202001555 지은미

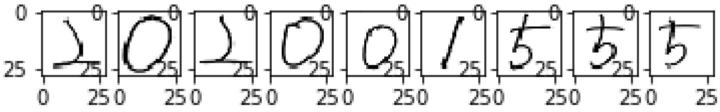
In [69]:

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
import numpy as np
import scipy.special
import scipy.ndimage
%matplotlib inline
import matplotlib.pyplot as plt
import imageio
import glob
```

In [76]:

```
#1-1

paint_correct_list = np.array([])
paint_data_list = np.empty((0,784),float)
for image_file_name in glob.glob('image/?_?.png'):
    correct_label = int(image_file_name[-5:-4])
    paint_correct_list = np.append(paint_correct_list, correct_label)
    print("***correct_label : ", correct_label)

    image_array = imageio.imread(image_file_name)

    x=plt.subplot(1,9,int(image_file_name[-7:-6])+1)
    x=plt.imshow(image_array, cmap='Reds', interpolation = 'None')

    image_array = image_array[:,:,0]

    image_data = 255.0 - image_array.reshape(784)
    image_data = (image_data/255.0*0.99) + 0.01

    paint_data_list = np.append(paint_data_list,[image_data],axis=0)

plt.show()
```

```
***correct_label :  2
***correct_label :  0
***correct_label :  2
***correct_label :  0
***correct_label :  0
***correct_label :  1
***correct_label :  5
***correct_label :  5
***correct_label :  5
```

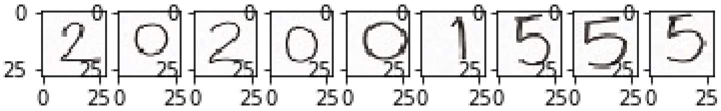


In [81]:

```
#1-2
write_correct_list = np.array([])
write_data_list = np.empty((0,784),float)
for image_file_name in glob.glob('image/number/?_?.jpg'):
    correct_label = int(image_file_name[-5:-4])
    write_correct_list = np.append(write_correct_list, correct_label)
    print("***correct_label : ", correct_label)

    image_array = imageio.imread(image_file_name)

    x=plt.subplot(1,9,int(image_file_name[-7:-6])+1)
    x=plt.imshow(image_array, cmap='Reds', interpolation = 'None')

    image_array = image_array[:,:,0]
```

```
        image_data = 255.0 - image_array.reshape(784)
        image_data = (image_data/255.0*0.99) + 0.01

        write_data_list = np.append(write_data_list,[image_data],axis=0)

    plt.show()
```

```
***correct_label :  2
***correct_label :  0
***correct_label :  2
***correct_label :  0
***correct_label :  0
***correct_label :  1
***correct_label :  5
***correct_label :  5
***correct_label :  5
```



In [87]:

```
#2
class neuralNetwork:
    def __init__(self, inputnodes, hidden1nodes, hidden2nodes, outputnodes, learningr
        self.inodes = inputnodes
        self.h1nodes = hidden1nodes
        self.h2nodes = hidden2nodes
        self.onodes = outputnodes
        self.lr = learningrate

        self.wih1 = np.random.normal(0.0, pow(self.h1nodes, -0.5), (self.h1nodes, sel
        self.wh1h2 = np.random.normal(0.0, pow(self.h2nodes, -0.5), (self.h2nodes, se
        self.wh2o = np.random.normal(0.0, pow(self.onodes, -0.5), (self.onodes, self.

        self.activation_function = lambda x: scipy.special.expit(x)

        pass

    def train(self, inputs_list, targets_list):
        inputs = np.array(inputs_list, ndmin = 2).T
        targets = np.array(targets_list, ndmin = 2).T

        hidden1_inputs = np.dot(self.wih1, inputs)
        hidden1_outputs = self.activation_function(hidden1_inputs)

        hidden2_inputs = np.dot(self.wh1h2, hidden1_outputs)
        hidden2_outputs = self.activation_function(hidden2_inputs)

        final_inputs = np.dot(self.wh2o, hidden2_outputs)
        final_outputs = self.activation_function(final_inputs)

        output_errors = targets - final_outputs

        hidden2_errors = np.dot(self.wh2o.T, output_errors)

        hidden1_errors = np.dot(self.wh1h2.T, hidden2_errors)

        self.wh2o += self.lr * np.dot((output_errors * final_outputs * (1.0 - final_
        self.wh1h2 += self.lr * np.dot((hidden2_errors * hidden2_outputs * (1.0 - hi
        self.wih1 += self.lr * np.dot((hidden1_errors * hidden1_outputs * (1.0 - hid

        pass

    def query(self, inputs_list):
        inputs = np.array(inputs_list, ndmin = 2).T
        hidden1_inputs = np.dot(self.wih1, inputs)
        hidden1_outputs = self.activation_function(hidden1_inputs)

        hidden2_inputs = np.dot(self.wh1h2, hidden1_outputs)
        hidden2_outputs = self.activation_function(hidden2_inputs)

        final_inputs = np.dot(self.wh2o, hidden2_outputs)
        final_outputs = self.activation_function(final_inputs)

        return final_outputs
```

In [88]:

