

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



**ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΑΛΓΟΡΙΘΜΙΚΑ ΠΡΟΒΛΗΜΑΤΑ**

**2η Εργασία:**

*Εκπαίδευση και αποτίμηση νευρωνικού δικτύου αυτοκωδικοποίησης  
εικόνων αριθμητικών ψηφίων. Χρήση του κωδικοποιητή για τη δημιουργία  
νευρωνικού δικτύου κατηγοριοποίησης των εικόνων*

ΒΕΚΡΑΚΗ ΑΘΗΝΑ

1115201400020

ΠΟΛΥΧΡΟΝΑΚΗΣ ΣΑΒΒΑΣ

1115201200150

## ΠΕΡΙΕΧΟΜΕΝΑ

Αρχεία για τον Autoencoder:

[autoencoder.py](#) | [autoencoder\\_functions.py](#)

Αρχεία για τον Classification:

[classification.py](#) | [classification\\_functions.py](#)

Αποθηκευμένο μοντέλο Autoencoder:

[autoencoder.h5](#)

καθώς και τα datasets [t10k-images-idx3-ubyte](#), [t10k-labels-idx1-ubyte](#), [train-images-idx3-ubyte](#) και [train-labels-idx1-ubyte](#)

## ΓΕΝΙΚΑ

Autoencoder: Αρχικά, το πρόγραμμα διαβάζει το dataset και το χωρίζει σε σύνολο εκπαίδευσης (training set) και σε σύνολο επικύρωσης (validation set). Στη συνέχεια ζητάει από το χρήστη τιμές για τις υπερπαραμέτρους (number of layers, filter size, number of filters in first layer, epochs, batch size) και κατασκευάζει το μοντέλο για τις συγκεκριμένες τιμές. Έπειτα με το μοντέλο που έχει κατασκευαστεί ξεκινάει η εκπαίδευσή του και στο τέλος αυτής ζητάει από το χρήστη αν θέλει να επαναλάβει το πείραμα με νέες τιμές υπερπαραμέτρων (επιλογή 1), αν θέλει να εμφανίσει τις γραφικές παραστάσεις του σφάλματος ως προς τις τιμές των υπερπαραμέτρων για τα εκτελεσθέντα πειράματα (επιλογή 2) και αν θέλει να αποθηκεύσει το μοντέλο που έχει εκπαιδευτεί με τις τελευταίες τιμές υπερπαραμέτρων (επιλογή 3).

Classification: Αρχικά, το πρόγραμμα διαβάζει τα datasets, τα οποία περιέχουν τις εικόνες και τις κατηγορίες τους και φορτώνει το αποθηκευμένο μοντέλο του autoencoder. Στη συνέχεια ζητάει από το χρήστη τιμές για τις υπερπαραμέτρους (epochs, batch size, number of neurons για το FC layer) και κατασκευάζει ένα μοντέλο συνδέοντας το output του encoder του autoencoder με ένα Fully Connected Layer και με ένα στρώμα εξόδου (softmax). Έπειτα εκπαιδεύει το μοντέλο που έχει κατασκευαστεί σε δυο στάδια. Στο πρώτο στάδιο γίνεται μάθηση μόνο των βαρών του Fully Connected Layer και στο δεύτερο στάδιο γίνεται μάθηση των βαρών του δικτύου συνολικά. Στο τέλος της εκπαίδευσης ζητάει από το χρήστη αν θέλει να επαναλάβει το πείραμα με άλλες τιμές υπερπαραμέτρων (επιλογή 1), αν θέλει να εμφανίσει τις γραφικές παραστάσεις του σφάλματος και της ορθότητας ως προς τις τιμές των υπερπαραμέτρων καθώς και πίνακες για τους υπόλοιπους δείκτες αξιολόγησης για τα εκτελεσθέντα πειράματα και (επιλογή 2) και αν θέλει να προχωρήσει στην κατηγοριοποίηση των εικόνων από το σύνολο ελέγχου(επιλογή 3).

## ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ

- Για το A: `python3 autoencoder.py -d <dataset>` , όπου:

`-d` : η επόμενη παράμετρος είναι το dataset

Αν ο χρήστης δε δώσει παραμέτρους, τότε χρησιμοποιούνται default τιμές.

