

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

**ОТЧЕТ**

**Лабораторная работа №2**

по курсу «Проектирование интеллектуальных систем»

**Тема « Логистическая регрессия и полносвязная сеть»**

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-22М  
Егоров С. А.

\_\_\_\_\_  
" \_\_ " \_\_\_\_\_ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Терехов В. И., к.т.н.,  
Доцент

\_\_\_\_\_  
" \_\_ " \_\_\_\_\_ 2020 г.

Москва - 2020

---

## Задание

1. Создать логистическую регрессию для классификации набора данных MNIST. Функция логистической регрессии выглядит следующим образом:

$$f(x) = \frac{1}{1 + e^{w*x+b}} \quad (5)$$

2. Создать нейронную сеть с 5 полносвязными слоями для классификации набора данных MNIST с количеством нейронов в слоях от первого до пятого - (200,100,60,30,10)

## Реализация

- 1) Подключение библиотек и источников данных

```
: import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
DATA_DIR = '/data'
NUM_STEPS = 1000
MINIBATCH_SIZE = 100
LEARNING_RATE = 0.5
data = input_data.read_data_sets (DATA_DIR , one_hot = True )
```

```
Extracting /data/train-images-idx3-ubyte.gz
Extracting /data/train-labels-idx1-ubyte.gz
Extracting /data/t10k-images-idx3-ubyte.gz
Extracting /data/t10k-labels-idx1-ubyte.gz
```

- 2) Задание 1

```
x = tf.placeholder (tf.float32 , [ None , 784])
W = tf.Variable (tf.zeros ([784 , 10]))
b = tf.Variable (tf.zeros ([10]) , dtype =tf.float32 )
y_true = tf.placeholder (tf.float32 , [ None , 10])

y_pred = 1/(tf.exp(tf.matmul(x,W) + b)+1)

cross_entropy = tf.reduce_mean (tf.nn.softmax_cross_entropy_with_logits (logits = y_pred , labels = y_true ))
gd_step = tf.train.GradientDescentOptimizer(LEARNING_RATE).minimize(cross_entropy)
correct_mask = tf.equal(tf.argmax(y_pred, 1), tf.argmax(y_true , 1))
accuracy = tf.reduce_mean (tf. cast ( correct_mask , tf.float32 ))
```

- 3) Задание 2

```
x = tf.placeholder (tf.float32 , [ None , 784])

l1 = tf.layers.dense (x , L1 , activation =tf.nn.relu , use_bias = True )
l2 = tf.layers.dense (l1, L2 , activation =tf.nn.relu , use_bias = True )
l3 = tf.layers.dense (l2, L3 , activation =tf.nn.relu , use_bias = True )
l4 = tf.layers.dense (l3, L4 , activation =tf.nn.relu , use_bias = True )
y_pred = tf.layers.dense (l4, L5 , activation =tf.nn.relu , use_bias = True )

y_true = tf.placeholder (tf.float32 , [ None , 10])
cross_entropy = tf.reduce_mean (tf.nn.softmax_cross_entropy_with_logits (logits = y_pred , labels = y_true ))
gd_step = tf.train.GradientDescentOptimizer(LEARNING_RATE).minimize(cross_entropy)
correct_mask = tf.equal(tf.argmax(y_pred,1), tf.argmax(y_true,1))
accuracy = tf.reduce_mean (tf.cast(correct_mask, tf.float32))
```

## Результаты

### 1) Задание 1

```
with tf.Session () as sess :
    # Train
    sess.run(tf.global_variables_initializer ())
    for i in range ( NUM_STEPS ):
        batch_x , batch_y = data.train.next_batch(MINIBATCH_SIZE)
        sess.run ( gd_step , feed_dict ={ x : batch_x , y_true : batch_y })
        ans = sess.run (accuracy , feed_dict ={ x : data.test.images ,
                                                y_true : data.test.labels })

print (" Accuracy : {:.4}% ". format ( ans *100))
```

Accuracy : 88.81%

### 2) Задание 2

```
with tf.Session () as sess :
    # Train
    sess.run (tf.global_variables_initializer ())
    for i in range ( NUM_STEPS ):
        batch_x , batch_y = data.train.next_batch ( MINIBATCH_SIZE )
        sess.run ( gd_step , feed_dict ={ x : batch_x , y_true : batch_y })
        ans = sess.run ( accuracy , feed_dict ={ x : data.test.images ,
                                                y_true : data.test.labels })

print (" Accuracy : {:.4}% ". format ( ans *100))
```

tensorflow.python.layers.core.Dense object at 0x000001C7019E0088>: AssertionError: Bad argument number for Name: 3, expecting 4  
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701A43128>> could not be transformed and will be executed as-is. Please report this to the AutoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH\_VERBOSITY=10`) and attach the full output. Cause: converting <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701A43128>>: AssertionError: Bad argument number for Name: 3, expecting 4  
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701A43128>> could not be transformed and will be executed as-is. Please report this to the AutoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH\_VERBOSITY=10`) and attach the full output. Cause: converting <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701A43128>>: AssertionError: Bad argument number for Name: 3, expecting 4  
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701ABED30>> could not be transformed and will be executed as-is. Please report this to the AutoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH\_VERBOSITY=10`) and attach the full output. Cause: converting <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701ABED30>>: AssertionError: Bad argument number for Name: 3, expecting 4  
WARNING:tensorflow:Entity <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701ABED30>> could not be transformed and will be executed as-is. Please report this to the AutoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH\_VERBOSITY=10`) and attach the full output. Cause: converting <bound method Dense.call of <tensorflow.python.layers.core.Dense object at 0x000001C701ABED30>>: AssertionError: Bad argument number for Name: 3, expecting 4  
Accuracy : 96.56%

## Выводы

В результате выполнения лабораторной работы были получены навыки в создании нейронной сети с 5 полносвязными слоями для классификации набора данных, так же данный набор данных был проанализирован с помощью логистической регрессии

## Ответы на вопросы

1. Что такое Variable?

Тип объектов, которые хранят фиксированные значения в графе.

2. Что такое placeholder?

Тип объектов, которые используют для добавления данных извне модели.

3. Что такое функция потерь?

Функция, определяющая качество работы нашей модели. (вычисляет разницы между предсказанием модели и истинным значением входного вектора)

4. Какие другие названия функции потери?

Функция стоимости.

5. Зачем нужна функция потери?

Смотри вопрос 3

6. Как запустить обучение модели?

Необходимо выполнить метод сессии `.run()`

7. Что делает `tf.global_variables_initializer()`?

Создает в оперативной памяти области для хранения переменных и их исходных значений

8. Что такое minibatch?

Подвыборка, небольшая порция примеров.

9. Какие бывают активационные функции?

Логистическая, тангенсальная и ReLU.

## Литература

[1] Google. Tensorflow. 2018. Feb. url - [https://www.tensorflow.org/install/install\\_windows](https://www.tensorflow.org/install/install_windows).

[2] url - <https://virtualenv.pypa.io/en/stable/userguide/>.

[3] Microsoft. about\_Execution\_Policies. 2018. url - <https://technet.microsoft.com/en-us/library/dd347641.aspx>.

[4] Jupyter Project. Installing Jupyter. 2018. url - <http://jupyter.org/install>.