Міністерство освіти і науки

України Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

ЗВІТ

лабораторної роботи №7

з курсу «Програмні засоби проєктування і реалізації неромережевих систем»

Тема: «Рекурентні нейронні мережі LSTM»

Перевірив:

Шимкович В. М.

Виконав:

Студент Гр. ІП-01 Шпилька В.С.

Київ 2023

Завдання: Написати програму, що реалізує рекурентну нейронну мережу LSTM для розпізнавання емоційного забарвлення тексту, використати датасет Yelp Dataset

1. Реалізація нейронної мережі:

```python
import tensorflow as tf

def LstmModel(vocab_size):
    model = tf.keras.Sequential([
        tf.keras.layers.Embedding(vocab_size, 64, mask_zero=True),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64,  return_sequences=True)),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32)),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ])

    return model
```

```
    model.summary()
[29] ✓ 0.0s

... Model: "sequential"
 _________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 64)          640128

 bidirectional (Bidirectiona  (None, None, 128)        66048
 l)

 bidirectional_1 (Bidirectio  (None, 64)               41216
 nal)

 dense (Dense)               (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 751,617
Trainable params: 751,617
Non-trainable params: 0
_________________________________________________________________
```

Нейронна мережа містить Embedding шар для кодування числа в багатовимірний вектор, далі декілька шарів lstm, які маю короткотривалу пам’ять та декілька лінійних шар для класифікації.

2. Створення пайплану для завантаження даних.

```python
def download_data(path = settings.DATA_PATH, val_percent = settings.VAL_PERCENT):
    dataset = tfds.load('yelp_polarity_reviews', data_dir=path, as_supervised=True)
    train_dataset, test_dataset = dataset['train'], dataset['test']

    train_size = len(train_dataset)
    val_size = int(train_size * val_percent)

    val_dataset = train_dataset.take(val_size)
    train_dataset = train_dataset.skip(val_size)

    return train_dataset, val_dataset, test_dataset

def create_train_pipeline(dataset, buffer_size = settings.BUFFER_SIZE, batch_size = settings.BATCH_SIZE):
    return dataset.shuffle(buffer_size).batch(batch_size).prefetch(tf.data.AUTOTUNE)

def create_test_pipeline(dataset, batch_size = settings.BATCH_SIZE):
    return dataset.batch(batch_size).prefetch(tf.data.AUTOTUNE)

def load_data(path = settings.DATA_PATH, val_percent = settings.VAL_PERCENT, buffer_size = settings.BUFFER_SIZE, batch_size = settings.BATCH_SIZE):
    train_ds, val_ds, test_ds = download_data(path, val_percent)

    train_ds = create_train_pipeline(train_ds, buffer_size, batch_size)
    val_ds = create_test_pipeline(val_ds, batch_size)
    test_ds = create_test_pipeline(test_ds, batch_size)

    return train_ds, val_ds, test_ds
```

Завантажуємо YelpDataset, виділяємо тренувальні, валідаційні та тестувальні дані, перемішуємо тренувальні та розбиваємо датасети на батчі.

Далі створюємо encoder для кодування тексту в числа, щоб надалі нейрона мережа могла трансформувати їх в багатовимірний вектор:

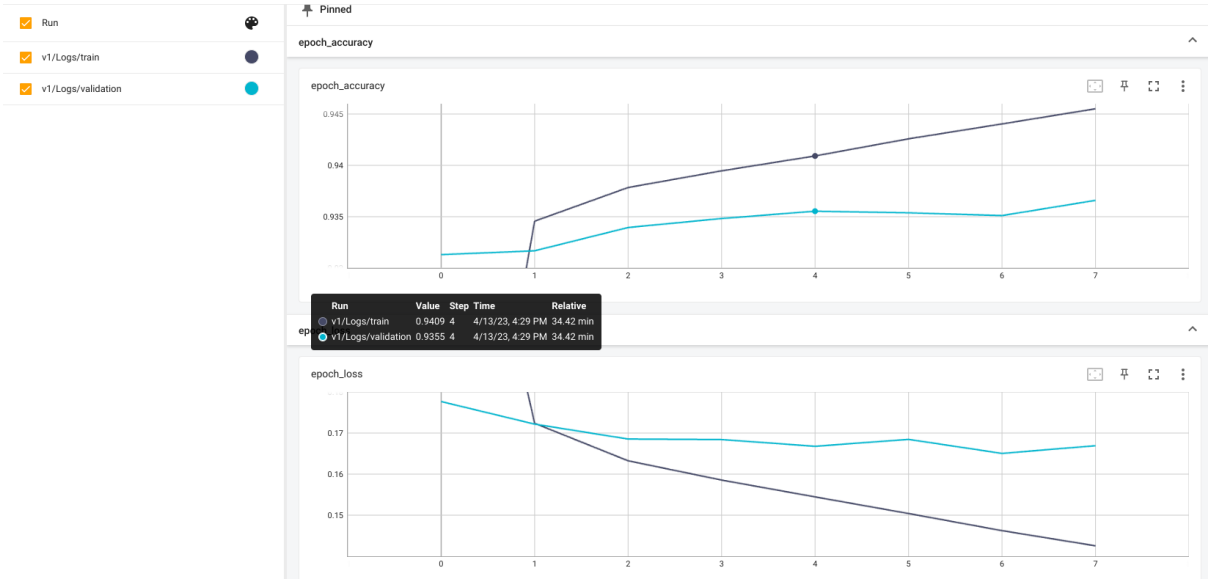
```
class YelpPreprocessing():
    def __init__(self, train_dataset) -> None:
        self.encoder = self.create_encoder(train_dataset)

    def create_encoder(self, dataset):
        encoder = tf.keras.layers.TextVectorization(
            max_tokens=settings.VOCAB_SIZE)
        encoder.adapt(dataset.map(lambda text, label: text))
        return encoder

    def vectorize_text(self, text, label):
        text = tf.expand_dims(text, -1)
        return self.encoder(text), label

    def get_encoder(self):
        return self.encoder
```

```
train_ds, val_ds, test_ds = YelpDataset.load_data(data_path, val_percent=settings.VAL_PERCENT, buffer_size=settings.BUFFER_SIZE, batch_size=batch_size)
preprocessor = YelpPreprocessing(train_ds)

train_ds = train_ds.map(preprocessor.vectorize_text)
val_ds = val_ds.map(preprocessor.vectorize_text)

vocab_size = len(preprocessor.get_encoder().get_vocabulary())
model = LstmModel(vocab_size)
```