Ως default τιμή ορίζεται:

`<dataset>` : train-images-idx3-ubyte

Στο τέλος κάθε πειράματος:

**Επιλογή 1:** Επανάληψη του πειράματος με διαφορετικές τιμές υπερπαραμέτρων.

**Επιλογή 2:** Γραφικές παραστάσεις

*Επιλογή 0:* Ως προς το πλήθος των layers

*Επιλογή 1:* Ως προς το μέγεθος των φίλτρων

*Επιλογή 2:* Ως προς τον αριθμό φίλτρων στο πρώτο στρώμα

*Επιλογή 3:* Ως προς τον αριθμό των εποχών

*Επιλογή 4:* Ως προς το batch size

**Επιλογή 3:** Αποθήκευση μοντέλου με filename που ζητάει από το χρήστη

- Για το B: `python3 classification.py -d <training set> -dl <training labels> -t <testset> -tl <test labels> -model <autoencoder h5>` , όπου:

`-d` : η επόμενη παράμετρος είναι training set

`-dl` : η επόμενη παράμετρος είναι training labels

`-t` : η επόμενη παράμετρος είναι testset

`-tl` : η επόμενη παράμετρος είναι test labels

`-model` : η επόμενη παράμετρος είναι autoencoder h5

Αν ο χρήστης δε δώσει παραμέτρους, τότε χρησιμοποιούνται default τιμές.

Ως default τιμές ορίζονται:

<training set>	:	train-images-idx3-ubyte
<training labels>	:	train-labels-idx1-ubyte
<testset>	:	t10k-images-idx3-ubyte
<test labels>	:	t10k-labels-idx1-ubyte
<autoencoder h5>	:	autoencoder_model

Στο τέλος κάθε πειράματος:

**Επιλογή 1:** Επανάληψη του πειράματος με διαφορετικές τιμές υπερπαραμέτρων.

**Επιλογή 2:** Γραφικές παραστάσεις

*Επιλογή 0:* Ως προς τον αριθμό των εποχών

*Επιλογή 1:* Ως προς το batch size

*Επιλογή 2:* Ως προς τον αριθμό των νευρώνων του πλήρως συνδεδεμένου στρώματος

και στη συνέχεια εμφανίζει πίνακες για τους υπόλοιπους δείκτες αξιολόγησης

**Επιλογή 3:** Κατηγοριοποίηση των εικόνων , εμφάνιση πλήθους σωστών και λανθασμένων προβλέψεων καθώς και δείγματος εικόνων αυτών.

## ΕΠΙΛΟΓΕΣ ΥΛΟΠΟΙΗΣΗΣ

Ως αριθμός των layers που δίνει ο χρήστης θεωρείται μόνο ο αριθμός των Convolutional layers και όχι ο συνολικός αριθμός των layers που θα χρησιμοποιηθούν για την εκπαίδευση του νευρωνικού δικτύου.

Στη δημιουργία του μοντέλου του autoencoder χρησιμοποιούνται 3 Pooling layers (Max Pooling στον encoder και UpSampling στον decoder αντίστοιχα) και δε χρησιμοποιείται Dropout layer, καθώς σύμφωνα με πειράματα δεν είχε κάποια ουσιαστική διαφορά στο αποτέλεσμα.

