

00

Assignment08.ipynb

☆

파일 수정 보기 삽입 런타임 도구 도움말 오후 8:40에 마지막으로 저장됨

찾기 공유 순회 순회

파일

입력도 새로고침

드라이브 머문드 현재

drive

My Drive

Abroad_Program

Classroom

Codeb Notebooks

Fall2018

Fall2019

Spring2018

Spring2019

Spring2020

Compiler

Database_System

Information_Security_Theory

Machine_Learning

CAU-MachineLearning

Assignment01

Assignment02

Assignment03

Assignment04

Assignment05

Assignment06

Assignment07

Assignment08

Assignment08.ipynb

assignment_08_a...

mnist_test.csv

README.md

PPT

testing_rpo

Multicore.Computing

Network_Application_and_D...

다스크 76.75 GB 사용 가능

RAM 디스크

수정 가능

Information

Writer : Junhyuck Woo
Std_ID : 20145337
Assignment08 : Forward Propagation in the Neural Networks
Deadline : May 21, 2020

Library

[] import matplotlib.pyplot as plt; import numpy as np

Data

[] file_data = "/content/drive/My Drive/Spring2020/Machine_Learning/CAU-MachineLearning/Assignment08/mnist_test.csv"
handle_file = open(file_data, "r")
data = handle_file.readlines()
handle_file.close()

size_row = 28 # height of the image
size_col = 28 # width of the image

num_image = len(data)
count = 0 # count for the number of images

make a matrix each column of which represents an images in a vector form

list_image = np.empty((size_row * size_col, num_image), dtype=float)
list_label = np.empty(num_image, dtype=int)

for line in data:
 line_data = line.split(',')
 label = line_data[0]
 im_vector = np.asfarray(line_data[1:])

 list_label[count] = label
 list_image[:, count] = im_vector / 255

 count += 1

Average Calculation

[] # Variable
num_digit_list = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
num_digit = len(num_digit_list)
avg_image = np.zeros((size_col * size_row, num_digit))

for i in range(num_image):
 # Get digit
 digit = list_label[i]

 # Count the number of each digit
 num_digit_list[digit] = num_digit_list[digit] + 1

 # Cumulate the image
 avg_image[:, digit] = avg_image[:, digit] + list_image[:, i]

Divide as number of digit for getting the average image
for i in range(num_digit):
 avg_image[:, i] = avg_image[:, i] / num_digit_list[i]

Generate Theta with N(0, 1)

[] mean = 0
sigma = 1
theta = np.random.normal(mean, sigma, size_col*size_row)

Comput the Forward Propagation

[] # Variable
result_list = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

for i in range(num_image):
 # Get digit
 digit = list_label[i]

 # Compute the propagation
 prop = np.sum(list_image[:, i] * theta)

 # Cumulate the result
 result_list[digit] = result_list[digit] + prop

for i in range(num_digit):
 # Average
 result_list[i] = result_list[i] / num_digit_list[i]

Sigmoid
result_list[i] = 1 / (1 + np.exp(-result_list[i]))

Result

1. Average Image

[] fl = plt.figure(figsize=(10,6))
plt.subplot('Average image', fontsize = 40)
for i in range(10):

 im_vector = avg_image[:, i]
 im_matrix = im_vector.reshape((size_row, size_col))

 plt.subplot(2, 5, i+1)
 plt.title(i, fontsize=20)
 plt.imshow(im_matrix, cmap='Greys', interpolation='None')

 frame = plt.gca()
 frame.axes.get_xaxis().set_visible(False)
 frame.axes.get_yaxis().set_visible(False)

plt.show()

Average Image

2. Output of the Neural Network with Random Weights

[] for index, value in enumerate(result_list):
 print("Label %d: %f" % (index, value))Label 0: 0.000355
Label 1: 0.007689
Label 2: 0.000867
Label 3: 0.009409
Label 4: 0.569556
Label 5: 0.007554
Label 6: 0.000002
Label 7: 0.130829
Label 8: 0.023124
Label 9: 0.056863