

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



## **Лабораторная работа №2 по дисциплине «Проектирование интеллектуальных систем»**

**ИСПОЛНИТЕЛЬ:**

Чечнев А.А.

Группа ИУ5-23М

\_\_\_\_\_ 2020 г.

## 1. Задание

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

$$f(x) = \frac{1}{1 + e^{\omega \cdot x + b}}$$

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

### Логистическая регрессия

```
%tensorflow_version 1.x
import tensorflow as tf
tf.__version__
from tensorflow.examples.tutorials.mnist import input_data
DATA_DIR = 'tmp/data'
NUM_STEPS = 1000
MINIBATCH_SIZE = 100
LEARNING_RATE = 0.5
```

### Создание вычислительного графа

```
data = input_data.read_data_sets(DATA_DIR, one_hot=True)
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 = tf.matmul(x, W) + b

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

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))
```

### Запуск сессии

```
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 : 88.84%

---

## Нейронная сеть

```

import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
DATA_DIR = 'tmp/data'
NUM_STEPS = 1000
MINIBATCH_SIZE = 100
LEARNING_RATE = 0.5
# layers sizes
L1 = 200
L2 = 100
L3 = 60
L4 = 30
L5 = 10
data = input_data.read_data_sets(DATA_DIR, one_hot = True)

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 () 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 : 87.21%
```

---

## Контрольные вопросы

### 1. Что такое Variable?

Узел графа с типом Variable имеет переменное значение. После прохода цикла обучения, новое значение узла Variable сохраняется. Его используют для обозначения весов модели.

### 2. Что такое placeholder?

Узел типа placeholder обозначается в виде пустого значения, для дальнейшего вставления в него входных значений.

### 3. Что такое функция потерь?

Функция потерь описывает правила расчета разницы истинных и предсказанных значений.

### 4. Какие другие названия функции потери?

Функция стоимости, разница между предсказанием модели и истинным значением.

### 5. Зачем нужна функция потери?

Для вычисления разницы между предсказанием модели и истинным значением входного вектора.

### 6. Как запустить обучение модели?

В TensorFlow сперва создается оптимизатор с помощью функции GradientDescentOptimizer() с желаемым значением шага обучения. Затем мы создаем операцию в вычислительном графе с помощью optimizer.minimize() и передаем в качестве аргумента функцию потери:

```
optimizer = tf.train.GradientDescentOptimizer ( learning_rate )
```

```
train = optimizer . minimize ( loss )
```

И обучение начинается после того, как операция train передается в качестве аргумента методу sess.run().

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

После запуска `tf.global_variables_initializer()` в сеансе переменные будут содержать значения, которые были указаны для хранения при их объявлении (`tf.Variable`) ...

8. Что такое `minibatch`?

Срез по количеству входных примеров, для расчета на подвыборке, а не на всем множестве входных параметров. Используется в стохастическом градиентном спуске.

9. Какие бывают активационные функции

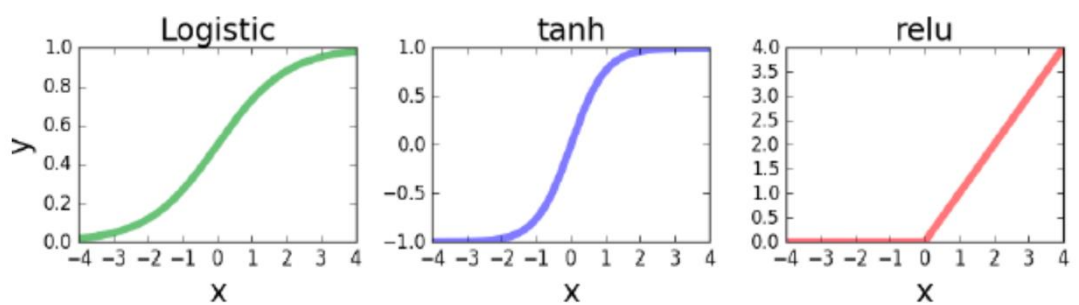


Рис. 3: логистическая, тангенсальная и ReLU активационные функции

## **1. Список литературы**

- 1) Черненко И. М., Методические указания к лабораторной работе №3.
- 2) Николенко С.И., Кадури А.А., Архангельская Е.О. Глубокое обучение. – Издательский дом "Питер", 2019. — 480 с.: ил. — (Серия «Библиотека программиста»).