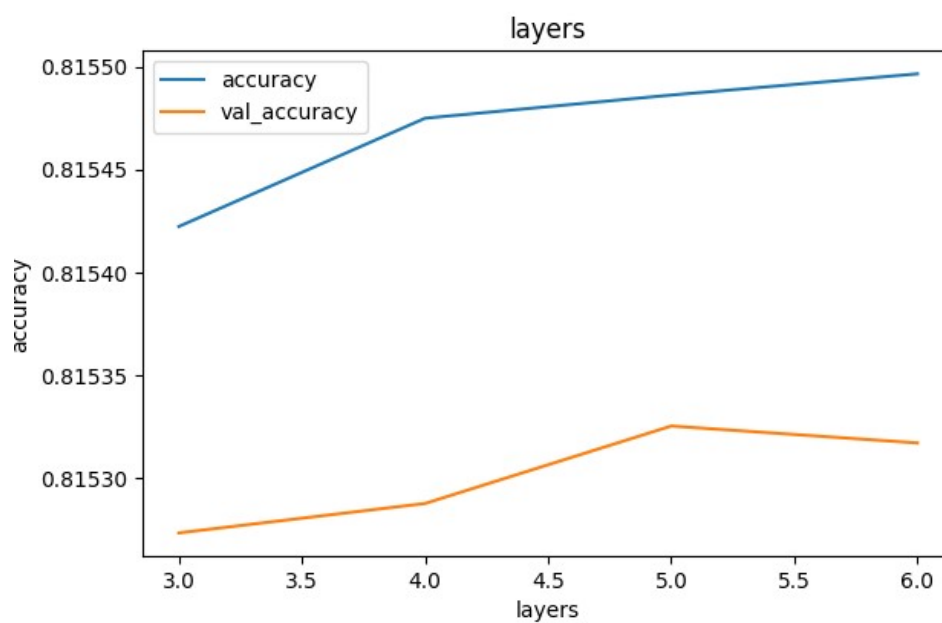
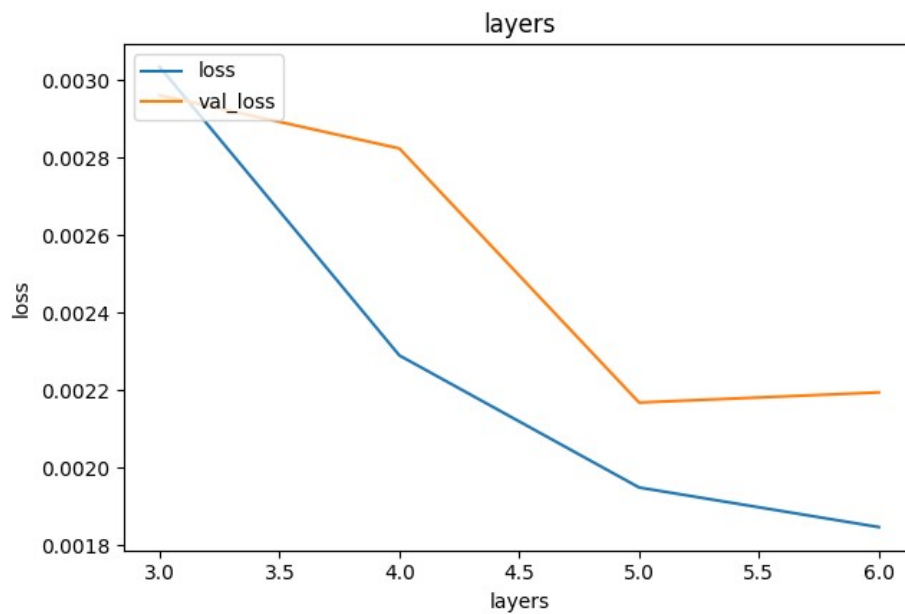
Λόγω της επιλογής του αριθμού των Pooling layers προτείνεται η χρήση φίλτρων μεγέθους 3.

