

**ΟΝΟΜΑ : Γιώργος Παναγιωτάτος**

**ΑΕΜ : 2627**

### **Νευρωνικά πρώτη εργασία**

Η υλοποίηση της εργασίας έγινε σε python 3.6 . Χρησιμοποίησα το dataset της mnist. Σε όλες τις περιπτώσεις φόρτωσα τα δεδομένα στο πρόγραμμα μου, τα χώρισα σε train set 60χιλιαδων δειγμάτων και σε test set 10χιλιαδων δειγμάτων, και έκανα τις απαραίτητες μετατροπές που χρειάζεται στις εικόνες για να μπορεί το μοντέλο μου να τις πάρει σαν όρισμα, έπειτα κανονικοποίησα τις τιμές των pixel μεταξύ του 0 και 1 με την χρήση βιβλιοθήκης.

Χρησιμοποιώ το dense format με 3 κρυφά στρώματα από 20 νευρώνες κάθε στρώμα, το batch size είναι 32, adam σαν optimizer όπου δεν ορίζεται. Στο πρώτο, δεύτερο και τρίτο κρυφό στρώμα χρησιμοποιώ για συνάρτηση ενεργοποίησης την relu και στο στρώμα εξόδου την sigmoid.

1)Με 10 εποχές

```
Epoch 5/10
60000/60000 [=====] - 4s - loss: 0.0081 - acc: 0.95
13 - val_loss: 0.0081 - val_acc: 0.9505
Epoch 6/10
60000/60000 [=====] - 4s - loss: 0.0076 - acc: 0.95
42 - val_loss: 0.0080 - val_acc: 0.9515
Epoch 7/10
60000/60000 [=====] - 4s - loss: 0.0070 - acc: 0.95
84 - val_loss: 0.0086 - val_acc: 0.9466
Epoch 8/10
60000/60000 [=====] - 4s - loss: 0.0066 - acc: 0.96
01 - val_loss: 0.0072 - val_acc: 0.9553
Epoch 9/10
60000/60000 [=====] - 4s - loss: 0.0062 - acc: 0.96
20 - val_loss: 0.0077 - val_acc: 0.9526
Epoch 10/10
60000/60000 [=====] - 4s - loss: 0.0059 - acc: 0.96
49 - val_loss: 0.0081 - val_acc: 0.9515
```

Out[51]: <keras.callbacks.History at 0x22a85ec6f98>

In [ ]:

## 2)Με 500 εποχές

```
model.fit(x_train, y_train, epochs=500, batch_size=32, verbose=1, validation_data=(x_val, y_val))
predictions = model.predict(x_train)
```

```
<
60000/60000 [=====] - 3s - loss: 0.0020 - acc: 0.9895 - val_loss: 0.0081 - val_acc: 0.9584
Epoch 497/500
60000/60000 [=====] - 3s - loss: 0.0019 - acc: 0.9895 - val_loss: 0.0078 - val_acc: 0.9592
Epoch 498/500
60000/60000 [=====] - 3s - loss: 0.0021 - acc: 0.9891 - val_loss: 0.0083 - val_acc: 0.9583
Epoch 499/500
60000/60000 [=====] - 3s - loss: 0.0020 - acc: 0.9894 - val_loss: 0.0084 - val_acc: 0.9580
Epoch 500/500
60000/60000 [=====] - 3s - loss: 0.0020 - acc: 0.9896 - val_loss: 0.0088 - val_acc: 0.9555
```

## 3)Με rca στις 100 διαστάσεις

```
10 - val_loss: 0.0077 - val_acc: 0.9521
Epoch 6/10
60000/60000 [=====] - 3s - loss: 0.0073 - acc: 0.9553 - val_loss: 0.0074 - val_acc: 0.9540
Epoch 7/10
60000/60000 [=====] - 3s - loss: 0.0069 - acc: 0.9581 - val_loss: 0.0073 - val_acc: 0.9553
Epoch 8/10
60000/60000 [=====] - 3s - loss: 0.0066 - acc: 0.9597 - val_loss: 0.0070 - val_acc: 0.9566
Epoch 9/10
60000/60000 [=====] - 3s - loss: 0.0063 - acc: 0.9623 - val_loss: 0.0075 - val_acc: 0.9546
Epoch 10/10
60000/60000 [=====] - 3s - loss: 0.0061 - acc: 0.9627 - val_loss: 0.0072 - val_acc: 0.9548
```

Out[54]: <keras.callbacks.History at 0x22a98a0de48>

In [ ]:

#### 4) Με pca στις 50 διαστάσεις

```
Epoch 5/10
60000/60000 [=====] - 3s - loss: 0.0079 - acc: 0.95
14 - val_loss: 0.0080 - val_acc: 0.9500
Epoch 6/10
60000/60000 [=====] - 3s - loss: 0.0074 - acc: 0.95
44 - val_loss: 0.0077 - val_acc: 0.9520
Epoch 7/10
60000/60000 [=====] - 3s - loss: 0.0071 - acc: 0.95
65 - val_loss: 0.0076 - val_acc: 0.9534
Epoch 8/10
60000/60000 [=====] - 3s - loss: 0.0068 - acc: 0.95
82 - val_loss: 0.0074 - val_acc: 0.9540
Epoch 9/10
60000/60000 [=====] - 3s - loss: 0.0066 - acc: 0.95
91 - val_loss: 0.0074 - val_acc: 0.9538
Epoch 10/10
60000/60000 [=====] - 3s - loss: 0.0064 - acc: 0.96
07 - val_loss: 0.0071 - val_acc: 0.9558

Out[53]: <keras.callbacks.History at 0x22a8d0cd7f0>
```

In [ ]:

#### 5) Με 10 κρυφά στρώματα

```
60000/60000 [=====] - 5s - loss: 0.0123 - acc: 0.92
40 - val_loss: 0.0129 - val_acc: 0.9210
Epoch 6/10
60000/60000 [=====] - 5s - loss: 0.0113 - acc: 0.92
98 - val_loss: 0.0108 - val_acc: 0.9313
Epoch 7/10
60000/60000 [=====] - 5s - loss: 0.0104 - acc: 0.93
59 - val_loss: 0.0107 - val_acc: 0.9330
Epoch 8/10
60000/60000 [=====] - 5s - loss: 0.0098 - acc: 0.94
00 - val_loss: 0.0104 - val_acc: 0.9348
Epoch 9/10
60000/60000 [=====] - 5s - loss: 0.0091 - acc: 0.94
35 - val_loss: 0.0102 - val_acc: 0.9372
Epoch 10/10
60000/60000 [=====] - 5s - loss: 0.0088 - acc: 0.94
56 - val_loss: 0.0091 - val_acc: 0.9432

55]: <keras.callbacks.History at 0x22a99df5f60>
```

[ ]:

## 6) Με ένα κρυφό στρώμα

```
/4 - val_loss: 0.0134 - val_acc: 0.9260
Epoch 6/10
60000/60000 [=====] - 3s - loss: 0.0130 - acc: 0.92
97 - val_loss: 0.0130 - val_acc: 0.9261
Epoch 7/10
60000/60000 [=====] - 3s - loss: 0.0126 - acc: 0.93
14 - val_loss: 0.0127 - val_acc: 0.9325
Epoch 8/10
60000/60000 [=====] - 3s - loss: 0.0124 - acc: 0.93
22 - val_loss: 0.0125 - val_acc: 0.9327
Epoch 9/10
60000/60000 [=====] - 3s - loss: 0.0122 - acc: 0.93
41 - val_loss: 0.0125 - val_acc: 0.9318
Epoch 10/10
60000/60000 [=====] - 3s - loss: 0.0120 - acc: 0.93
45 - val_loss: 0.0125 - val_acc: 0.9322
```

Out[56]: <keras.callbacks.History at 0x22aadfaaf98>

In [ ]:

## 7) Χρησιμοποιώντας για την συνάρτηση απώλειας την 'mean\_squared\_logarithmic\_error'

```
73 - val_loss: 0.0041 - val_acc: 0.9500
Epoch 5/10
60000/60000 [=====] - 4s - loss: 0.0037 - acc: 0.95
32 - val_loss: 0.0038 - val_acc: 0.9520
Epoch 6/10
60000/60000 [=====] - 4s - loss: 0.0034 - acc: 0.95
67 - val_loss: 0.0037 - val_acc: 0.9540
Epoch 7/10
60000/60000 [=====] - 4s - loss: 0.0032 - acc: 0.95
88 - val_loss: 0.0035 - val_acc: 0.9566
Epoch 8/10
60000/60000 [=====] - 4s - loss: 0.0031 - acc: 0.96
11 - val_loss: 0.0034 - val_acc: 0.9574
Epoch 9/10
60000/60000 [=====] - 4s - loss: 0.0029 - acc: 0.96
37 - val_loss: 0.0034 - val_acc: 0.9565
Epoch 10/10
60000/60000 [=====] - 4s - loss: 0.0028 - acc: 0.96
54 - val_loss: 0.0033 - val_acc: 0.9570
```

