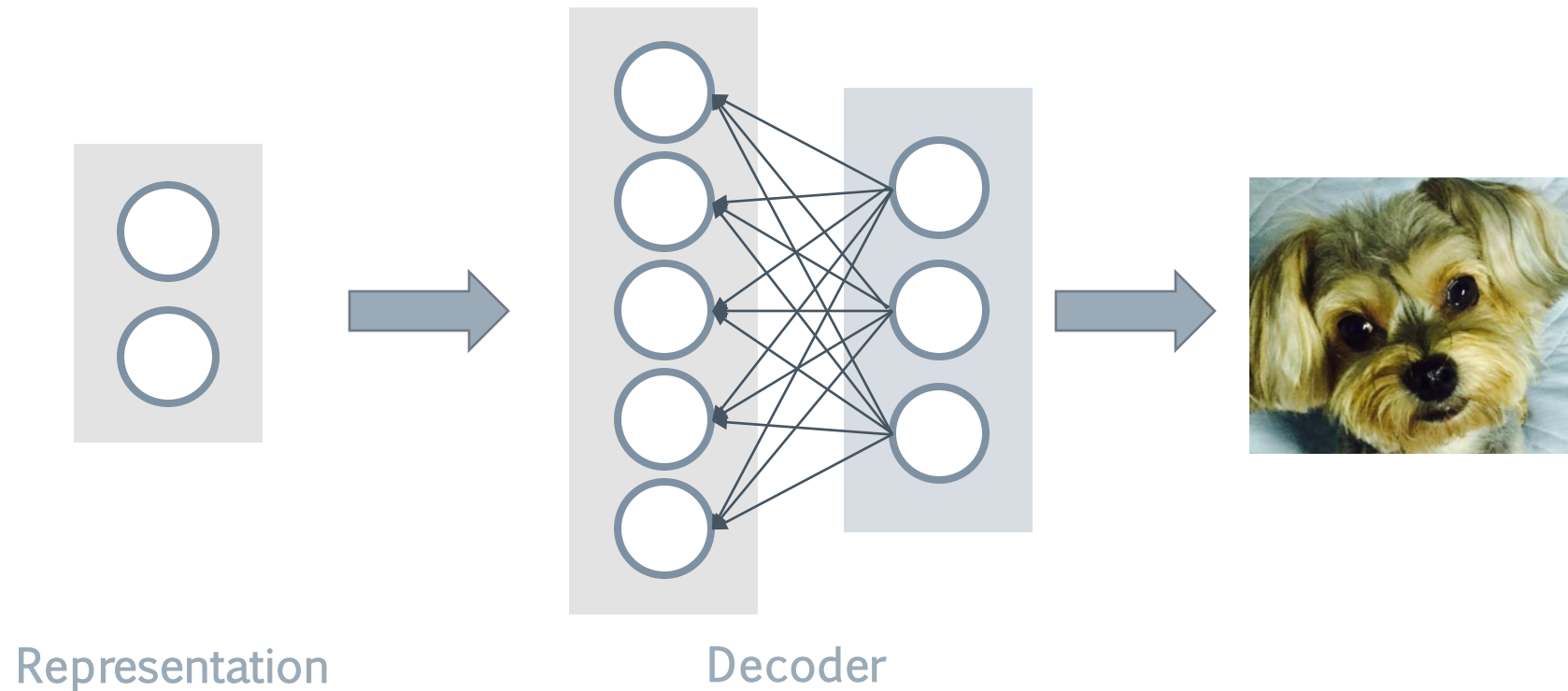
Auto-encoders ?

› 編碼 → 將輸入編碼成潛在表示法(嵌入向量)