## ΔΟΚΙΜΕΣ - ΠΕΙΡΑΜΑΤΑ

- AUTOENCODER

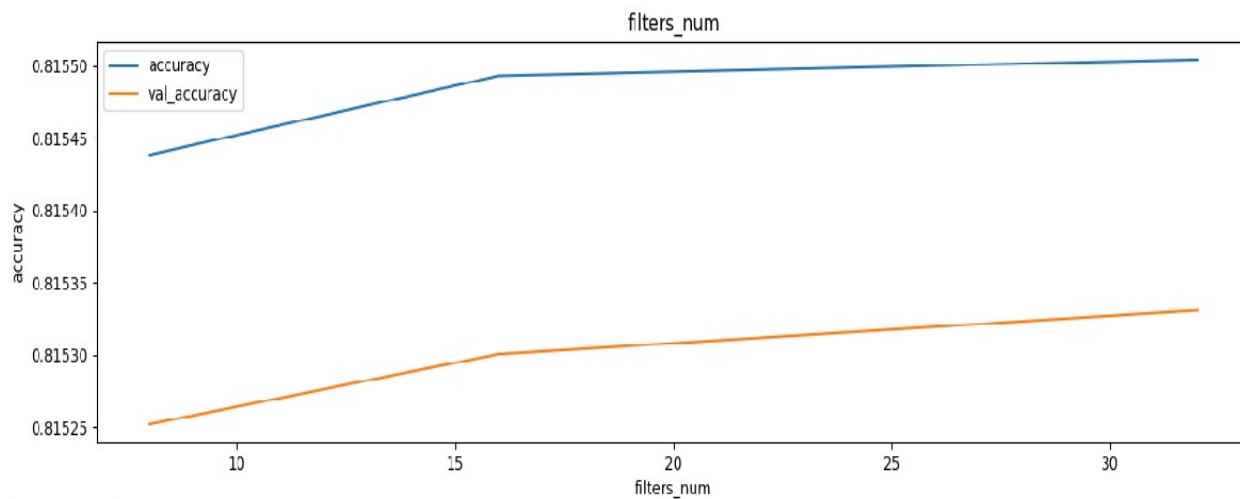
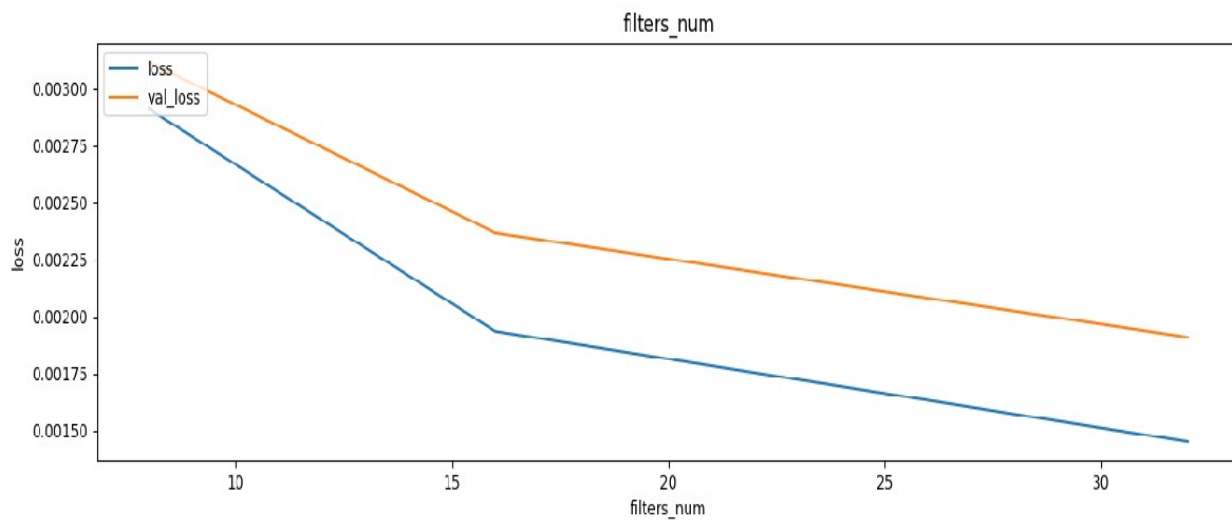
Δοκιμές για αριθμό layers:

(κρατώντας σταθερές τιμές epochs = 20, batch\_size = 64, filters\_size = 3, filters\_num = 16)



