

Попробуйте обучить нейронную сеть LSTM на любом другом датасете (любимый временной ряд, текст на русском (другом языке) как генератор или классификатор, или прилагаемый набор `airline-passengers` - пассажиропоток для авиалиний).

*Опишите, какой результат вы получили? Что помогло вам улучшить ее точность?

*Попробуйте на numpy реализовать нейронную сеть архитектуры LSTM

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers
import tensorflow.keras.utils as ku
import numpy as np
```

```
from tensorflow.keras.callbacks import EarlyStopping
```

```
# Создаем объект для токенизации
tokenizer = Tokenizer(num_words=10000)
!wget --no-check-certificate \
    https://storage.googleapis.com/laurencemoroney-blog.appspot.com/sonnets.txt \
    -O /tmp/sonnets.txt
data = open('/content/drive/MyDrive/learning/neural network/2/ПТЭ.txt').read()
#Читаем текст
data_text = data.lower().split(" ")
```

```
--2021-12-21 00:10:05-- https://storage.googleapis.com/laurencemoroney-blog.appspot
Resolving storage.googleapis.com (storage.googleapis.com)... 173.194.193.128, 173.194.
Connecting to storage.googleapis.com (storage.googleapis.com)|173.194.193.128|:443..
HTTP request sent, awaiting response... 200 OK
Length: 93578 (91K) [text/plain]
Saving to: '/tmp/sonnets.txt'
```

```
/tmp/sonnets.txt 100%[=====>] 91.38K --.-KB/s in 0.001s
```

```
2021-12-21 00:10:05 (82.1 MB/s) - '/tmp/sonnets.txt' saved [93578/93578]
```

убираю пунктуацию

```
import string

for p in string.punctuation:
    if p in data:
        data = data.replace(p, '')
```

data

'Требования к рабочим местам работников перевозчика используемым работниками перевозчика техническим средствам и порядок их содержания определяются перевозчиком с учетом требований трудового законодательства Российской Федерации положений настоящих ПТЭ

\nПеревозчик обязан обеспечить эксплуатацию работниками метрополитена объектов инфраструктуры метрополитена в соответствии с правилами и инструкциями по охране труда по жарной безопасности установленными для выполняемой работы и возможность немедленного извещения уполномоченного работника перевозчика о любой ситуации угрожающей жизни и здоровью людей или безопасности движения\nПраво управления подвижным составом сигналами аппаратами механизмами другими устройствами связанными с обеспечением безопаснос

