**LANGUAGE TANSLATION**

CODE:

|  |  |
| --- | --- |
|  | import tensorflow as  from tensorflow.keras.preprocessing.text import Tokenizer  from tensorflow.keras.preprocessing.sequence import pad\_sequences  input\_sentences = ['I love coding', 'He is playing football']  target\_sentences = ['J\'aime coder', 'Il joue au football']  input\_tokenizer = Tokenizer()  input\_tokenizer.fit\_on\_texts(input\_sentences)  input\_seq = input\_tokenizer.texts\_to\_sequences(input\_sentences)  target\_tokenizer = Tokenizer()  target\_tokenizer.fit\_on\_texts(target\_sentences)  target\_seq = target\_tokenizer.texts\_to\_sequences(target\_sentences)  input\_seq = pad\_sequences(input\_seq, padding='post')  target\_seq = pad\_sequences(target\_seq, padding='post')  encoder\_inputs = tf.keras.layers.Input(shape=(None,))  encoder\_embedding = tf.keras.layers.Embedding(input\_dim=len(input\_tokenizer.word\_index)+1, output\_dim=256, mask\_zero=True)(encoder\_inputs)  encoder\_outputs, state\_h, state\_c = tf.keras.layers.LSTM(256, return\_state=True)(encoder\_embedding)  encoder\_states = [state\_h, state\_c]  decoder\_inputs = tf.keras.layers.Input(shape=(None,))  decoder\_embedding = tf.keras.layers.Embedding(input\_dim=len(target\_tokenizer.word\_index)+1, output\_dim=256, mask\_zero=True)(decoder\_inputs)  decoder\_lstm = tf.keras.layers.LSTM(256, return\_sequences=True, return\_state=True)  decoder\_outputs, \_, \_ = decoder\_lstm(decoder\_embedding, initial\_state=encoder\_states)  decoder\_dense = tf.keras.layers.Dense(len(target\_tokenizer.word\_index)+1, activation='softmax')  decoder\_outputs = decoder\_dense(decoder\_outputs)  model = tf.keras.models.Model([encoder\_inputs, decoder\_inputs], decoder\_outputs)  model.compile(optimizer='rmsprop', loss='sparse\_categorical\_crossentropy')  model.fit([input\_seq, target\_seq[:,:-1]], target\_seq[:,1:], batch\_size=64, epochs=50, validation\_split=0.2)Epoch 1/50 |

DATASET LINK

https://www.kaggle.com/datasets/devicharith/language-translation-englishfrench