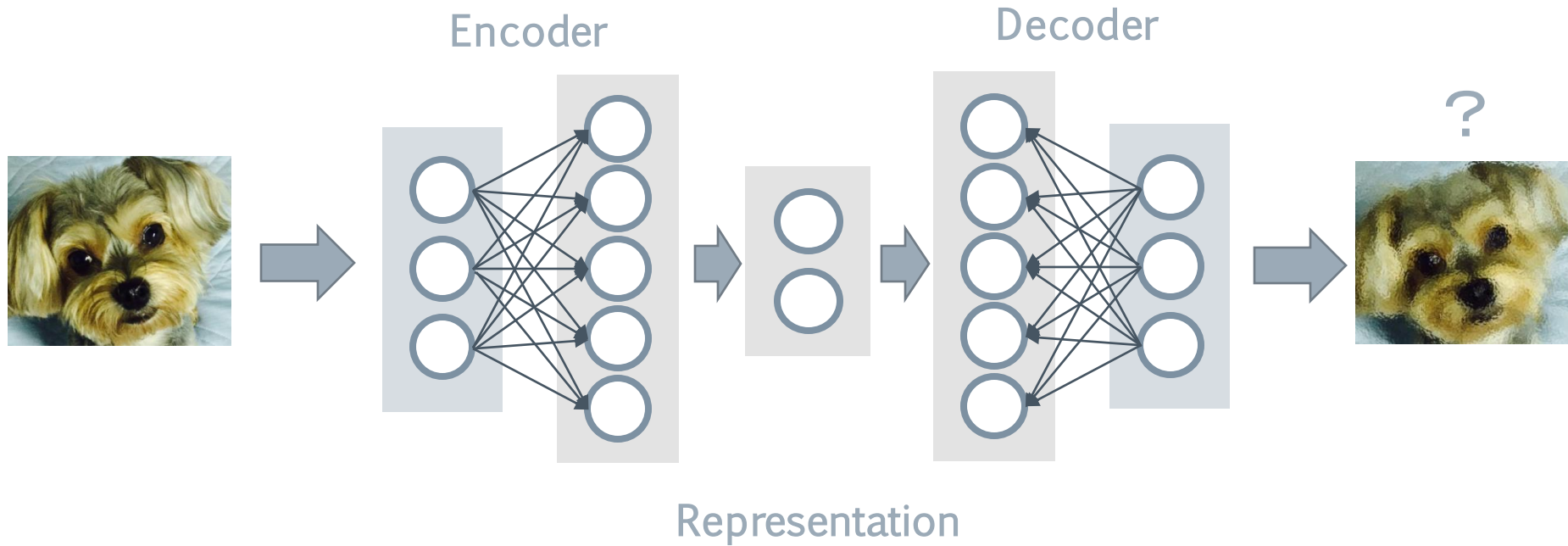
How to Check the Code

- › 需要一個解碼器透過解析潛在表示來重建或還原對象



Auto-encoders的目標

- › 檢查重建的物件是否與原始物件相似



Encoders and Decoders

› Encoder

- 將輸入資料的特徵進行編碼
- 使潛在表示法(嵌入向量)可以用來表示輸入資料

› Decoder

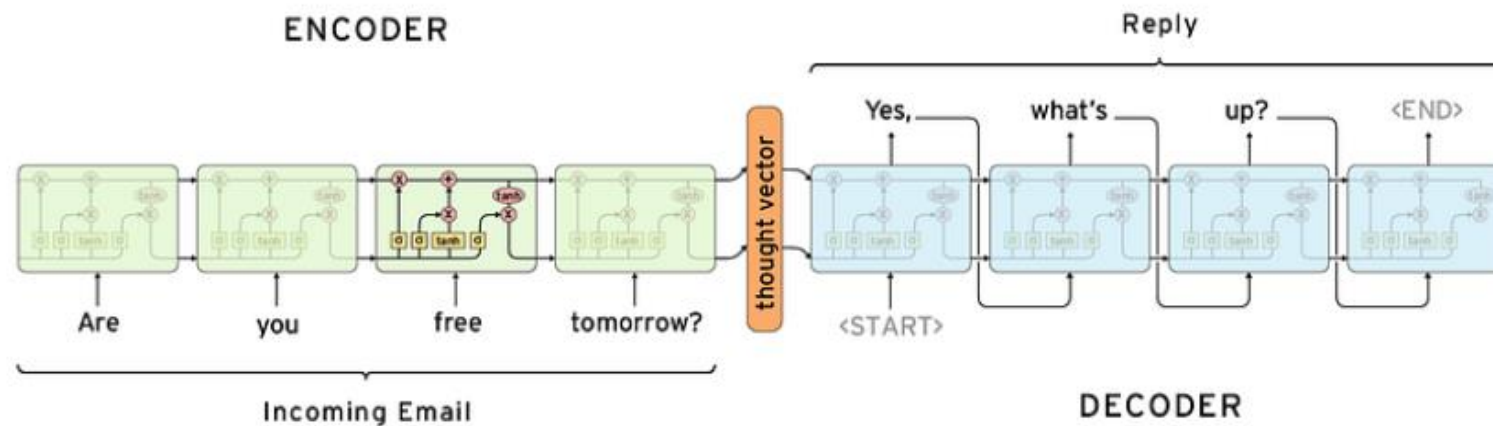
- 將潛在表示法(嵌入向量)重建成資料
- 使得重建(還原)資料可以相近於原始輸入資料