```
corpus = data.lower().split("\n")
```

corpus

'при одновременной неисправности на одной станции нескольких попутных входных и

'предельно допускаемая скорость движения хозяйственных поездов устанавливается п

'перевозчик не вправе устанавливать скорость движения хозяйственных поездов боле

'25 кмчпри движении в неправильном направлениипри транспортировке груженых рельс

'20 кмчпри проследовании светофора с запрещающим показаниемпри маневровых передв

'15 кмчпри движении на участках пути с уклоном более 40% хозяйственного поезда о

'10 кмчпри следовании с негабаритным грузомпри въезде на занятый путь станциипри

'5 кмчпри транспортировке подвижного состава на вспомогательных тележках по крив

'допускается отправление хозяйственного поезда следующего к месту работ требующи

'при необходимости приёма двух хозяйственных поездов с различных направлений на

'перед приёмом этих поездов дежурный по станции обязан оградить путь приёма с об

'приём второго поезда разрешается после остановки ранее принятого поезда и устно

'расцепка хозяйственного поезда допускается по указанию руководителя работ после

'расцепка рельсовозных тележек расположенных между тяговыми подвижными единицами

'передвижение прицепных единиц вручную допускается по указанию руководителя рабо

'передвижение хозяйственного поезда на закрытом пути производится по указанию ру

'отправление хозяйственных поездов с перегона производится по указанию руководит

'при производстве работ не требующих закрытия пути на уклоне более 40 % пропуск :

'движение хозяйственных поездов в неправильном направлении производится по прика

'право на следование хозяйственных поездов в неправильном направлении подтвержда

'приказ адресуется станцииограничивающим путь следования в неправильном направл

'до передачи приказа поездной диспетчер должен дать указание дежурному по станци

'если поезд будет следовать на перегон или до начала пассажирской платформы – си

'если поезд будет следовать к пассажирской платформе станции назначения – сигнал

'установка переносного сигнала остановки производится до передачи приказа поездн

'после передачи приказа на движение поезда в неправильном направлении перевод ст

'копия приказа вручается машинисту хозяйственного поезда под подпись на станции

'если движение хозяйственного поезда в неправильном направлении начинается с пер

'в случае отправления нескольких хозяйственных поездов в неправильном направлени

'движение хозяйственных поездов на пути следования которых в неправильном направ

'до отправления подвижного состава в неправильном направлении перевозчик письмен

'порядок отправления в неправильном направлении нескольких хозяйственных поездов

'отправление нескольких хозяйственных поездов в неправильном направлении на пере

'уклоном более 40 % запрещается данное требование не распространяется на поезда

'при необходимости отправление нескольких хозяйственных поездов на перегон с укл

'каждый следующий хозяйственный поезд отправляется со станции на перегон по сигн

'возвращение с перегона имеющего уклон более 40 % вслед нескольких хозяйственных

'если при следовании поезда в неправильном направлении имеются светофоры полуавт

'при запрещающем показании светофоров ограждения проследование их осуществляется

'при запрещающем показании дополнительного сигнала опасности «доп» проследование

'если на дополнительном сигнале опасности «доп» запрещающее показание стрелка ус

'формы приказов передаваемых поездным диспетчером при назначении вспомогательных

'маневровые передвижения на парковых депоовских и прочих путях разрешается произв

'маневровые передвижения в дневное время при отсутствии перед тяговой подвижной
 'скорость маневровых передвижений на парковых депокских и прочих путях устанавли
 'скорость движения должна быть не более 15 кмчпри производстве маневровых передв
 'скорость движения должна быть не более 10 кмчпри производстве маневровых передв
 'скорость следования должна быть не более 5 кмчпри транспортировке подвижного со
 'машиниста на прицепной единице прицепленной впереди мотовоза и сопровождения им
 'скорость движения должна быть не более 3 кмчпри подходе к подвижному составу ту
 'порядок закрепления подвижного состава на парковых путях а также на путях элект
 'погрузка размещение крепление перевозка и выгрузка хозяйственных грузов в том ч
 'порядок погрузки выгрузки и перевозки рельсовых плетей и отдельных рельсов на р
 'подача и маневры подвижного состава железных дорог на путях метрополитена',
 'передвижение подвижного состава имеющего габариты подвижного состава железных д
 'при необходимости подачи подвижного состава имеющего габариты подвижного состав
 'порядок организации следования к месту производства работ восстановительного по,

Исполним токенизацию и кодирование

```
tokenizer.fit_on_texts( corpus)
total_words = len(tokenizer.word_index) + 1

# создание входных последовательностей с использованием списка токенов
input_sequences = []
for line in corpus:
    token_list = tokenizer.texts_to_sequences([line])[0]
    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        input_sequences.append(n_gram_sequence)

# последовательности прокладок
max_sequence_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequence_len, padding

# создание предикторов и меток
predictors, label = input_sequences[:, :-1], input_sequences[:, -1]

label = ku.to_categorical(label, num_classes=total_words)

token_list

[111, 179, 74, 32, 535, 174, 43, 3834, 22, 13, 4, 81, 123, 400, 28]

tokenizer.index_word[10]

'пути'

tokenizer.word_index['светофор']

191

input_sequences[11:15,:]
```

```
array([[ 0,    0,    0, ..., 1359,    2,  111],
       [ 0,    0,    0, ...,    2,  111,   86],
       [ 0,    0,    0, ...,  111,   86, 1026],
       [ 0,    0,    0, ...,   86, 1026,  673]], dtype=int32)
```

```
label.shape
```

```
(21911, 3835)
```

```
callback = EarlyStopping(monitor='val_loss', patience=30) # создаем обратный вызов - calba
```

