Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Лабораторная работа №8
Дисциплина «Разработка интеллектуальных систем»

Выполнил:
Съестов Дмитрий Вячеславович
Группа P3417

Преподаватель:
Жукова Наталия Александровна

Санкт-Петербург
2020

Программа обучения

```
import numpy
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils

filename = "text.txt"
raw_text = open(filename).read()
raw_text = raw_text.lower()

chars = sorted(list(set(raw_text)))
char_to_int = dict((c, i) for i, c in enumerate(chars))

n_chars = len(raw_text)
n_vocab = len(chars)
print("Total Characters: ", n_chars)
print("Total Vocab: ", n_vocab)

seq_length = 100
seq_length = 100
dataX = []
dataY = []
for i in range(0, n_chars - seq_length, 1):
    seq_in = raw_text[i:i + seq_length]
    seq_out = raw_text[i + seq_length]
    dataX.append([char_to_int[char] for char in seq_in])
    dataY.append(char_to_int[seq_out])
n_patterns = len(dataX)
print("Total Patterns: ", n_patterns)

# reshape X to be [samples, time steps, features]
X = numpy.reshape(dataX, (n_patterns, seq_length, 1))
# normalize
X = X / float(n_vocab)
# one hot encode the output variable
y = np_utils.to_categorical(dataY)

model = Sequential()
model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(y.shape[1], activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')

# define the checkpoint
filepath="weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')
callbacks_list = [checkpoint]

model.fit(X, y, epochs=20, batch_size=128, callbacks=callbacks_list)
```

Программа генерации

```
# Load LSTM network and generate text
import sys
import numpy
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils

# load ascii text and covert to lowercase
filename = "text.txt"
raw_text = open(filename).read()
raw_text = raw_text.lower()

# create mapping of unique chars to integers, and a reverse mapping
chars = sorted(list(set(raw_text)))
char_to_int = dict((c, i) for i, c in enumerate(chars))
int_to_char = dict((i, c) for i, c in enumerate(chars))

# summarize the loaded data
n_chars = len(raw_text)
n_vocab = len(chars)
print("Total Characters: ", n_chars)
print("Total Vocab: ", n_vocab)

# prepare the dataset of input to output pairs encoded as integers
seq_length = 100
dataX = []
dataY = []
for i in range(0, n_chars - seq_length, 1):
    seq_in = raw_text[i:i + seq_length]
    seq_out = raw_text[i + seq_length]
    dataX.append([char_to_int[char] for char in seq_in])
    dataY.append(char_to_int[seq_out])

n_patterns = len(dataX)
print("Total Patterns: ", n_patterns)

# reshape X to be [samples, time steps, features]
X = numpy.reshape(dataX, (n_patterns, seq_length, 1))

# normalize
X = X / float(n_vocab)

# one hot encode the output variable
y = np_utils.to_categorical(dataY)

# define the LSTM model
model = Sequential()
model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(y.shape[1], activation='softmax'))

# load the network weights
filename = "weights-improvement-20-2.4303.hdf5"
```

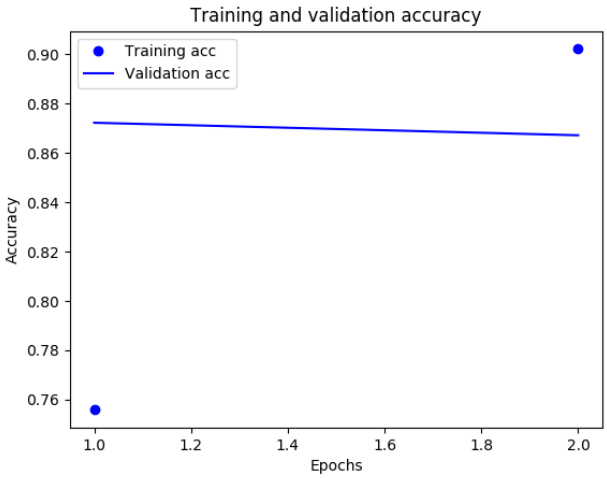
```
model.load_weights(filename)
model.compile(loss='categorical_crossentropy', optimizer='adam')

# pick a random seed
start = numpy.random.randint(0, len(dataX)-1)
pattern = dataX[start]
print("Seed:")
print("\"", ''.join([int_to_char[value] for value in pattern]), "\"")

# generate characters
for i in range(1000):
    x = numpy.reshape(pattern, (1, len(pattern), 1))
    x = x / float(n_vocab)
    prediction = model.predict(x, verbose=0)
    index = numpy.argmax(prediction)
    result = int_to_char[index]
    seq_in = [int_to_char[value] for value in pattern]
    sys.stdout.write(result)
    pattern.append(index)
    pattern = pattern[1:len(pattern)]

print("\nDone.")
```

