

LSTM

December 29, 2023

Sprawozdanie

Matematyka Konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 11

19.12.2023

Algorytm LSTM dla tekstu Maksymilian Grygiel

Wariant 10

“Computer scientists and philosophers have since suggested that AI may become an existential risk to humanity if its rational capacities are not steered towards beneficial goals”

Link do repozytorium: <https://github.com/Maksiolo20/MK>

```
[4]: import numpy as np
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense

# Define the text
text = "Computer scientists and philosophers have since suggested that AI may_
      ↳become an existential risk to humanity if its rational capacities are not_
      ↳steered towards beneficial goals"

# Tokenize the text
tokenizer = Tokenizer()
tokenizer.fit_on_texts([text])
total_words = len(tokenizer.word_index) + 1

# Create input sequences and labels
input_sequences = []
for i in range(1, len(text.split())):
    n_gram_sequence = text.split()[:i+1]
    input_sequences.append(" ".join(n_gram_sequence))

# Pad sequences
max_sequence_len = max([len(seq.split()) for seq in input_sequences])
input_sequences = pad_sequences(tokenizer.texts_to_sequences(input_sequences),
                               maxlen=max_sequence_len, padding='pre')
```

```

# Create predictors and label
X, y = input_sequences[:, :-1], input_sequences[:, -1]
y = to_categorical(y, num_classes=total_words)

# Build the LSTM model
model = Sequential()
model.add(Embedding(total_words, 50, input_length=max_sequence_len-1))
model.add(LSTM(100))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',
    ↪metrics=['accuracy'])

# Train the model (you might need to adjust the number of epochs)
model.fit(X, y, epochs=100, verbose=1)

# Generate text using the trained model
def generate_text(seed_text, next_words, model, max_sequence_len):
    for _ in range(next_words):
        token_list = tokenizer.texts_to_sequences([seed_text])[0]
        token_list = pad_sequences([token_list], maxlen=max_sequence_len-1,
    ↪padding='pre')
        predicted = np.argmax(model.predict(token_list), axis=-1)
        output_word = ""
        for word, index in tokenizer.word_index.items():
            if index == predicted:
                output_word = word
                break
        seed_text += " " + output_word
    return seed_text

# Test the text generation
generated_text = generate_text("Computer", next_words=20, model=model,
    ↪max_sequence_len=max_sequence_len)
print(generated_text)

```

Epoch 1/100

1/1 [=====] - 2s 2s/step - loss: 3.2951 - accuracy:
0.0400

Epoch 2/100

1/1 [=====] - 0s 12ms/step - loss: 3.2905 - accuracy:
0.0400

Epoch 3/100

1/1 [=====] - 0s 13ms/step - loss: 3.2859 - accuracy:
0.1200

Epoch 4/100

1/1 [=====] - 0s 13ms/step - loss: 3.2810 - accuracy:

```

0.2400
Epoch 5/100
1/1 [=====] - 0s 12ms/step - loss: 3.2758 - accuracy:
0.2400
Epoch 6/100
1/1 [=====] - 0s 12ms/step - loss: 3.2702 - accuracy:
0.2400
Epoch 7/100
1/1 [=====] - 0s 12ms/step - loss: 3.2638 - accuracy:
0.2800
Epoch 8/100
1/1 [=====] - 0s 12ms/step - loss: 3.2565 - accuracy:
0.2400
Epoch 9/100
1/1 [=====] - 0s 11ms/step - loss: 3.2478 - accuracy:
0.2000
Epoch 10/100
1/1 [=====] - 0s 12ms/step - loss: 3.2373 - accuracy:
0.2000
Epoch 11/100
1/1 [=====] - 0s 11ms/step - loss: 3.2241 - accuracy:
0.2000
Epoch 12/100
1/1 [=====] - 0s 12ms/step - loss: 3.2073 - accuracy:
0.2000
Epoch 13/100
1/1 [=====] - 0s 11ms/step - loss: 3.1855 - accuracy:
0.1600
Epoch 14/100
1/1 [=====] - 0s 13ms/step - loss: 3.1575 - accuracy:
0.0800
Epoch 15/100
1/1 [=====] - 0s 12ms/step - loss: 3.1245 - accuracy:
0.0800
Epoch 16/100
1/1 [=====] - 0s 12ms/step - loss: 3.0943 - accuracy:
0.0800
Epoch 17/100
1/1 [=====] - 0s 12ms/step - loss: 3.0695 - accuracy:
0.0800
Epoch 18/100
1/1 [=====] - 0s 13ms/step - loss: 3.0291 - accuracy:
0.0800
Epoch 19/100
1/1 [=====] - 0s 12ms/step - loss: 2.9738 - accuracy:
0.0800
Epoch 20/100
1/1 [=====] - 0s 11ms/step - loss: 2.9228 - accuracy:

