Nama : Hafidh Fikri Rasyid

NIM : 1301142190

Kelas : IF-38-Gab1

Assignment 1

Machine Learning

Programming Section

7. Pada latihan ini, anda diharuskan mengimplementasikan proses klasifikasi menggunakan *classifier* yang sangat sederhana berbasiskan prototype, disebut sebagai *prototype-based classifier*, untuk mengklasifikasi tulisan digit menggunakan dataset MNIST, kemudian membandingkan hasilnya dengan *nearest-neighbor classifier*. Download dataset MNIST yang telah dikirimkan melalui email dan Telegram. Pada dataset tersebut terdapat juga beberapa fungsi yang mempermudah proses *load* data untuk bahasa pemrograman Matlab, Octave dan R serta beberapa contoh *codingan* yang dapat digunakan. Lihat file README untuk keterangan lebih lanjut.
   a. Load 5000 image pertama menggunakan fungsi yang telah disediakan. Gunakan fungsi yang telah disediakan untuk melakukan plot terhadap 100 data image yang anda pilih secara acak dan tunjukkan label-label dari image tersebut. Lakukan verifikasi bahwa label-label tersebut sesuai dengan masing-masing gambar digitnya. (Verfikasi ini adalah suatu keharusan untuk membuktikan bahwa data yang anda miliki sudah dalam format yang benar.)
   b. Bagilah data menjadi dua bagian, 'data training' terdiri atas 2500 gambar (beserta label) dan 'data testing' terdiri atas 2500 gambar (beserta label). Untuk setiap 10 kelas (digit 0-9), buatlah prototype dari masing-masing kelas dengan cara menggunakan nilai rata-rata dari seluruh gambar pada data training untuk kelas yang sama. Sebagai contoh, lakukan pengambilan seluruh gambar dari kelas 0 (pada data training) kemudian hitung rata-rata dari gambar tersebut. Nilai rata-rata inilah yang kita sebut sebagai prototype. Lakukan hal tersebut untuk seluruh kelas dan plot hasilnya. Apakah hasil tersebut sesuai dengan yang anda inginkan?
   c. Untuk setiap gambar di data testing, hitunglah jarak Euclidean-nya terhadap 10 prototype yang telah dihasilkan dari soal 7b, kemudian klasifikasikan gambar tersebut kedalam kelas yang memiliki jarak Euclidean terkecil. Oleh karena itu, jika suatu gambar di data testing memiliki jarak yang paling dekat dengan prototype '3', maka kelas dari gambar tersebut adalah kelas 3. Hitung dan tampilkan hasil klasifikasi tersebut dalam bentuk *confusion matrix*.
   d. Klasifikasikan setiap data dari data testing menggunakan *nearest neighbor classifier*. Caranya: untuk setiap data testing, hitunglah jarak Euclidean-nya terhadap seluruh data training (2500 gambar), kemudian kelas prediksi dari gambar testing tersebut ditentukan oleh kelas dari gambar (di data training) yang paling dekat dengan data testing tersebut. Hitung dan tampilkan hasil klasifikasi dalam bentuk confusion matrix.
   e. Hitung dan bandingkan nilai error dari kedua classifier tersebut (*prototype-based classifier* dan *nearest neighbor classifier*). Classifier mana yang memberikan hasil yang terbaik? Berdasarkan confusion matrix yang telah diperoleh, digit mana yang paling banyak salah diklasifikasikan kedalam digit lain? Mengapa demikian?

Cara Menjalankan Aplikasi :

Cara menjalankan program dapat dilakukan dengan du acara. Yang pertama buka masing-masing file yang dijelaskan pada bagian penjelasan aplikasi dibawah lalu klik run pada masing-masing file. Cara yang kedua adalah menjalankan file executabledari masing-masing file yang dijelaskan di bawah.

Catatan : jika menjalankan aplikasi dengan file executable memerlukan waktu yang cukup lama agar program dapat running. Untuk program no7a dan no7b akan muncul jendela plot baru disamping program executable.

