

Практика: Генерация текста с помощью нейросети

Grigory Sapunov  CTO / Intento



Практика #1: Давайте соберём простую RNN для генерации символов

Генерация текста с помощью RNN

Ресурсы

- Пример:
(keras 1)
https://github.com/fchollet/keras/blob/keras-1/examples/lstm_text_generation.py
(keras 2)
https://github.com/fchollet/keras/blob/master/examples/lstm_text_generation.py
- Разбор похожего кейса
http://ml4a.github.io/guides/recurrent_neural_networks/
- Разбор другого похожего кейса
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

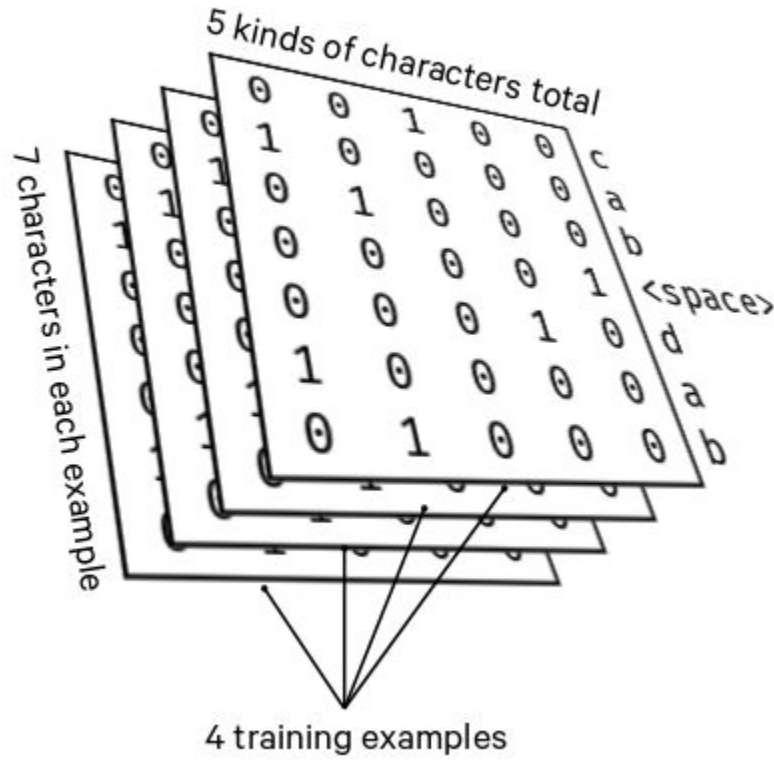
Подготовка данных

```
maxlen = 40
step = 3
sentences = []
next_chars = []
for i in range(0, len(text) - maxlen, step):
    sentences.append(text[i: i + maxlen])
    next_chars.append(text[i + maxlen])
print('nb sequences:', len(sentences))
```

Подготовка данных

```
print('Vectorization...')  
X = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)  
y = np.zeros((len(sentences), len(chars)), dtype=np.bool)  
for i, sentence in enumerate(sentences):  
    for t, char in enumerate(sentence):  
        X[i, t, char_indices[char]] = 1  
    y[i, char_indices[next_chars[i]]] = 1
```

Пример для алфавита из 5 символов



Создание модели

```
model = Sequential()  
model.add(LSTM(128, input_shape=(maxlen, len(chars))))  
model.add(Dense(len(chars)))  
model.add(Activation('softmax'))  
  
optimizer = RMSprop(lr=0.01)  
model.compile(loss='categorical_crossentropy',  
optimizer=optimizer)
```


Обучение модели

```
for iteration in range(1, 60):  
    print()  
    print('-' * 50)  
    print('Iteration', iteration)  
    model.fit(X, y, batch_size=128, nb_epoch=1)
```

Генерация текста

```
generated = ''  
sentence = text[start_index: start_index + maxlen]  
generated += sentence  
print('----- Generating with seed: "' + sentence + '"')  
sys.stdout.write(generated)
```

Генерация текста

```
for i in range(400):  
    x = np.zeros((1, maxlen, len(chars)))  
    for t, char in enumerate(sentence):  
        x[0, t, char_indices[char]] = 1.  
  
    preds = model.predict(x, verbose=0)[0]  
    next_index = sample(preds, diversity)  
    next_char = indices_char[next_index]  
  
    generated += next_char  
    sentence = sentence[1:] + next_char
```

Усложнение модели

Усложнение модели: больше слоёв

```
model = Sequential()  
model.add(LSTM(128, input_shape=(maxlen, len(chars)),  
              return_sequences=True))  
model.add(LSTM(128))  
model.add(Dense(len(chars)))  
model.add(Activation('softmax'))  
  
optimizer = RMSprop(lr=0.01)  
model.compile(loss='categorical_crossentropy',  
              optimizer=optimizer)
```

Усложнение модели: двунаправленность

```
from keras.layers import Bidirectional
```

```
model = Sequential()
```

```
model.add(Bidirectional(LSTM(128),  
    input_shape=(maxlen, len(chars))))
```

```
model.add(Dense(len(chars)))
```

```
model.add(Activation('softmax'))
```

```
optimizer = RMSprop(lr=0.01)
```

```
model.compile(loss='categorical_crossentropy',  
optimizer=optimizer)
```