```

0.1600
Epoch 21/100
1/1 [=====] - 0s 12ms/step - loss: 2.8803 - accuracy:
0.1200
Epoch 22/100
1/1 [=====] - 0s 13ms/step - loss: 2.8252 - accuracy:
0.2000
Epoch 23/100
1/1 [=====] - 0s 12ms/step - loss: 2.7583 - accuracy:
0.2800
Epoch 24/100
1/1 [=====] - 0s 12ms/step - loss: 2.7264 - accuracy:
0.1200
Epoch 25/100
1/1 [=====] - 0s 12ms/step - loss: 2.6598 - accuracy:
0.1600
Epoch 26/100
1/1 [=====] - 0s 12ms/step - loss: 2.6331 - accuracy:
0.2000
Epoch 27/100
1/1 [=====] - 0s 12ms/step - loss: 2.5671 - accuracy:
0.1600
Epoch 28/100
1/1 [=====] - 0s 12ms/step - loss: 2.5667 - accuracy:
0.1600
Epoch 29/100
1/1 [=====] - 0s 12ms/step - loss: 2.4873 - accuracy:
0.2000
Epoch 30/100
1/1 [=====] - 0s 12ms/step - loss: 2.4775 - accuracy:
0.2000
Epoch 31/100
1/1 [=====] - 0s 12ms/step - loss: 2.4435 - accuracy:
0.2400
Epoch 32/100
1/1 [=====] - 0s 12ms/step - loss: 2.3789 - accuracy:
0.2400
Epoch 33/100
1/1 [=====] - 0s 13ms/step - loss: 2.3958 - accuracy:
0.2800
Epoch 34/100
1/1 [=====] - 0s 12ms/step - loss: 2.3262 - accuracy:
0.2400
Epoch 35/100
1/1 [=====] - 0s 12ms/step - loss: 2.3040 - accuracy:
0.2800
Epoch 36/100
1/1 [=====] - 0s 12ms/step - loss: 2.2854 - accuracy:

0.2800
Epoch 37/100
1/1 [=====] - 0s 12ms/step - loss: 2.2227 - accuracy: 0.2400
Epoch 38/100
1/1 [=====] - 0s 12ms/step - loss: 2.2245 - accuracy: 0.2800
Epoch 39/100
1/1 [=====] - 0s 12ms/step - loss: 2.1719 - accuracy: 0.3200
Epoch 40/100
1/1 [=====] - 0s 12ms/step - loss: 2.1518 - accuracy: 0.2800
Epoch 41/100
1/1 [=====] - 0s 12ms/step - loss: 2.1424 - accuracy: 0.2400
Epoch 42/100
1/1 [=====] - 0s 12ms/step - loss: 2.0846 - accuracy: 0.3600
Epoch 43/100
1/1 [=====] - 0s 13ms/step - loss: 2.0824 - accuracy: 0.3200
Epoch 44/100
1/1 [=====] - 0s 13ms/step - loss: 2.0786 - accuracy: 0.2400
Epoch 45/100
1/1 [=====] - 0s 13ms/step - loss: 2.0071 - accuracy: 0.4400
Epoch 46/100
1/1 [=====] - 0s 13ms/step - loss: 2.0183 - accuracy: 0.3200
Epoch 47/100
1/1 [=====] - 0s 12ms/step - loss: 2.0205 - accuracy: 0.2400
Epoch 48/100
1/1 [=====] - 0s 12ms/step - loss: 1.9366 - accuracy: 0.4400
Epoch 49/100
1/1 [=====] - 0s 12ms/step - loss: 1.9833 - accuracy: 0.3200
Epoch 50/100
1/1 [=====] - 0s 12ms/step - loss: 1.9489 - accuracy: 0.2800
Epoch 51/100
1/1 [=====] - 0s 13ms/step - loss: 1.8914 - accuracy: 0.4400
Epoch 52/100
1/1 [=====] - 0s 13ms/step - loss: 1.9366 - accuracy:

```

0.4000
Epoch 53/100
1/1 [=====] - 0s 13ms/step - loss: 1.8372 - accuracy:
0.5600
Epoch 54/100
1/1 [=====] - 0s 12ms/step - loss: 1.8684 - accuracy:
0.3200
Epoch 55/100
1/1 [=====] - 0s 12ms/step - loss: 1.8015 - accuracy:
0.6800
Epoch 56/100
1/1 [=====] - 0s 12ms/step - loss: 1.8188 - accuracy:
0.5200
Epoch 57/100
1/1 [=====] - 0s 12ms/step - loss: 1.7733 - accuracy:
0.6400
Epoch 58/100
1/1 [=====] - 0s 12ms/step - loss: 1.7686 - accuracy:
0.6000
Epoch 59/100
1/1 [=====] - 0s 12ms/step - loss: 1.7384 - accuracy:
0.6400
Epoch 60/100
1/1 [=====] - 0s 12ms/step - loss: 1.7220 - accuracy:
0.6400
Epoch 61/100
1/1 [=====] - 0s 11ms/step - loss: 1.7075 - accuracy:
0.6400
Epoch 62/100
1/1 [=====] - 0s 12ms/step - loss: 1.6801 - accuracy:
0.6400
Epoch 63/100
1/1 [=====] - 0s 12ms/step - loss: 1.6758 - accuracy:
0.6400
Epoch 64/100
1/1 [=====] - 0s 12ms/step - loss: 1.6390 - accuracy:
0.7200
Epoch 65/100
1/1 [=====] - 0s 11ms/step - loss: 1.6405 - accuracy:
0.7200
Epoch 66/100
1/1 [=====] - 0s 12ms/step - loss: 1.6046 - accuracy:
0.7600
Epoch 67/100
1/1 [=====] - 0s 12ms/step - loss: 1.6029 - accuracy:
0.7200
Epoch 68/100
1/1 [=====] - 0s 11ms/step - loss: 1.5781 - accuracy:

```

0.7600
Epoch 69/100
1/1 [=====] - 0s 11ms/step - loss: 1.5590 - accuracy: 0.7600
Epoch 70/100
1/1 [=====] - 0s 11ms/step - loss: 1.5524 - accuracy: 0.7200
Epoch 71/100
1/1 [=====] - 0s 11ms/step - loss: 1.5219 - accuracy: 0.7200
Epoch 72/100
1/1 [=====] - 0s 12ms/step - loss: 1.5129 - accuracy: 0.6800
Epoch 73/100
1/1 [=====] - 0s 12ms/step - loss: 1.4980 - accuracy: 0.7600
Epoch 74/100
1/1 [=====] - 0s 12ms/step - loss: 1.4731 - accuracy: 0.7200
Epoch 75/100
1/1 [=====] - 0s 12ms/step - loss: 1.4638 - accuracy: 0.6800
Epoch 76/100
1/1 [=====] - 0s 11ms/step - loss: 1.4500 - accuracy: 0.7600
Epoch 77/100
1/1 [=====] - 0s 11ms/step - loss: 1.4284 - accuracy: 0.7600
Epoch 78/100
1/1 [=====] - 0s 11ms/step - loss: 1.4144 - accuracy: 0.8000
Epoch 79/100
1/1 [=====] - 0s 13ms/step - loss: 1.4052 - accuracy: 0.8000
Epoch 80/100
1/1 [=====] - 0s 11ms/step - loss: 1.3912 - accuracy: 0.7600
Epoch 81/100
1/1 [=====] - 0s 11ms/step - loss: 1.3715 - accuracy: 0.8000
Epoch 82/100
1/1 [=====] - 0s 11ms/step - loss: 1.3557 - accuracy: 0.8000
Epoch 83/100
1/1 [=====] - 0s 11ms/step - loss: 1.3450 - accuracy: 0.8000
Epoch 84/100
1/1 [=====] - 0s 11ms/step - loss: 1.3362 - accuracy:

```

0.8000
Epoch 85/100
1/1 [=====] - 0s 11ms/step - loss: 1.3310 - accuracy:
0.8000
Epoch 86/100
1/1 [=====] - 0s 12ms/step - loss: 1.3229 - accuracy:
0.8000
Epoch 87/100
1/1 [=====] - 0s 12ms/step - loss: 1.3244 - accuracy:
0.7200
Epoch 88/100
1/1 [=====] - 0s 12ms/step - loss: 1.3055 - accuracy:
0.7600
Epoch 89/100
1/1 [=====] - 0s 12ms/step - loss: 1.2907 - accuracy:
0.8000
Epoch 90/100
1/1 [=====] - 0s 12ms/step - loss: 1.2591 - accuracy:
0.8400
Epoch 91/100
1/1 [=====] - 0s 12ms/step - loss: 1.2401 - accuracy:
0.8400
Epoch 92/100
1/1 [=====] - 0s 11ms/step - loss: 1.2346 - accuracy:
0.8000
Epoch 93/100
1/1 [=====] - 0s 12ms/step - loss: 1.2322 - accuracy:
0.8400
Epoch 94/100
1/1 [=====] - 0s 12ms/step - loss: 1.2285 - accuracy:
0.8000
Epoch 95/100
1/1 [=====] - 0s 11ms/step - loss: 1.2051 - accuracy:
0.8400
Epoch 96/100
1/1 [=====] - 0s 12ms/step - loss: 1.1848 - accuracy:
0.8400
Epoch 97/100
1/1 [=====] - 0s 12ms/step - loss: 1.1737 - accuracy:
0.8400
Epoch 98/100
1/1 [=====] - 0s 11ms/step - loss: 1.1699 - accuracy:
0.8800
Epoch 99/100
1/1 [=====] - 0s 11ms/step - loss: 1.1671 - accuracy:
0.8000
Epoch 100/100
1/1 [=====] - 0s 11ms/step - loss: 1.1514 - accuracy:

```


0.8800

```
1/1 [=====] - 0s 349ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 16ms/step
```

Computer scientists scientists and philosophers have suggested that ai may become an existential risk to humanity if its rational capacities are

[]: