

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

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

ОТЧЕТ

Лабораторная работа № 1
по дисциплине «Проектирование интеллектуальных систем»

Тема: «Свёрточная нейронная сеть»

ИСПОЛНИТЕЛЬ:

группа ИУ5-22М

Егоров С.А.

ФИО

подпись

"__" _____ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Терехов В.И.

ФИО

подпись

"__" _____ 2020 г.

Москва - 2020

Задание

Обучить нейронную сеть на наборе данных CIFAR10. Точность модели должна достигать 70%. Сеть должна состоять из трех свёрточных слоев и полносвязной сети.

Реализация

1) Описание дополнительных функций:

```
def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)

def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)

def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')

def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

def conv_layer(input, shape):
    W = weight_variable(shape)
    b = bias_variable([shape[3]])
    return tf.nn.relu(conv2d(input, W) + b)

def full_layer(input, size):
    in_size = int(input.get_shape()[1])
    W = weight_variable([in_size, size])
    b = bias_variable([size])
    return tf.matmul(input, W) + b

def get_batch(features, labels, batch_size):
    num_images = features.shape[0]
    idx = np.random.choice(num_images,
                           size=batch_size,
                           replace=False)
    features_batch = features[idx, :, :, :]
    labels_batch = labels[idx, :]
    return features_batch, labels_batch
```

2) Описание слоёв

```
with tf.name_scope('conv_1'):
    conv1 = conv_layer(x , shape=[3, 3, img_channels, 64])
    conv1_pool = max_pool_2x2( conv1 )

with tf.name_scope('conv_2'):
    conv2 = conv_layer(conv1_pool , shape=[3, 3, 64, 128])
    conv2_pool = max_pool_2x2 ( conv2 )

with tf.name_scope('conv_3'):
    conv3 = conv_layer(conv2_pool , shape=[3, 3, 128, 256] )
    conv3_pool = max_pool_2x2 ( conv3 )
    conv3_flat = tf.contrib.layers.flatten(conv3_pool)

with tf.name_scope('full_1'):
    full_1 = tf.nn.relu(full_layer(conv3_flat , 512))

with tf.name_scope('dropout'):
    full1_drop = tf.nn.dropout(full_1 , keep_prob=keep_prob)

with tf.name_scope('activations'):
    y_conv = full_layer(full1_drop , 10)
    tf.summary.scalar('cross_entropy_loss',y_conv)

with tf.name_scope('cross'):
    cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(logits=y_conv , labels=y_))
```

3) Обучение и вывод результатов

```
with tf.device("/gpu:0"):
    with tf.Session() as sess:
        sess.run(tf.global_variables_initializer())
        start_time = time.time()

        for j in range(EPOCH_NUM):
            for i in range(SAMPLES_PER_EPOCH // BATCH_SIZE):
                batch_trainf, batch_trainl = get_batch(train_features,train_labels,BATCH_SIZE)
                sess.run(train_step , feed_dict={x: batch_trainf, y_: batch_trainl, keep_prob: 0.5})
                batch_trainf, batch_trainl = get_batch(test_features,test_labels,32)
                train_accuracy = sess.run(accuracy , feed_dict={x: batch_trainf, y_: batch_trainl, keep_prob: 1.0})
                print("time {}, epoch {}, training accuracy {}".format(time.time() - start_time, j, train_accuracy))

        test_accuracy = np.mean([sess.run(accuracy , feed_dict={x:test_features, y_:test_labels, keep_prob:1.0})])
        print("test accuracy: {}".format(test_accuracy))
```

Результаты

```
time 11290.97111940384, epoch 993, training accuracy 0.6875
time 11302.334324121475, epoch 994, training accuracy 0.75
time 11313.818868875504, epoch 995, training accuracy 0.71875
time 11325.27911233902, epoch 996, training accuracy 0.5625
time 11336.714158296585, epoch 997, training accuracy 0.59375
time 11348.202131271362, epoch 998, training accuracy 0.65625
time 11359.553630590439, epoch 999, training accuracy 0.75
test accuracy: 0.6740000247955322
```

Выводы

В результате выполнения лабораторной работы были получены навыки по созданию свёрточных нейронных сетей.

Ответы на вопросы

1. Что такое свертка?

Свертка — это специализированный вид линейной операции. В сверточных слоях каждый элемент связан с определенным количеством элементов в предыдущем слое.

2. Напишите математическую операцию свертки.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] * h[n - k]$$

3. Какие свойства свёрточного слоя?

- разреженные взаимодействия
- разделение параметров
- эквивариантные изменения.

4. Сколько этапов в свёрточном слое? Какие?

Свёрточный слой можно разбить на три этапа:

- Применение фильтра
- Применение нелинейности
- Применение операции дискретитизации (pooling)

5. Что такое регуляризация? Зачем она нужна?

Это функция для обозначения ограничения оптимизации, чтобы не происходило переобучение. Переобучение происходит тогда, когда модель обучилась таким образом, что показывает идеальные результаты на обучающем наборе, но с плохим результатом на новых данных.

6. Как вид регуляризации использовался в лабораторной?

Dropout. Данный слой с определенной вероятностью "выключает" один из нейронов в слое.

```
with tf.name_scope('dropout'):
    full1_drop = tf.nn.dropout(full_1 , keep_prob=keep_prob)
```

Литература

[1] Backpropagation applied to handwritten zip code recognition / Yann LeCun, Bernhard Boser, John S Denker [и др.] // Neural computation. 1989. Т. 1, No 4. С. 541–551.