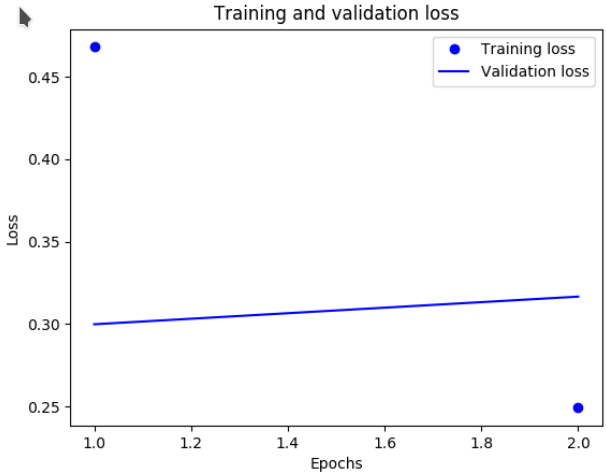
Результат генерации

```
Total Characters:  101298
Total Vocab:  56
Total Patterns:  101198
Seed:
" t be heirs
of the late wilbur whateley. they found the countryside in great
agitation, both because  "
the sireo of the farle and the fore that the siate of the soate of the saat
the soade oin sas and the sare and the that le the fart  a denl tf the forld
oe the thateleys and the siae and the the sare of the fart  a sentin' an'
shen ti the woane oo the roate of the toane to the riate of the toate oe the
soate oe the soate of the saat the sare and the that and the for oar so the
siate of the saat the sare and the that and the fart  a denl tf the forld oe
the thateleys and the siae and tee the sare of the fart  a sentin' an' shen
ti the woane oo the rare of tee the saat at' the shateley's alo th the toate
oa tenten sere th the siae and the that ae ael the tiate on the sare of the
soate of the saat th the tiate on the sare and the that ae tent th the soate
of the saat and the thate oe the saate on the soate of the saat the sare and
the that and the for oar and the siaee oo the soate of the saat the sare and
the that and the forl  a dunll tone to the siate of the toate oe the saat and
the t
Done.
```

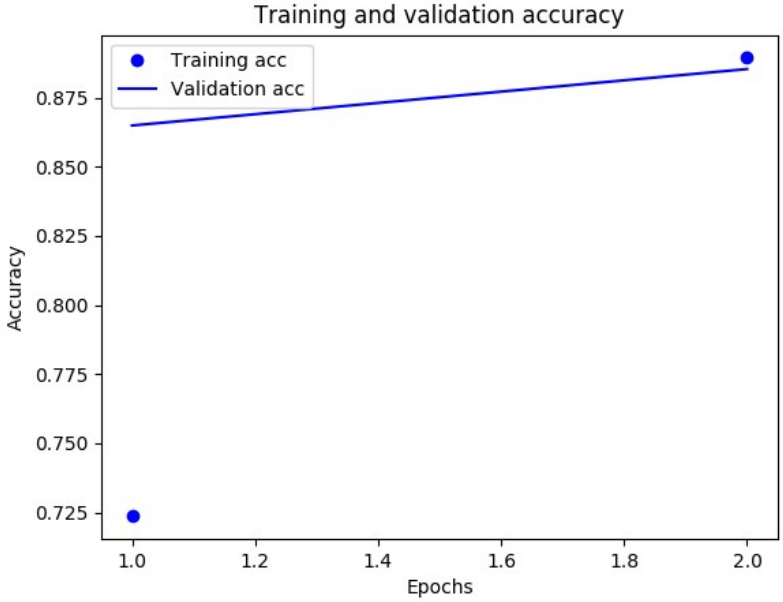
Результаты

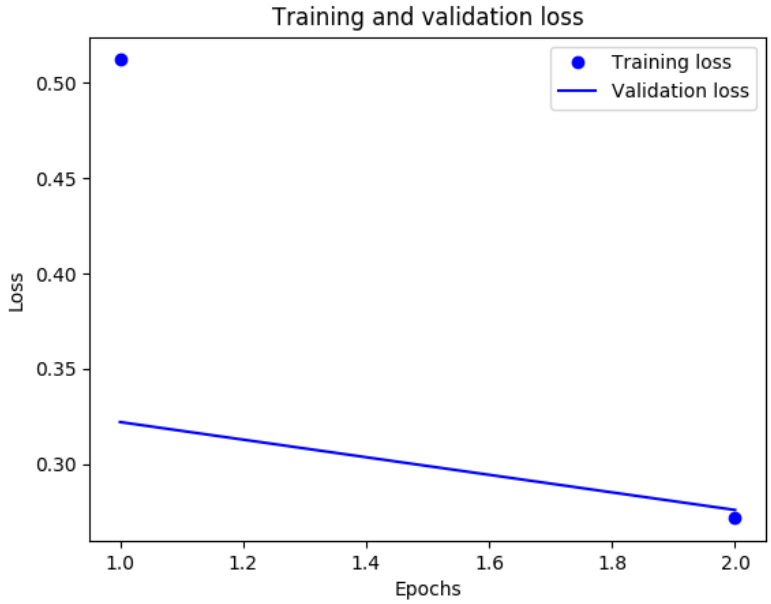
Без слоёв Dropout:





Со слоями Dropout:





Вывод

Как видно по графикам, без новых слоёв происходило переобучение, о чём свидетельствует снижение точности и повышение потерь по мере обучения. Добавление двух дополнительных слоёв Dropout решило эту проблему.