```
total_words
```

```
3835
```

```
from tensorflow.python.keras.regularizers import L2
model = Sequential()
model.add(Embedding(total_words, 256, input_length=max_sequence_len-1)) # вложение тренир
# на вход вложения (эмбединга) передаем вектор словарь (длина равна словарю, вектор разреж
# model.add(Bidirectional(LSTM(50, return_sequences = True)))

model.add(LSTM(512, return_sequences=False))
# model.add(LSTM(256))
# model.add(LSTM(128, return_sequences=True))
# model.add(LSTM(64))
model.add(Dropout( 0.2))
model.add(Dense(total_words, activation='tanh', kernel_regularizer= regularizers.L2(0.05))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====		
embedding (Embedding)	(None, 378, 256)	981760
lstm (LSTM)	(None, 512)	1574912
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 3835)	1967355
dense_1 (Dense)	(None, 3835)	14711060

```
=====
Total params: 19,235,087
Trainable params: 19,235,087
Non-trainable params: 0
```

```
None
```

```
history = model.fit(predictors, label, epochs=100, validation_split=0.33, callbacks=[callb
# history = model.fit(predictors, label, epochs=50, verbose=1)
```

```
" history: model:1.12xprecursors, 1000, epochs: 30, verbose: 1,
```

```
Epoch 3/100
459/459 [=====] - 91s 199ms/step - loss: 6.3628 - accura
Epoch 4/100
459/459 [=====] - 90s 197ms/step - loss: 6.0506 - accura
Epoch 5/100
459/459 [=====] - 90s 196ms/step - loss: 5.7234 - accura
Epoch 6/100
459/459 [=====] - 90s 196ms/step - loss: 5.3831 - accura
Epoch 7/100
459/459 [=====] - 90s 197ms/step - loss: 5.0576 - accura
Epoch 8/100
459/459 [=====] - 90s 197ms/step - loss: 4.7527 - accura
Epoch 9/100
459/459 [=====] - 91s 198ms/step - loss: 4.4607 - accura
Epoch 10/100
459/459 [=====] - 91s 198ms/step - loss: 4.1936 - accura
Epoch 11/100
459/459 [=====] - 91s 198ms/step - loss: 3.9336 - accura
Epoch 12/100
459/459 [=====] - 91s 197ms/step - loss: 3.6806 - accura
Epoch 13/100
459/459 [=====] - 91s 198ms/step - loss: 3.4806 - accura
Epoch 14/100
459/459 [=====] - 90s 197ms/step - loss: 3.2800 - accura
Epoch 15/100
459/459 [=====] - 91s 198ms/step - loss: 3.1016 - accura
Epoch 16/100
459/459 [=====] - 91s 198ms/step - loss: 2.9121 - accura
Epoch 17/100
459/459 [=====] - 91s 198ms/step - loss: 2.7677 - accura
Epoch 18/100
459/459 [=====] - 91s 198ms/step - loss: 2.6349 - accura
Epoch 19/100
459/459 [=====] - 91s 197ms/step - loss: 2.5101 - accura
Epoch 20/100
459/459 [=====] - 91s 198ms/step - loss: 2.3890 - accura
Epoch 21/100
459/459 [=====] - 91s 199ms/step - loss: 2.2908 - accura
Epoch 22/100
459/459 [=====] - 91s 199ms/step - loss: 2.1881 - accura
Epoch 23/100
459/459 [=====] - 91s 197ms/step - loss: 2.1385 - accura
Epoch 24/100
459/459 [=====] - 91s 199ms/step - loss: 2.1026 - accura
Epoch 25/100
459/459 [=====] - 91s 199ms/step - loss: 2.0564 - accura
Epoch 26/100
459/459 [=====] - 91s 199ms/step - loss: 2.0233 - accura
Epoch 27/100
459/459 [=====] - 91s 199ms/step - loss: 1.9360 - accura
Epoch 28/100
459/459 [=====] - 91s 198ms/step - loss: 1.8682 - accura
Epoch 29/100
459/459 [=====] - 90s 197ms/step - loss: 1.8307 - accura
Epoch 30/100
459/459 [=====] - 90s 197ms/step - loss: 1.8416 - accura
Epoch 31/100
459/459 [=====] - 90s 197ms/step - loss: 1.8829 - accura
```

```

import matplotlib.pyplot as plt
acc = history.history['accuracy']
acc_v = history.history['val_accuracy']
loss = history.history['loss']
loss_v = history.history['val_loss']

epochs = range(len(acc))

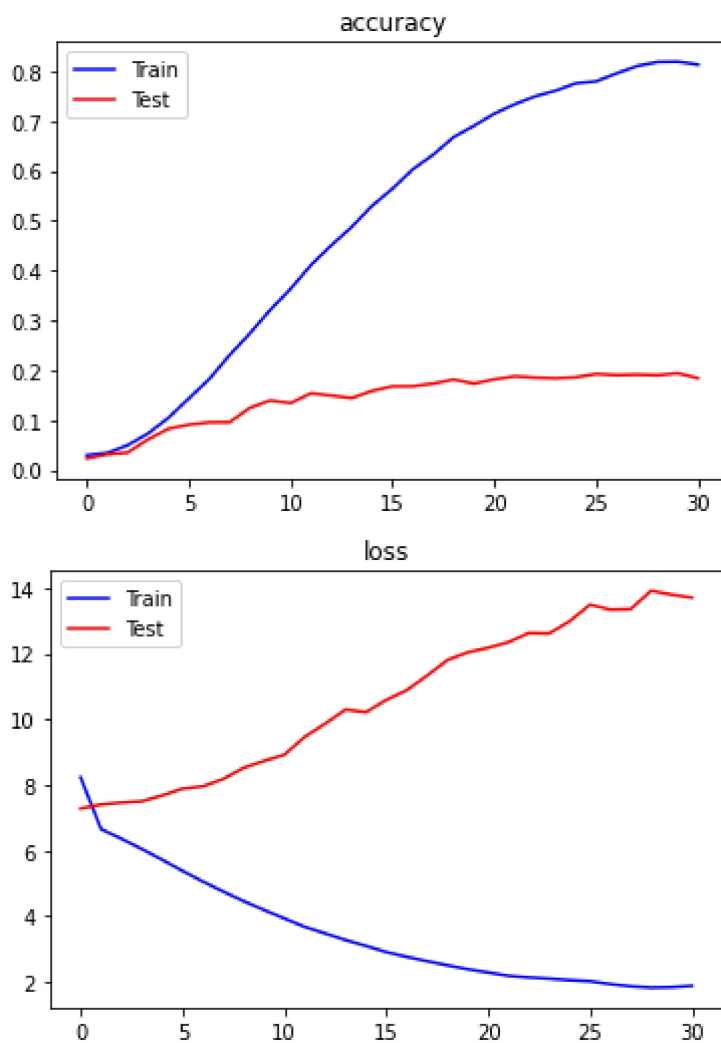
plt.plot(epochs, acc, 'b', label='Train')
plt.plot(epochs, acc_v, 'r', label='Test')
plt.legend()
plt.title('accuracy')

plt.figure()

plt.plot(epochs, loss, 'b', label='Train')
plt.plot(epochs, loss_v, 'r', label='Test')
plt.title('loss')
plt.legend()

plt.show()

```



```

seed_text = "машинист несёт ответственность"
seed_text = seed_text.lower()

```

```
#seed_text = "She is my girl, I see in"
next_words = 50
```

```
for _ in range(next_words):
    token_list = tokenizer.texts_to_sequences([seed_text])[0]
    # print(token_list)
    token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
    #predicted = model.predict_classes(token_list, verbose=0)
    output_word = ""
    y_p=model.predict(token_list)
    y_predict = y_p * np.random.random( size = (1,y_p.shape[1]))
    predicted_ind = np.argmax(y_predict,axis = 1)
    for word, index in tokenizer.word_index.items():
        if index == predicted_ind:
            output_word = word
            break
    seed_text += " " + output_word
print(seed_text)
```

ачения специального назначения с отклоненного устройства соблюдение а с установить ус

Посмотрим на предсказание через вероятные ответы (не одно слово имеет большую вероятность)

```
y_p=model.predict(token_list)
```

```
plt.figure(figsize=(16,4))
plt.plot(y_p[0,:])
plt.xlabel('номер слова')
plt.ylabel('вероятность слова')
plt.title('вероятности по словам')
plt.grid('on')
plt.show()
```



```
ind_prob=np.where(y_p[0,:]>0.0410)[0]
```

```
ind_prob
```

```
array([ 184,  916, 1369, 1466])
```

```
for i,ind in enumerate(ind_prob.tolist()):  
    print(tokenizer.index_word[ind], ' : ', y_p[0,ind])
```

```
«0» : 0.096599564  
блокировкой : 0.052315116  
доступ : 0.048804503  
замыкания : 0.22407739
```

▼ Вывод

модель стала лучше предсказывать следующее слово после обработки текста, удаления ненужных символов, знаков препинания, увеличение кол-ва слоев результата не дало, только обучаться сильно дольше стала модель, точность стала выше после увеличения эпох и кол-во нейронов

✓ 0 сек. выполнено в 03:58