```
input_nodes = 784
hidden_nodes1 = 300
hidden_nodes2 = 100
output_nodes = 10

learning_rate = 0.01

n = neuralNetwork(input_nodes, hidden_nodes1, hidden_nodes2, output_nodes, learning_ra
```

In [89]:

```
training_data_file = open("dataset/mnist_train.csv", 'r')
training_data_list = training_data_file.readlines()
training_data_file.close()

test_data_file = open("dataset/mnist_test.csv", 'r')
test_data_list = test_data_file.readlines()
test_data_file.close()
```

In [90]:

```
epochs = 8

for i in range(epochs):
    for record in training_data_list:
        all_values = record.split(',')
        inputs = (np.asfarray(all_values[1:]) / 255.0 * 0.99) + 0.01

        inputs_rotap10 = scipy.ndimage.interpolation.rotate(inputs.reshape(28, 28), 1

        inputs_rotam10 = scipy.ndimage.interpolation.rotate(inputs.reshape(28, 28), -

        targets = np.zeros(output_nodes) + 0.01
        targets[int(all_values[0])] = 0.99

        n.train(inputs, targets)
        n.train(inputs_rotam10, targets)
        n.train(inputs_rotap10, targets)

        pass
    print("epochs :", i + 1)
```

```
epochs : 1
epochs : 2
epochs : 3
epochs : 4
epochs : 5
epochs : 6
epochs : 7
epochs : 8
```

In [94]:

```
scorecard = []

for record in test_data_list:
    all_values = record.split(',')
    correct_label = int(all_values[0])
    inputs = (np.asfarray(all_values[1:]) / 255.0 * 0.99) + 0.01
    outputs = n.query(inputs)
    label = np.argmax(outputs)
    if label == correct_label:
        scorecard.append(1)
    else:
        scorecard.append(0)
        pass
    pass

scorecard_array = np.asarray(scorecard)
print("performance =", scorecard_array.sum() / scorecard_array.size)
print("hidden node1 =", hidden_nodes1)
print("hidden node2 =", hidden_nodes2)
print("rate =", learning_rate)
print("eposhs =", epochs)
print("rotate =", 10)
```

```
performance = 0.9762
hidden node1 = 300
hidden node2 = 100
rate = 0.01
eposhs = 8
rotate = 10
```

In [92]:

```
# 3-1
scorecard = []

for i, record in enumerate(paint_data_list):
    correct_label = int(paint_correct_list[i])
    outputs = n.query(record)
    label = np.argmax(outputs)
    print("***correct_label :", correct_label)
    print("nn answer is :", label)
    if label == correct_label:
        scorecard.append(1)
    else:
        scorecard.append(0)
        pass
    pass
scorecard_array = np.asarray(scorecard)
print("performance =", scorecard_array.sum() / scorecard_array.size)
```

```
***correct_label : 2
nn answer is : 3
***correct_label : 0
nn answer is : 3
***correct_label : 2
nn answer is : 2
***correct_label : 0
nn answer is : 0
***correct_label : 0
nn answer is : 0
***correct_label : 1
nn answer is : 1
***correct_label : 5
nn answer is : 5
***correct_label : 5
nn answer is : 5
***correct_label : 5
nn answer is : 8
performance = 0.6666666666666666
```

In [93]:

```
# 3-2
scorecard = []

for i, record in enumerate(write_data_list):
    correct_label = int(write_correct_list[i])
    outputs = n.query(record)
    label = np.argmax(outputs)
    print("***correct_label :", correct_label)
    print("nn answer is :", label)
    if label == correct_label:
        scorecard.append(1)
    else:
        scorecard.append(0)
        pass
    pass
scorecard_array = np.asarray(scorecard)
print("performance =", scorecard_array.sum() / scorecard_array.size)
```

```
***correct_label : 2
nn answer is : 2
***correct_label : 0
nn answer is : 9
***correct_label : 2
nn answer is : 2
***correct_label : 0
nn answer is : 0
***correct_label : 0
nn answer is : 5
***correct_label : 1
nn answer is : 3
***correct_label : 5
nn answer is : 5
***correct_label : 5
nn answer is : 5
***correct_label : 5
nn answer is : 5
performance = 0.6666666666666666
```

나만의 손글씨 인식에서 오답이 나오는 이유

1. 노이즈 문제

종이 손글씨의 이미지를 스캔하다보면 노이즈가 생길 수 밖에 없다. 이러한 노이즈때문에 손글씨로 쓴 숫자를 잘 인식하지 못해 정확도가 떨어지게 된다.

1. 사용자문제

데이터셋의 크기가 많지 않을 뿐더러 사용자가 쓴 글씨가 데이터셋에 존재하지 않을 수도 있다. 그렇게 되면 정답을 맞추는데 어려움이 있을 수 있다.

202001555 지은미