МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №__7__
по дисциплине «Проектирование интеллектуальных систем»

Тема: «Использование нейронных сетей для анализа текста»

ИСПОЛНИТЕЛЬ: Шапиев М.М.
ФИО
группа ИУ5-24М
подпись
"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ: Терехов В.И.
ФИО
подпись
"__"_____2020 г.

Москва - 2020

Задание

Итоговый код для обучения нейросети и оценки ее точности содержится в приложении. Необходимо увеличить количество скрытых слоев до 3-ех, а количество нейронов в этих слоях так, чтобы обеспечить точность работы нейросети не менее 75%. Темы текстов необходимо изменить в соответствии с вариантом:

comp.windows.x, misc.forsale, comp.windows.x, sci.electronics

Реализация

1) Изменение в нейросети:

```
categories = ["comp.windows.x","misc.forsale","comp.windows.x","sci.electronics"]
newsgroups_train = fetch_20newsgroups(subset='train', categories=categories)
newsgroups_test = fetch_20newsgroups(subset='test', categories=categories)
```

```
def get_batch(df, i, batch_size):
    batches = []
    results = []

    texts = df.data[i * batch_size:i * batch_size + batch_size]
    categories = df.target[i * batch_size:i * batch_size + batch_size]

    for text in texts:
        layer = np.zeros(total_words, dtype=float)
        for word in text.split(' '):
            layer[word2index[word.lower()]] += 1
        batches.append(layer)

    for category in categories:
        y = np.zeros((4), dtype=float)
        if category == 0:
            y[0] = 1.
        elif category == 1:
            y[1] = 1.
        elif category == 2:
            y[2] = 1.
        else:
            y[3] = 1.
        results.append(y)

    return np.array(batches), np.array(results)
```

```
def multilayer_perceptron(input_tensor, weights, biases):
    # скрытый слой 1
    layer_1_multiplication = tf.matmul(input_tensor, weights['h1'])
    layer_1_addition = tf.add(layer_1_multiplication, biases['b1'])
    layer_1 = tf.nn.relu(layer_1_addition)

    # скрытый слой 2
    layer_2_multiplication = tf.matmul(layer_1, weights['h2'])
    layer_2_addition = tf.add(layer_2_multiplication, biases['b2'])
    layer_2 = tf.nn.relu(layer_2_addition)

    # скрытый слой 3
    layer_3_multiplication = tf.matmul(layer_2, weights['h3'])
    layer_3_addition = tf.add(layer_3_multiplication, biases['b3'])
    layer_3 = tf.nn.relu(layer_3_addition)

    # выходной слой
    out_layer_multiplication = tf.matmul(layer_3, weights['out'])
    out_layer_addition = out_layer_multiplication + biases['out']
    return out_layer_addition
```

```
# Параметры обучения
learning_rate = 0.01
training_epochs = 10
batch_size = 150
display_step = 1

# Network Parameters
n_hidden_1 = 300 # скрытый слой 1
n_hidden_2 = 200 # скрытый слой 2
n_hidden_3 = 100 # скрытый слой 3
n_input = total_words # количество уникальных слов в наших текстах
n_classes = 4 # 4 класса

input_tensor = tf.placeholder(tf.float32,[None, n_input],name="input")
output_tensor = tf.placeholder(tf.float32,[None, n_classes],name="output")
```

```
# инициализация параметров сети
weights = {
    'h1': tf.Variable(tf.random_normal([n_input, n_hidden_1])),
    'h2': tf.Variable(tf.random_normal([n_hidden_1, n_hidden_2])),
    'h3': tf.Variable(tf.random_normal([n_hidden_2, n_hidden_3])),
    'out': tf.Variable(tf.random_normal([n_hidden_3, n_classes]))
}
biases = {
    'b1': tf.Variable(tf.random_normal([n_hidden_1])),
    'b2': tf.Variable(tf.random_normal([n_hidden_2])),
    'b3': tf.Variable(tf.random_normal([n_hidden_3])),
    'out': tf.Variable(tf.random_normal([n_classes]))
}
```

2) Результаты:

```
Эпоха: 0001 loss= 31369.3453036221690127
Эпоха: 0002 loss= 3101.2925803444604753
Эпоха: 0003 loss= 2084.2115866921167253
Эпоха: 0004 loss= 1146.3242478804154416
Эпоха: 0005 loss= 1669.0364858453922352
Эпоха: 0006 loss= 11497.2010830965900823
Эпоха: 0007 loss= 3505.7710460316052377
Эпоха: 0008 loss= 1456.5894163305110851
Эпоха: 0009 loss= 126.5116780020973977
Эпоха: 0010 loss= 7.2016653255982828
Обучение завершено!
Точность: 0.75382006
```

Контрольные вопросы

1.Какие вы знаете задачи обработки текстов, в чем они заключаются?

Классификация, кластеризация, построение ассоциативных правил, машинный перевод и др.

2. Зачем нужна предобработка текста для машинного обучения?
Чтобы получить матрицу слов, где каждый элемент соответствует тому или иному слову.
3. Какие виды предобработки текста вы знаете?
Матрицы слов без учёта порядка слов в тексе.
Матрицы слов, где учитывается порядок слов в предложениях.
Удаление шумовых слов.
Стемминг
4. Что такое стемминг?
Процесс замены различных форм слова одной формой и различных синонимов - одним словом.
5. Что такое 20 Newsgroups?
Это набор, состоящий из примерно 20 тысяч постов по 20 различным темам.
6. Чему должно равняться число входных и выходных нейронов в задаче классификации текстов?

Число входных нейронов ровняется числу уникальных слов в наших текстах. Нейроны выходного слоя соответствуют одной из четырёх тем.

Литература

[1] Документация по tensorflow. https://www.tensorflow.org.

[2] Описание 20 Newsgroups. http://qwone.com/~jason/20Newsgroups/

[3] Глубокое обучение для NLP. https://nlp.stanford.edu/courses/NAACL2013/