Для оптимізації, даний інкодер створюється на етапі препроцесінгу і в нейронну мережу поступають тільки числа, щоб не виконувати це перетворення кожної епохи, але для збереження моделі можна приєднувати цей шар до основної нейронної мережі:

```
export_model = tf.keras.Sequential([
    self.encoder,
    self.model])

export_model.compile(loss=tf.keras.losses.BinaryCrossentropy(),
                     metrics=['accuracy'],
                     optimizer=tf.keras.optimizers.Adam(learning_rate=self.lr))

export_model.save(self.export_path)
```

Навчаємо нейрону мережу:



Майже відразу нейрона мережа досягла найкращої точності, а подальші покращенні є невеликими

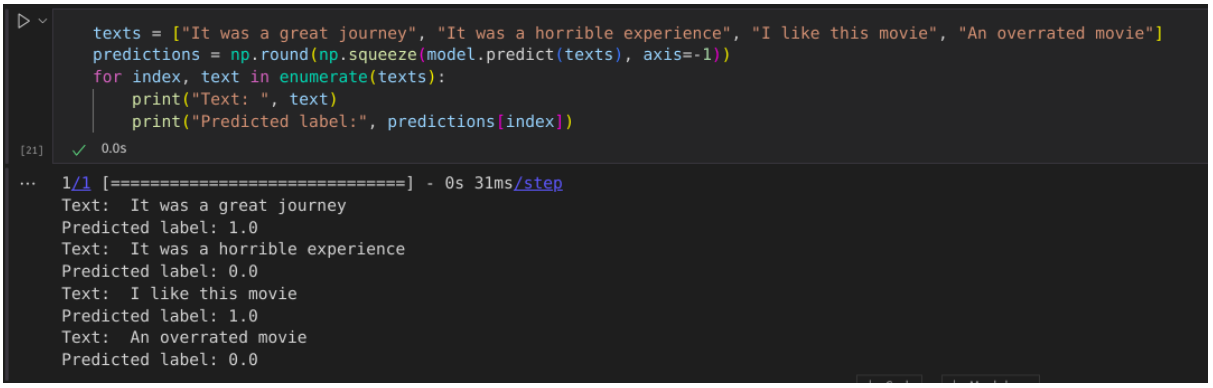
Перевіряємо точність на тестовому датасеті:

```
model.evaluate(test_ds)
✓ 2m 40.3s

75/75 [==============================] - 160s 2s/step - loss: 0.1564 - accuracy: 0.9386

[0.15640635788440704, 0.9386052489280701]
```

Декілька прикладів з текстового датасету:

```
for texts, label in test_ds.take(1):
    predictions = np.round(np.squeeze(model.predict(texts), axis=-1))
    label = label.numpy()
    for index, text in enumerate(texts):
        text = text.numpy()
        print("Text: ", text)
        print("True label:", label[index])
        print("Predicted label:", predictions[index])
        if(index > 2):
            break
[14] ✓ 1.7s Python

16/16 [==============================] - 2s 101ms/step
Text:  b'Was not impressed, and will not return.'
True label: 0
Predicted label: 0.0
Text:  b'I went in to purchase overalls and was treated so rudely I had to walk out even though he had the item I wanted.  I will never step foot in this placec of business again.'
True label: 0
Predicted label: 0.0
Text:  b'This place really is horrible... Every time I wind up getting convinced to go here, I always walk out feeling like my pocket has been picked. The food isn\'t \\""bad,\\"" but a
True label: 0
Predicted label: 0.0
Text:  b'First time visit.....  enjoyed their little cheese biscuits .... Had the ribs,beef brisket,mashed taters with gravy, Mac and cheese was goooood. Very cheesy and creamy just how
True label: 1
Predicted label: 1.0
```

Декілька власних прикладів:



Висновок: В результаті виконання лабораторної роботи було побудовано рекурентну нейрону мережу LSTM для визначення емоційного забарвлення тексту. Всього нейрона мережа має 751тис параметрів. Для Yelp датасету дана нейрона мережа показала гарні результати, а саме точність в 93,8 відсотків