Penjelasan Aplikasi :

Program dibagi menjadi lima buah file :

1. no7a.m: Berisi jawaban dari soal a
2. no7b.m: Berisi jawaban dari soal b
3. no7c.m: Berisi jawaban dari soal c
4. no7d.m: Berisi jawaban dari soal d
5. ConfMat.m : Berisi fungsi dari menghitung confusion matrix yang saya buat sendiri

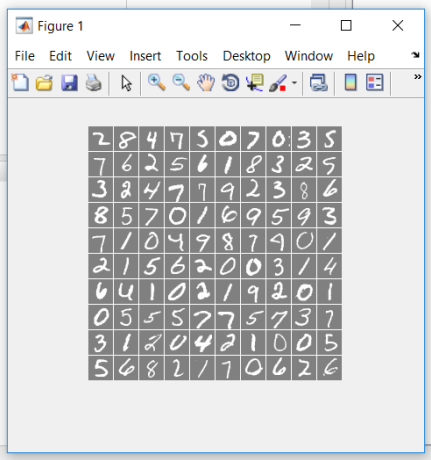
Jawaban :

a. Pada bagian ini saya me-*load* 5000 gambat menggunakan fungsi yang telah disediakan. Setelah itu saya menapilkan 100 gambar secara acak menggunakan strategi pemanggilan index dari gambar dengan memanggil fungsi randperm() agar tidak ada angka dari index gambar yang berulang. Setelah menampilkan data sebanyak 100 buah gambar, saya melakukan verifikasi dengan menggunakan cara yang sama. Untuk hasil dapat dilihat dibawah ini :

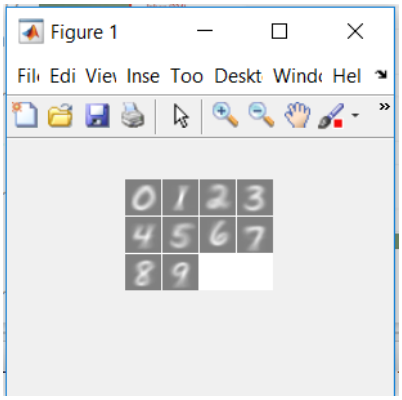
```
194     %     akurasi = (tp+tn)/length(LblPred)
195     %     error_rate = (fp+fn)/length(LblPred)
196     %     tp_rate = tp/(fn+tp)
197     %     fp_rate = fp/(tn+fp)
198     %     specificity = tn/(tn+fp)
199     %     precision = tp/(fp+tp)
200     %     prevalence = (fn+tp)/length(LblPred)
201     % end
202
```

Command Window

```
Verifikasi data dengan yang ada di plot
     2     8     4     7     5     0     7     0     3     5
     7     6     2     5     6     1     8     3     2     5
     3     2     4     7     7     9     2     3     8     6
     8     5     7     0     1     6     9     5     9     3
     7     1     0     4     9     8     7     4     0     1
     2     1     5     6     2     0     0     3     1     4
     6     4     1     0     2     1     9     2     0     1
     0     5     5     5     7     7     5     7     3     7
     3     1     2     0     4     2     1     0     0     5
     5     6     8     2     1     7     0     6     2     6
```



b. Data berhasil saya pisahkan perkelas dan berhasil saya hitung rata-rata dari setiap kelas. Untuk data nilai rata-rata perkelas saya tamping dalam sebuah variabel. Setelah dihitung rata-rata nilai tersebut, saya mencoba menampilkan gambar yang berasal dari data yang telah dirata-ratakan tersebut (gambar prototype). Dan Setelah ditampilkan, gambar yang tampil sesuai dengan yang diinginkan yaitu muncul angka 0 hingga 9 yang dapat dilihat dengan jelas secara visual. Berikut adalah hasil gambar dari pengerjaan soal ini beserta gambar sebagian data dari array penampung yang disimpan di variabel array yang bernama arrMean :



