

# 人工智能导论作业

---

## 用LeNet-5完成MNIST数据集上的分类

19070231-王雪原

### LeNet-5模型

LeNet-5的整个网络模型分为2个卷积模块、2个池化模块、2个全连接模块，最后再连接一个softmax模块，输出层为10个节点，分别代表1到9共10个数字。

#### 卷积层C1

```
W_conv1 = weight_variable([5, 5, 1, 6])
b_conv1 = bias_variable([6])
h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1)+b_conv1)
```

#### 池化层S2

```
h_pool1 = max_pool_2x2(h_conv1)
```

#### 卷积层C3

```
W_conv2 = weight_variable([5, 5, 6, 16])
b_conv2 = bias_variable([16])
h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2)+b_conv2)
```

#### 池化层S4

```
h_pool2 = max_pool_2x2(h_conv2)
```

#### 全连接层C5

```
W_fc1 = weight_variable([7*7*16, 120])
b_fc1 = bias_variable([120])
h_fc1 = tf.nn.relu(tf.add(tf.matmul(h_pool2_flat, W_fc1), b_fc1))
```

#### 全连接层F6

```
W_fc2 = weight_variable([120, 10])
b_fc2 = bias_variable([10])
h_fc2 = tf.nn.softmax(tf.add(tf.matmul(h_fc1, W_fc2), b_fc2))
```

全连接层Output,计算正确率

```
correct_prediction = tf.equal(tf.argmax(h_fc2, 1), tf.argmax(y, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float32"))
```

## 训练过程

```
for i in range(1000):
    batch_xs, batch_ys = mnist_data_set.train.next_batch(200)
    if i % 2 == 0:
        train_accuracy = accuracy.eval(feed_dict={x: batch_xs, y: batch_ys})
        c.append(train_accuracy)
        print("step %d, train accuracy %g" % (i, train_accuracy))
        train_step.run(feed_dict={x: batch_xs, y: batch_ys})
```

## 训练结果

```
step 986, train accuracy 0.86
step 988, train accuracy 0.92
step 990, train accuracy 0.91
step 992, train accuracy 0.94
step 994, train accuracy 0.925
step 996, train accuracy 0.915
step 998, train accuracy 0.955
```

在不断的训练中,可以看到结果的准确率在提升

