

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



“Проектирование интеллектуальных систем”

**ЛАБОРАТОРНАЯ РАБОТА № 2. «Логистическая регрессия и полносвязная сеть»**

Студент группы ИУ5-24М

Петропавлов Д.М.

\_\_\_\_\_ Дата

\_\_\_\_\_ Подпись

## Задание:

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

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

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

## Реализация:

### Задание 1)

```
In [3]: import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
NUM_STEPS = 1000
MINIBATCH_SIZE = 100
LEARNING_RATE = 0.5
data = input_data.read_data_sets ("C:/Users/Denis.Petropavlov/virtualenvs/tensorflow/Scripts/data/" , one_hot = True )
```

Extracting C:/Users/Denis.Petropavlov/virtualenvs/tensorflow/Scripts/data/train-images-idx3-ubyte.gz  
Extracting C:/Users/Denis.Petropavlov/virtualenvs/tensorflow/Scripts/data/train-labels-idx1-ubyte.gz  
Extracting C:/Users/Denis.Petropavlov/virtualenvs/tensorflow/Scripts/data/t10k-images-idx3-ubyte.gz  
Extracting C:/Users/Denis.Petropavlov/virtualenvs/tensorflow/Scripts/data/t10k-labels-idx1-ubyte.gz

```
In [4]: 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 ))

with tf.Session (config=tf.ConfigProto(log_device_placement=True)) 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))
```

### Задание 2)

```
In [5]: # Layers sizes
L1 = 200
L2 = 100
L3 = 60
L4 = 30
L5 = 10
```

```
In [6]: 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))
with tf.Session (config=tf.ConfigProto(log_device_placement=True)) 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))
```

## Результаты:

### Задание 1)

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

WARNING:tensorflow:From <ipython-input-4-b895c54f3853>:8: softmax\_cross\_entropy\_with\_logits is deprecated and will be removed in a future version.  
Instructions for updating:

Future major versions of TensorFlow will allow gradients to flow into the labels input on backprop by default.

See tf.nn.softmax\_cross\_entropy\_with\_logits\_v2.

Accuracy : 88.85%

### Задание 2)

```
accuracy = tf.reduce_mean (tf.cast(correct_mask, tf.float32))
with tf.Session (config=tf.ConfigProto(log_device_placement=True)) 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 : batch_y })
        ans = sess.run ( accuracy , feed_dict = { x : batch_x , y : batch_y })
    print (" Accuracy : {:.4}% ". format ( ans *100))
```

Accuracy : 94.93%

## Вывод:

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 (используется для расчета ошибки между реальными и полученными ответами)
4. Какие другие названия функции потери?  
Функция стоимости
5. Зачем нужна функция потери?  
Используется для расчета ошибки между реальными и полученными ответами
6. Как запустить обучение модели?  
Выполнить метод сессии .run()
7. Что делает tf.global\_variables\_initializer()?  
Создает в оперативной памяти области для хранения переменных и их исходных значениях

8. Что такое minibatch?

Подвыборка (минибатч), небольшая порция примеров из общего датасета, объемом от 50 до 500 примеров

9. Какие бывают активационные функции?

Ступенчатая, линейная, сигмоида, гиперболический тангенс и ReLU

**Список литературы**

- [1] Ciresan Dan C., Meier Ueli, Schmidhuber J · urgen. Multi-column Deep Neural Networks for Image Classification // CoRR. 2012. Т. abs/1202.2745. URL: <http://arxiv.org/abs/1202.2745>.
- [2] Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit / Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald [и др.] // Nature. 2000. Т. 405, № 6789. с. 947.
- [3] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. С. 1097–1105.