Editor - Prog2.m | Variables - arrMean

arrMean

10x784 double

	95	96	97	98	99	100	101	102	103	104	105	106	107	108
1	0	0.2711	1.3333	3.8089	3.9867	5.5067	5.2889	3.6844	1.5156	1.1644	0.8756	0	0	0
2	0	0	0.0073	1.5927	2.5491	1.0436	0.0182	0	0.6364	0.5345	0.8073	0.3673	0	0
3	22.4576	29.8771	39.4958	47.3941	48.1398	44.4237	36.5127	27.7331	17.6780	8.5339	4.3941	2.2585	1.1864	0.0339
4	2.7531	3.5314	2.7615	2.2301	1.9707	1.8494	2.5941	2.2762	1.0669	1	0.2134	0	0	0
5	0	0	0	0	0	0	0	0	0.2197	0.0227	0.8902	1.9091	1.1591	0
6	0	0.2355	1.0785	0.6942	2.4298	2.7397	2.5372	2.3512	1.6983	2.3678	2.4959	2.6736	2.9050	1.2273
7	11.7160	16.2000	18.1960	28.5600	50.2360	78.5120	95.1280	95.0800	79.5640	58.0120	26.6080	9.6480	3.4360	1.1760
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0.0553	0.4213	0.9574	1.2596	1.3234	1.9915	0.7830	1.0851	0.3660	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11														
12														
13														
14														
15														
16														
17														
18														

Untuk gambar data diatas setiap kelas dari 0 hingga 9 direpresentasikan berdasarkan baris pada array (array index baris ke 1 merepresentasikan data rata-rata dari kelas 0, array index baris ke 2 merepresentasikan data rata-rata dari kelas 1, array index baris ke 3 merepresentasikan data rata-rata dari kelas 2 dan seterusnya).

c. Proses yang saya lakukan dengan menggunakan classifier prototype-based menghasilkan confusion matrix sebagai berikut :

Command Window

```
Data ke 2500
Confusion Matrix pada kelas 0
Nilai True Positive = 220
Nilai True Negative = 2237
Nilai False Positive = 17
Nilai False Negative = 26
Nilai Akurasi= 0.9828
Nilai Error Rate = 0.0172

Confusion Matrix pada kelas 1
Nilai True Positive = 271
Nilai True Negative = 2116
Nilai False Positive = 105
Nilai False Negative = 8
Nilai Akurasi= 0.9548
Nilai Error Rate = 0.0452

Confusion Matrix pada kelas 2
Nilai True Positive = 178
Nilai True Negative = 2225
Nilai False Positive = 31
Nilai False Negative = 66
Nilai Akurasi= 0.9612
Nilai Error Rate = 0.0388
```

Command Window

```
Confusion Matrix pada kelas 3
Nilai True Positive = 183
Nilai True Negative = 2189
Nilai False Positive = 69
Nilai False Negative = 59
Nilai Akurasi= 0.9488
Nilai Error Rate = 0.0512

Confusion Matrix pada kelas 4
Nilai True Positive = 235
Nilai True Negative = 2167
Nilai False Positive = 58
Nilai False Negative = 40
Nilai Akurasi= 0.9608
Nilai Error Rate = 0.0392

Confusion Matrix pada kelas 5
Nilai True Positive = 134
Nilai True Negative = 2221
Nilai False Positive = 62
Nilai False Negative = 83
Nilai Akurasi= 0.942
Nilai Error Rate = 0.058
```

Command Window

```
Confusion Matrix pada kelas 6
Nilai True Positive = 213
Nilai True Negative = 2232
Nilai False Positive = 27
Nilai False Negative = 28
Nilai Akurasi= 0.978
Nilai Error Rate = 0.022

Confusion Matrix pada kelas 7
Nilai True Positive = 243
Nilai True Negative = 2193
Nilai False Positive = 23
Nilai False Negative = 41
Nilai Akurasi= 0.9744
Nilai Error Rate = 0.0256

Confusion Matrix pada kelas 8
Nilai True Positive = 150
Nilai True Negative = 2243
Nilai False Positive = 30
Nilai False Negative = 77
Nilai Akurasi= 0.9572
Nilai Error Rate = 0.0428
```

```
Confusion Matrix pada kelas 9
Nilai True Positive = 188
Nilai True Negative = 2192
Nilai False Positive = 63
Nilai False Negative = 57
Nilai Akurasi= 0.952
Nilai Error Rate = 0.048

fx >>
```

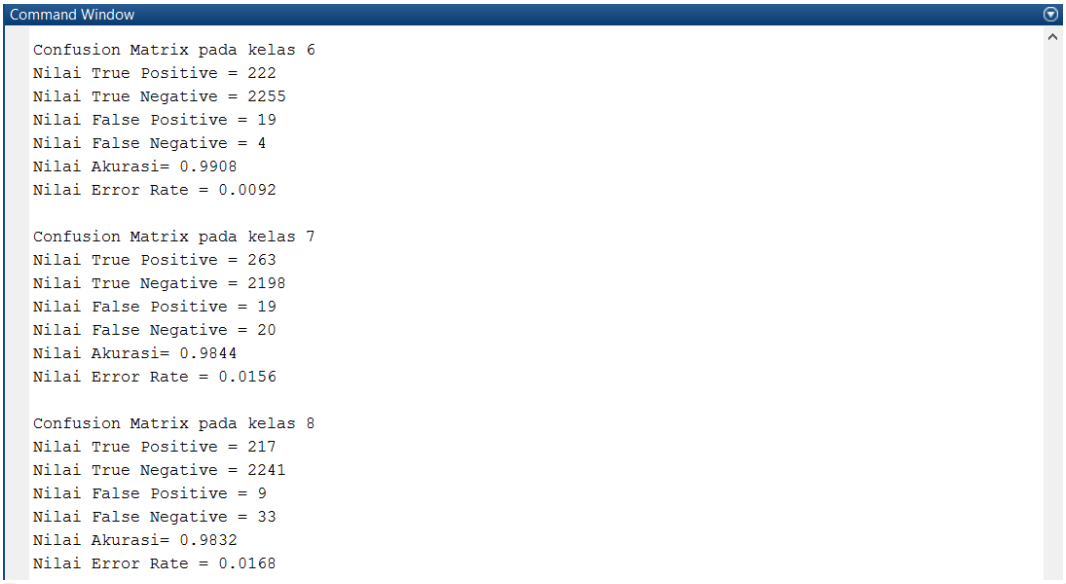
d. Proses yang saya lakukan dengan menggunakan classifier nearest neighbor menghasilkan confusion matrix sebagai berikut :

Command Window

```
Soal D
Confusion Matrix pada kelas 0
Nilai True Positive = 233
Nilai True Negative = 2254
Nilai False Positive = 9
Nilai False Negative = 4
Nilai Akurasi= 0.9948
Nilai Error Rate = 0.0052

Confusion Matrix pada kelas 1
Nilai True Positive = 282
Nilai True Negative = 2183
Nilai False Positive = 27
Nilai False Negative = 8
Nilai Akurasi= 0.986
Nilai Error Rate = 0.014

Confusion Matrix pada kelas 2
Nilai True Positive = 216
Nilai True Negative = 2238
Nilai False Positive = 16
Nilai False Negative = 30
Nilai Akurasi= 0.9816
Nilai Error Rate = 0.0184
```

Command Window

```
Confusion Matrix pada kelas 3
Nilai True Positive = 218
Nilai True Negative = 2250
Nilai False Positive = 16
Nilai False Negative = 16
Nilai Akurasi= 0.9872
Nilai Error Rate = 0.0128

Confusion Matrix pada kelas 4
Nilai True Positive = 242
Nilai True Negative = 2217
Nilai False Positive = 17
Nilai False Negative = 24
Nilai Akurasi= 0.9836
Nilai Error Rate = 0.0164

Confusion Matrix pada kelas 5
Nilai True Positive = 183
Nilai True Negative = 2276
Nilai False Positive = 16
Nilai False Negative = 25
Nilai Akurasi= 0.9836
Nilai Error Rate = 0.0164
```




```
Confusion Matrix pada kelas 9
Nilai True Positive = 238
Nilai True Negative = 2202
Nilai False Positive = 38
Nilai False Negative = 22
Nilai Akurasi= 0.976
Nilai Error Rate = 0.024
```

e. Dari kedua percobaan menggunakan classifier yang berbeda diatas, dapat disimpulkan bahwa classifier yang memberikan hasil yang terbaik adalah dengan menggunakan classifier nearest neighbor. Alasan dari terpilihnya classifier nearest neighbor adalah pada classifier ini adalah karena nilai rata-rata dari error rate yang dihasilkan berdasarkan confusion matrix diatas lebih kecil dibandingkan dengan classifier prototype-based. Untuk digit yang paling banyak salah diklasifikasikan ke digit yang lain berdasarkan nilai confusion matrix diatas adalah angka 2 (dua). Alasannya adalah, angka 2 (dua) pada dataset lebih mirip angka 7(tujuh) dan komputer salah mengklasifikasikannya meskipun kesalahan yang terjadi tidak terlalu sering terbukti di kedua model klasifikasi nilai error rate yang ditunjukkan tidak lebih dari 0.04 .