Seq2Seq 模型的架構

- › Seq2Seq 是由 Encoder 與 Decoder 兩個構成
 - Encoder 就是負責將輸入序列處理並轉成一個向量
 - 通常把這個向量稱為 context vector
 - 顧名思義，這個向量會囊括原序列的重要訊息
- › Decoder 則是根據 context vector 來生成文字

Seq2Seq 模型

- › Sequence to Sequence 將編碼器和解碼器串接起來
 - 第一個 RNN 負責將長度為 M 的序列給壓成 1 個向量
 - 第二個 RNN 則根據這 1 個向量產生出 N 個輸出
 - 這 $M \rightarrow 1$ 與 $1 \rightarrow N$ 接起來就構建出了 M to N 的模型



Encoder-Decoder架構特性

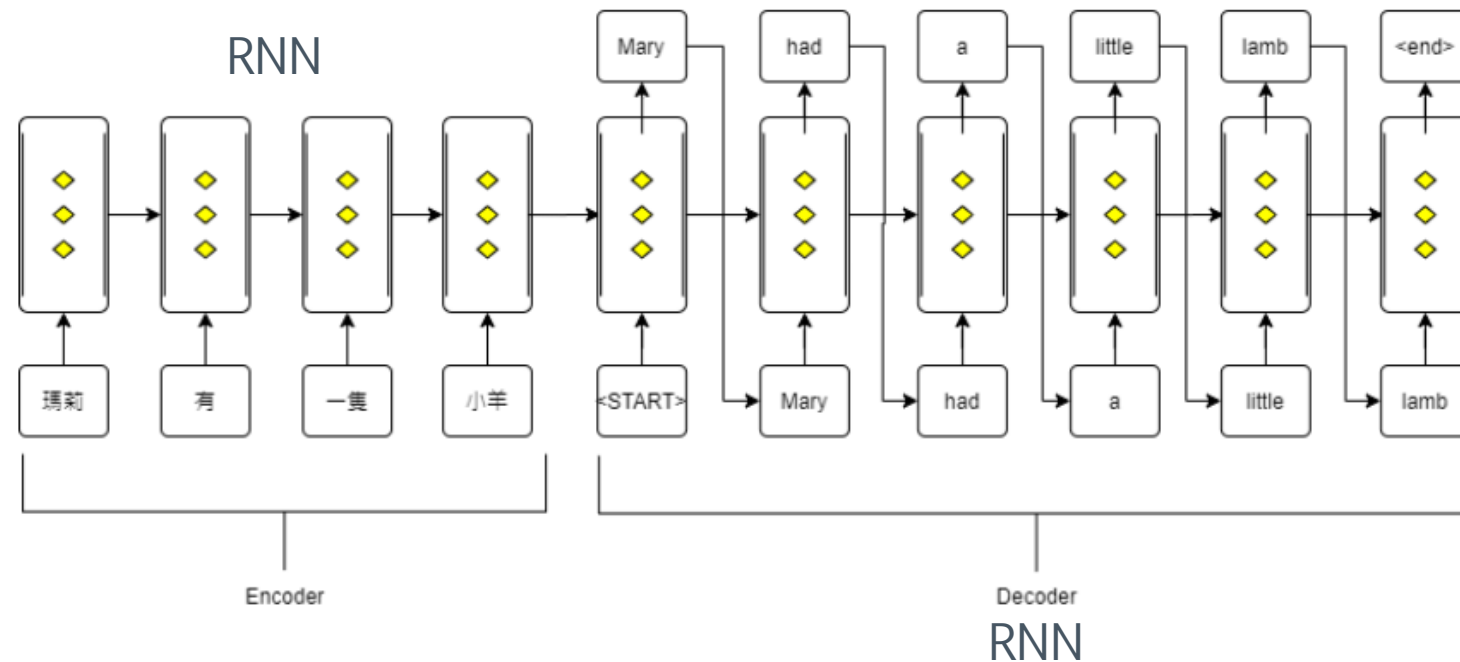
- › 不論輸入和輸出的長度是什麼，中間的「向量 c 」長度都是固定的
- › 根據不同的任務可以選擇不同的編碼器和解碼器（可以是 CNN/RNN/BiRNN/GRU/LSTM 等等）

機器翻譯任務

› 中翻英 NLP 的任務

- 輸入給定的中文被翻譯句
- 輸出為我們期望翻譯完成的英文翻譯句

› 一個典型的 sequence to sequence 模型應用

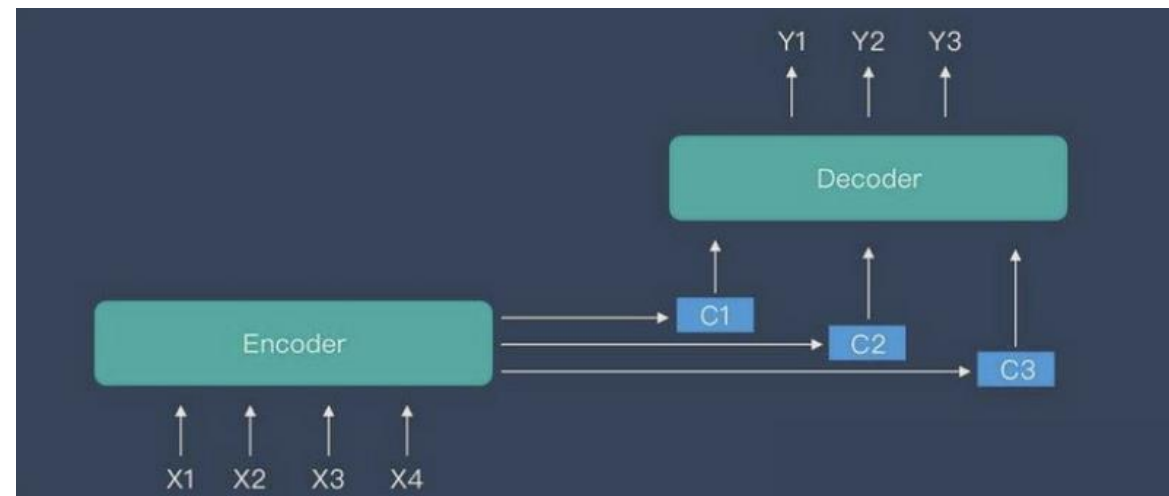


Seq2Seq機器翻譯模型問題

- › RNN是一個字一個字逐一運算的，
 - 經過多次循環神經單元計算後，被翻譯句句較前面的資訊可能會遺失
- › 每個字的產生，可能和輸入句序列中的某幾個字特別有關，而和其他的字相關比較少
 - Mary 這個字顯然和“瑪莉”這兩個字比較有關

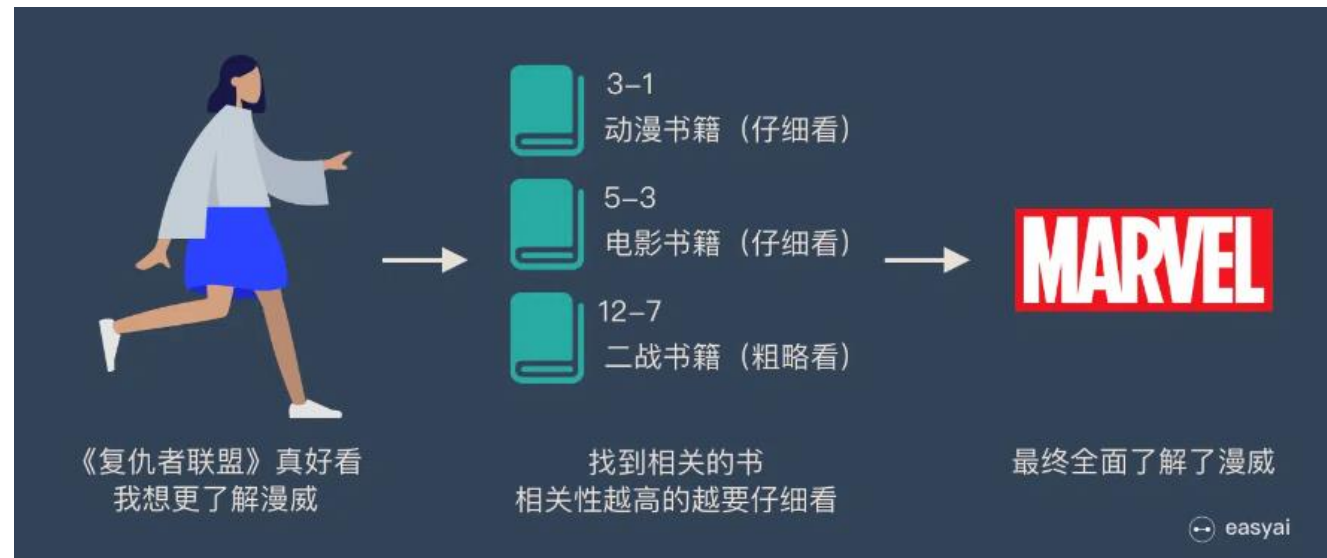
Attention 注意力機制

- › Attention 是為了解決以下困境
 - 資訊過長，資訊丟失
 - 對於不同輸入增加不同重要程度的機制
- › 不再將整個輸入序列編碼為固定長度的「中間向量C」，而是編碼成一個向量的序列
- › 產生每一個輸出的時候，都能夠做到充分利用輸入序列攜帶的資訊



Attention 注意力機制

- › 假設圖書館裡面有很多書 (value)
- › 為了方便查找，我們為書做編號 (key)
- › 當我們想要了解一件事情例如: 漫威 (query)
 - 我們就可以看看那些動漫、電影、甚至二戰 (美國隊長) 相關的書籍
 - 為了提高效率，並不是所有的書都會仔細看，針對漫威來說，動漫，電影相關的會看的仔細一些 (權重高)，但是二戰的就只需要簡單掃一下即可 (權重低)

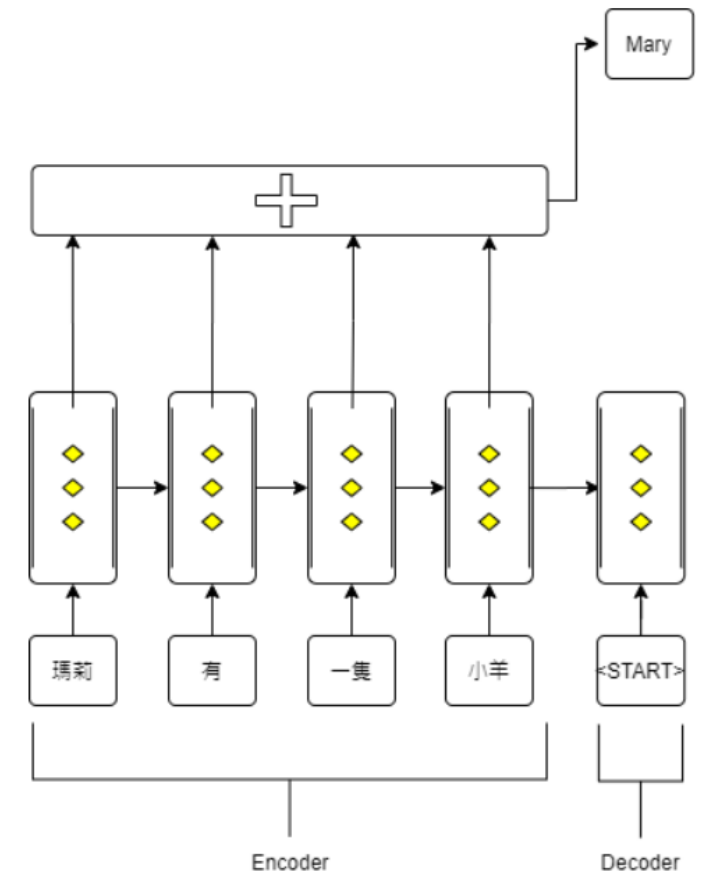


RNN seq2seq + Attention

- › 產生每個字的時候不論遠近都可以參考到輸入句每個字的表示向量
- › 對於相關性比較高的翻譯句輸入字，我們讓模型參照更多一點
- › 讓模型在處理文字時，將注意力只放在某些字詞上