Out[57]: <keras.callbacks.History at 0x22aaed33e80>

In [ ]:

## 8) Χρησιμοποιώντας ως optimizer τον 'sgd'

```
Epoch 6/10  
60000/60000 [=====] - 4s - loss: 0.0877 - acc: 0.33  
38 - val_loss: 0.0869 - val_acc: 0.3679  
Epoch 7/10  
60000/60000 [=====] - 4s - loss: 0.0858 - acc: 0.37  
50 - val_loss: 0.0844 - val_acc: 0.3802  
Epoch 8/10  
60000/60000 [=====] - 4s - loss: 0.0826 - acc: 0.39  
53 - val_loss: 0.0800 - val_acc: 0.4100  
Epoch 9/10  
60000/60000 [=====] - 4s - loss: 0.0775 - acc: 0.42  
94 - val_loss: 0.0743 - val_acc: 0.4607  
Epoch 10/10  
60000/60000 [=====] - 4s - loss: 0.0717 - acc: 0.47  
48 - val_loss: 0.0685 - val_acc: 0.5127
```

Out[60]: <keras.callbacks.History at 0x22a861b02e8>

In [ ]:

## 9) Χρησιμοποιώντας διάφορες συναρτήσεις ενεργοποίησης

```
87 - val_loss: 0.0077 - val_acc: 0.9514  
Epoch 6/10  
60000/60000 [=====] - 3s - loss: 0.0064 - acc: 0.96  
15 - val_loss: 0.0072 - val_acc: 0.9535  
Epoch 7/10  
60000/60000 [=====] - 3s - loss: 0.0060 - acc: 0.96  
39 - val_loss: 0.0071 - val_acc: 0.9539  
Epoch 8/10  
60000/60000 [=====] - 3s - loss: 0.0057 - acc: 0.96  
59 - val_loss: 0.0071 - val_acc: 0.9552  
Epoch 9/10  
60000/60000 [=====] - 3s - loss: 0.0054 - acc: 0.96  
73 - val_loss: 0.0073 - val_acc: 0.9522  
Epoch 10/10  
60000/60000 [=====] - 3s - loss: 0.0052 - acc: 0.96  
93 - val_loss: 0.0073 - val_acc: 0.9534
```

Out[4]: <keras.callbacks.History at 0x1fc9d8d8fd0>

In [ ]:

10) Στο τέλος κατέληξα, μετα από πολλές δοκιμές, σε ένα μοντέλο που χρησιμοποιεί rca για να πέσει στις 100 διαστάσεις και σε 3 κρυφά στρώματα των 200 percerptrons το κάθε ένα και παίρνω αυτά τα αποτελέσματα

```
77 - val_loss: 0.0033 - val_acc: 0.9791
Epoch 6/10
60000/60000 [=====] - 4s - loss: 0.0018 - acc: 0.98
90 - val_loss: 0.0036 - val_acc: 0.9785
Epoch 7/10
60000/60000 [=====] - 4s - loss: 0.0016 - acc: 0.99
04 - val_loss: 0.0034 - val_acc: 0.9800
Epoch 8/10
60000/60000 [=====] - 4s - loss: 0.0015 - acc: 0.99
10 - val_loss: 0.0037 - val_acc: 0.9771
Epoch 9/10
60000/60000 [=====] - 4s - loss: 0.0013 - acc: 0.99
22 - val_loss: 0.0032 - val_acc: 0.9803
Epoch 10/10
60000/60000 [=====] - 4s - loss: 0.0013 - acc: 0.99
24 - val_loss: 0.0034 - val_acc: 0.9799
```

Out[6]: <keras.callbacks.History at 0x1fca05124e0>

In [ ]: