

SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 8

28.11.2023

Temat: Liniowe RNN

Wariant 1

Jarosław Waliczek
Informatyka II stopień,
stacjonarne,
2 semestr,
Gr.2

1. Polecenie: Zadanie dotyczy zaimplementowania sieci rekurencyjnej RNN.

2. Wprowadzane dane:

Dane wejściowe składają się z 30 sekwencji po 20 kroków czasowych każda. Każda sekwencja wejściowa jest generowana z jednolitego rozkładu losowego, który jest zaokrąglany do 0, 0.5 lub 1. Cele wyjściowe `t` to liczba wystąpień „0.5” w sekwencji.

3. Wykorzystane komendy:

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense

# Funkcja do generowania danych wejściowych i wyjściowych
def generate_data(num_samples, seq_length):
    X = np.random.choice([0, 0.5, 1], size=(num_samples, seq_length))
    y = np.sum(X == 0.5, axis=1)
    return X, y

# Parametry
num_samples = 30
seq_length = 20
input_dim = 1 # Każdy krok czasowy zawiera jedną wartość (0, 0.5, lub 1)
output_dim = 1 # Jeden wynik na koniec sekwencji

# Generowanie danych
X_train, y_train = generate_data(num_samples, seq_length)

# Konfiguracja modelu RNN
model = Sequential()
model.add(SimpleRNN(units=10, input_shape=(seq_length, input_dim)))
model.add(Dense(units=output_dim, activation='linear'))

# Kompilacja modelu
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])

# Trenowanie modelu
model.fit(X_train, y_train, epochs=100, batch_size=1, verbose=2)

# Testowanie modelu na nowych danych
X_test, y_test = generate_data(5, seq_length)
predictions = model.predict(X_test)

# Wyświetlenie wyników
for i in range(len(X_test)):
    print("Input:", X_test[i].flatten())
    print("True Output:", y_test[i])
    print("Predicted Output:", predictions[i][0])
    print("\n")
```

4. Wynik działania:

rzuty ekranu:

```
WARNING:tensorflow:From c:\Python39\lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.nn.sparse_softmax_cross_entropy_with_logits instead.
WARNING:tensorflow:From c:\Python39\lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
WARNING:tensorflow:From c:\Python39\lib\site-packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

Epoch 1/100
WARNING:tensorflow:From c:\Python39\lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensor is deprecated. Please use tf.ragged.RaggedTensorType instead.
WARNING:tensorflow:From c:\Python39\lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.nn.conv2d is deprecated. Please use tf.nn.conv2d_v2 instead.

30/30 - 1s - loss: 30.7383 - accuracy: 0.0000e+00 - 997ms/epoch - 33ms/step
Epoch 2/100
30/30 - 0s - loss: 27.1312 - accuracy: 0.0000e+00 - 61ms/epoch - 2ms/step
Epoch 3/100
30/30 - 0s - loss: 22.3640 - accuracy: 0.0000e+00 - 59ms/epoch - 2ms/step
Epoch 4/100
30/30 - 0s - loss: 17.0916 - accuracy: 0.0000e+00 - 63ms/epoch - 2ms/step
Epoch 5/100
30/30 - 0s - loss: 12.7592 - accuracy: 0.0000e+00 - 60ms/epoch - 2ms/step
Epoch 6/100
30/30 - 0s - loss: 9.8766 - accuracy: 0.0000e+00 - 62ms/epoch - 2ms/step
Epoch 7/100
30/30 - 0s - loss: 7.9260 - accuracy: 0.0000e+00 - 63ms/epoch - 2ms/step
Epoch 8/100
...
True Output: 8
Predicted Output: 5.8964515

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

5. **Wnioski:** Model osiągnął dobre dopasowanie do danych treningowych, co sugeruje, że nauczył się pewnych zależności w danych treningowych

Repo: <https://github.com/Jaro233/MK>