RNN seq2seq + Attention

- › 為了讓模型在產生“Mary”的時候能看到輸入句的每個字的代表向量
 - 將輸入句的每個字的向量相加
 - 再將結果除以句子的長度，消除句長這個因素
 - 這樣等於可以參考輸入句子的每一個字(平均向量)

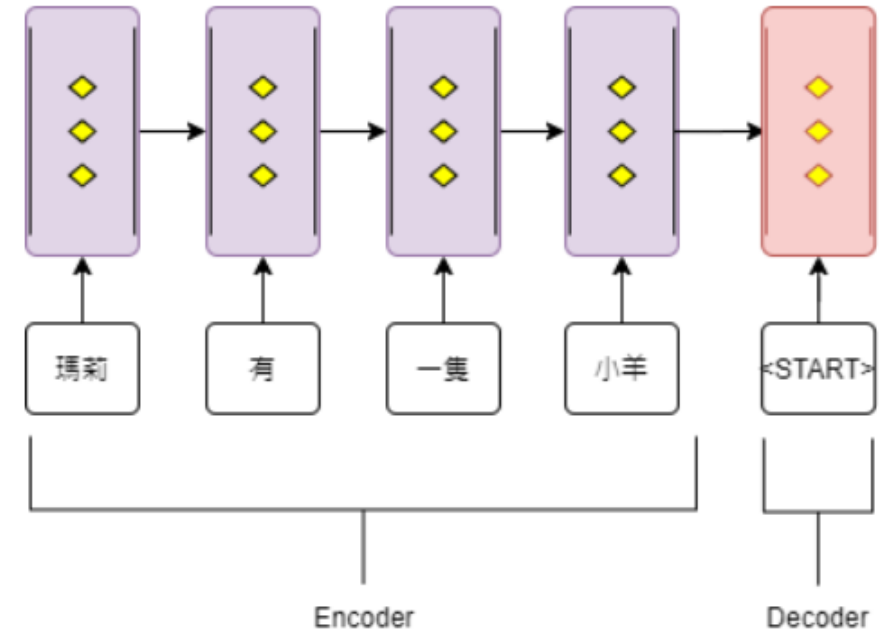


RNN seq2seq + Attention

- › 對於相關性比較高的翻譯句輸入字，我們要如何讓模型參照更多一點

$$attention(Q, K, V) = \sum_{i=1}^n similarity(Q, K_i) * V_i$$

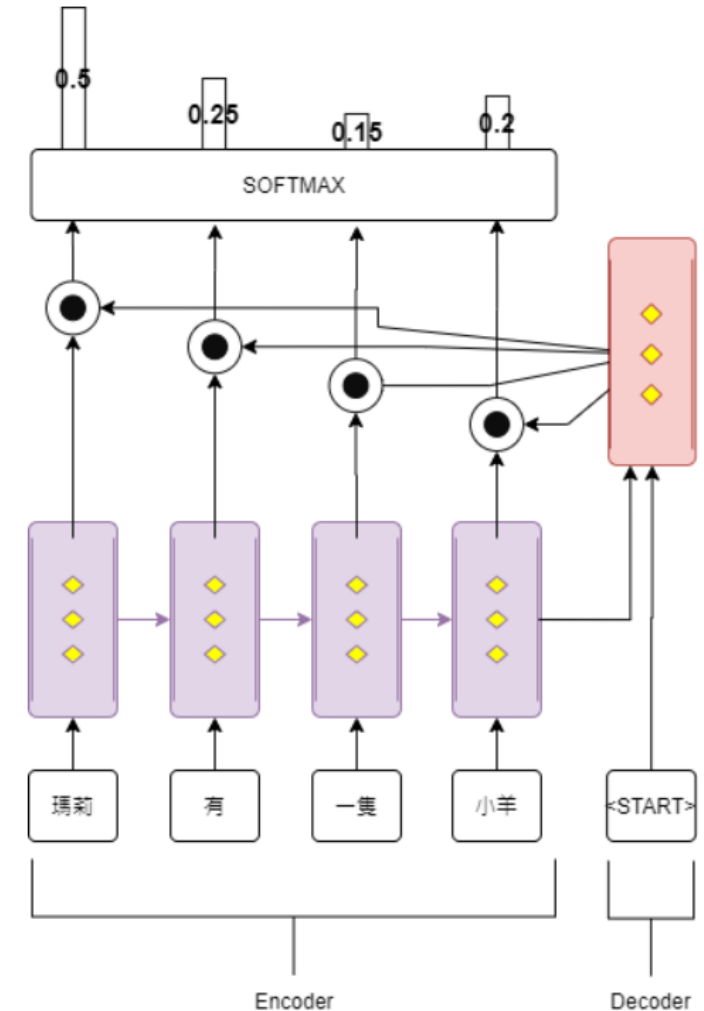
- Q : QUERY 是 我們要問的問題
- K : KEY 代表我們的輸入值被查詢的鍵值
- V : VALUE 代表我們的輸入值的內容
- Similarity() 是計算 Q 和 K 相關性的函數



RNN seq2seq + Attention

$$attention(Q, K, V) = \sum_{i=1}^n similarity(Q, K_i) * V_i$$

- › 第一步：query 和 key 進行相似度計算，得到權值
- › 第二步：將權值歸一化，得到直接可用的權重
- › 第三步：將權重和 value 進行加權求和



Keras Seq2Seq+Attention 模型

› 編碼器

```
from keras.layers import Input, LSTM, Embedding
from keras.models import Model

# 設定超參數
embedding_size = 256
encoder_hidden_units = 512

# 編碼器模型
encoder_inputs = Input(shape=(None,))
encoder_embedding = Embedding(input_dim=num_encoder_tokens, output_dim=embedding_size)
encoder_lstm = LSTM(encoder_hidden_units, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(encoder_embedding)
encoder_states = [state_h, state_c]
```

Keras Seq2Seq+Attention 模型

› 編碼器

```
decoder_hidden_units = encoder_hidden_units

# 解碼器模型
decoder_inputs = Input(shape=(None,))
decoder_embedding = Embedding(input_dim=num_decoder_tokens, output_dim=embedding_size)
decoder_lstm = LSTM(decoder_hidden_units, return_sequences=True, return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_embedding, initial_state=encoder_states)
```

Keras Seq2Seq+Attention 模型

› 注意力機制 (Attention Mechanism)

```
from keras.layers import Concatenate, Dense, Attention

# 注意力層
attention_layer = Attention()
attention_result = attention_layer([decoder_outputs, encoder_outputs])

# 將注意力的結果和解碼器的輸出合併
concat_layer = Concatenate(axis=-1)
decoder_combined_context = concat_layer([decoder_outputs, attention_result])

# 輸出層
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_combined_context)
```

Keras Seq2Seq+Attention 模型

› 編譯模型

```
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
```

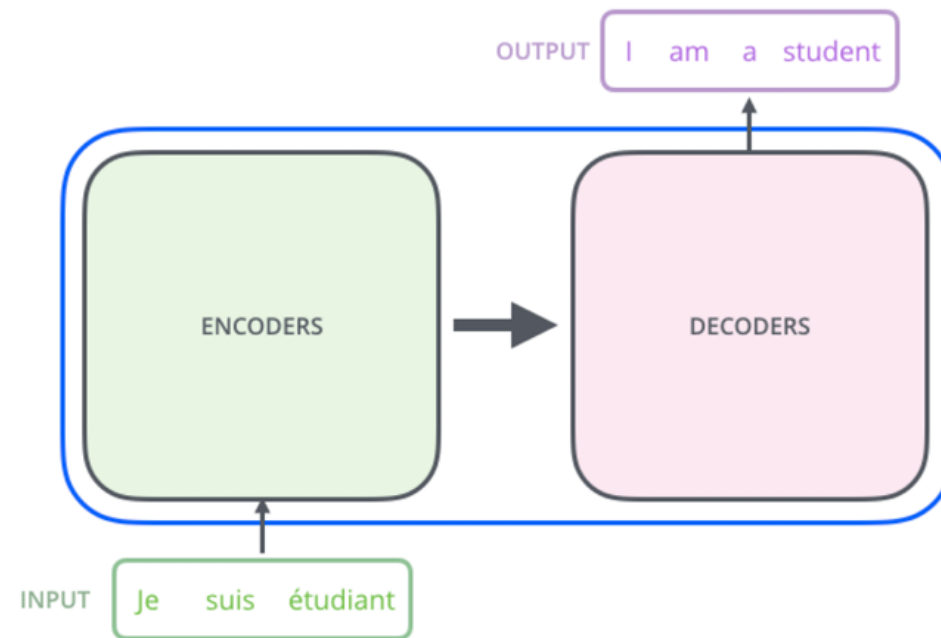
› 訓練模型

- 假設已經有處理好的輸入資料

```
model.fit([encoder_input_data, decoder_input_data], decoder_target_data,
          batch_size=batch_size,
          epochs=epochs,
          validation_split=0.2)
```

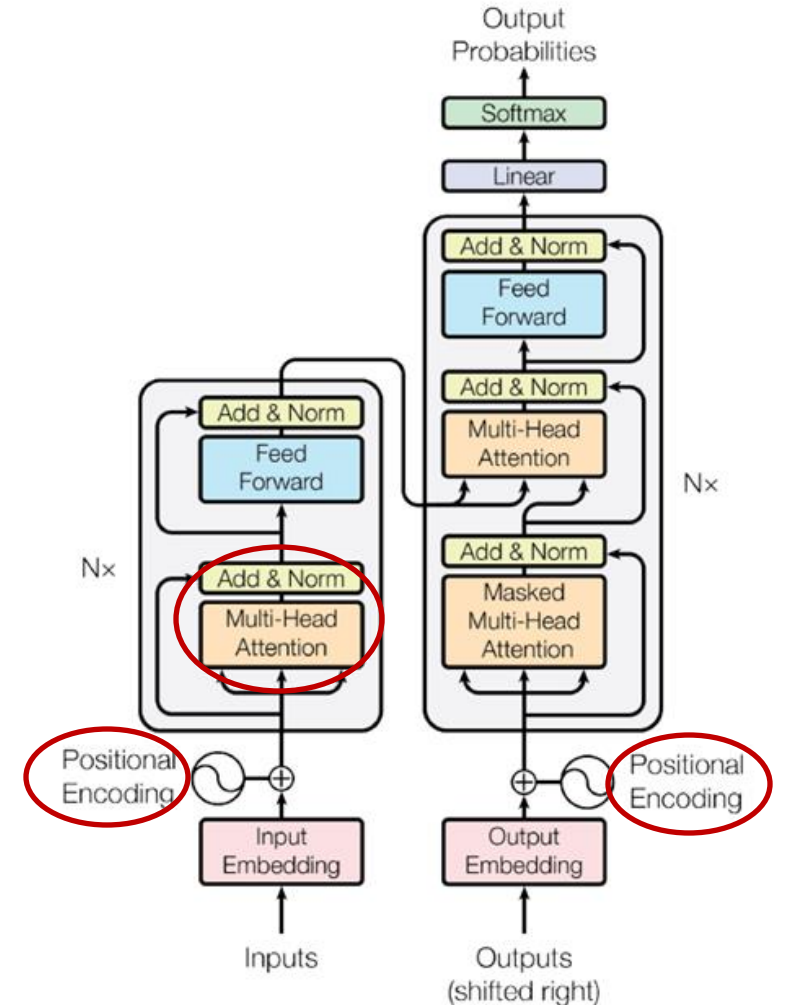
Transformer模型

- › Transformer 模型本來是為了翻譯任務而設計的。
- › 在訓練過程中，Encoder 接受原始語言的句子作為輸入，而 Decoder 則接受目標語言的翻譯作為輸入



Transformer特色

- › 編碼神經網路/解碼神經網路
- › 自注意力機制(self-attention mechanism)
- › 位置嵌入(positional embedding)



練習

- › 從<https://www.manythings.org/anki/>下載不同語言的 Sentence Pairs
- › 做資料前處理
- › 建立翻譯模型
- › 將資料輸入翻譯模型訓練
- › 測試模型(輸入一個句子→輸出成另外一個語言的句子)