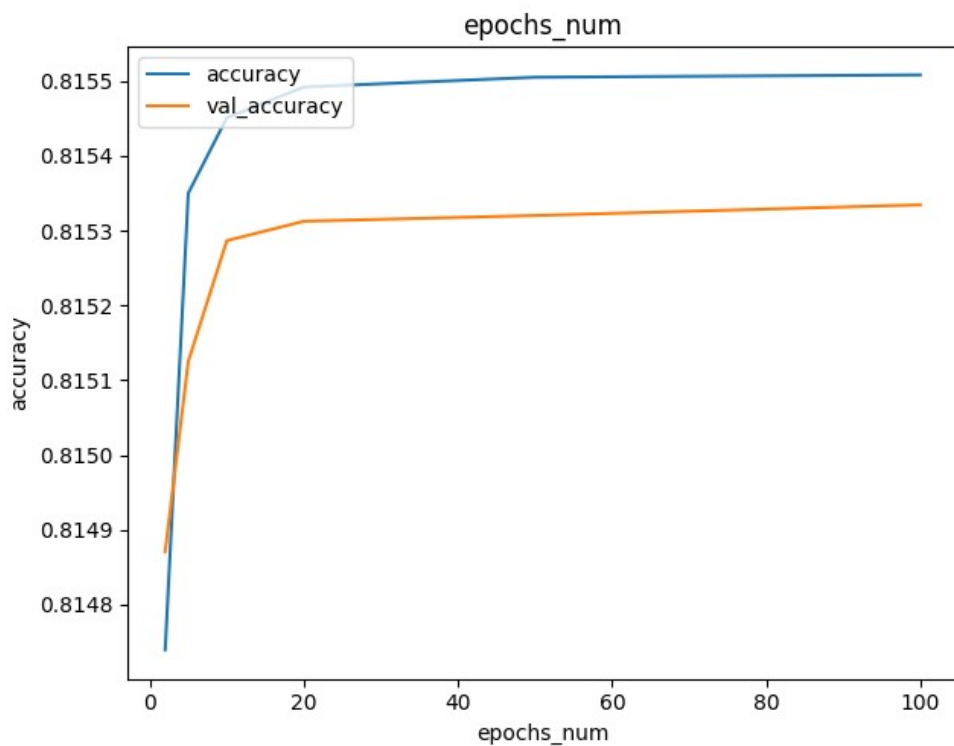
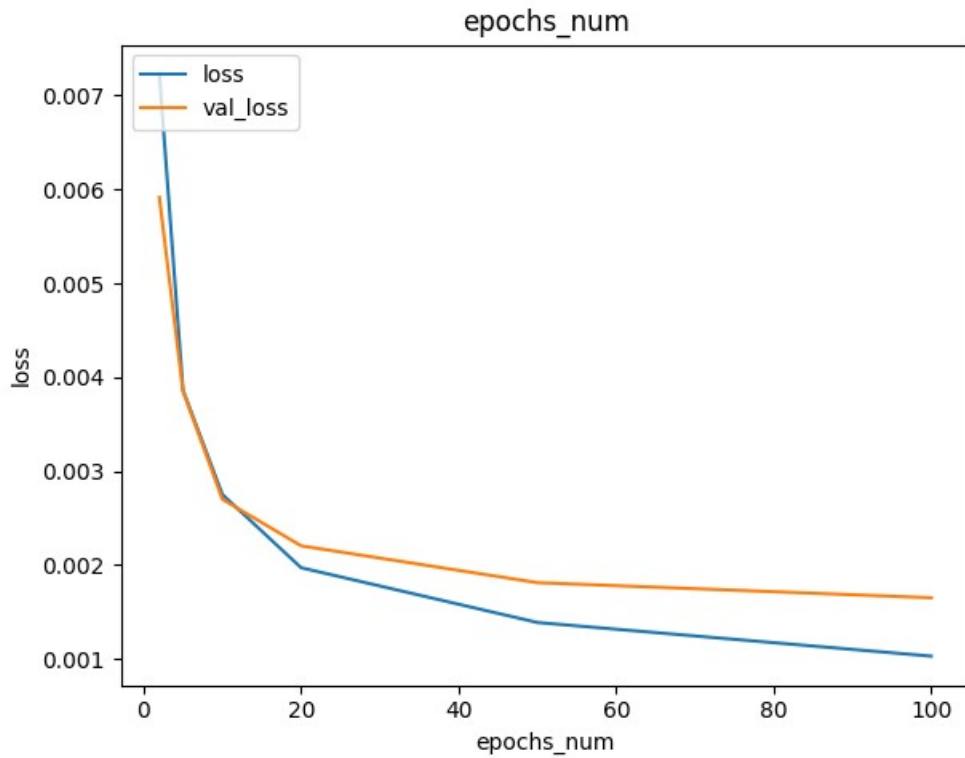
### Δοκιμές για αριθμό φίλτρων στο πρώτο layer:

(κρατώντας σταθερές τιμές layers = 5, epochs = 20, batch\_size = 64, filters\_size = 3)



### Δοκιμές για αριθμό epochs:

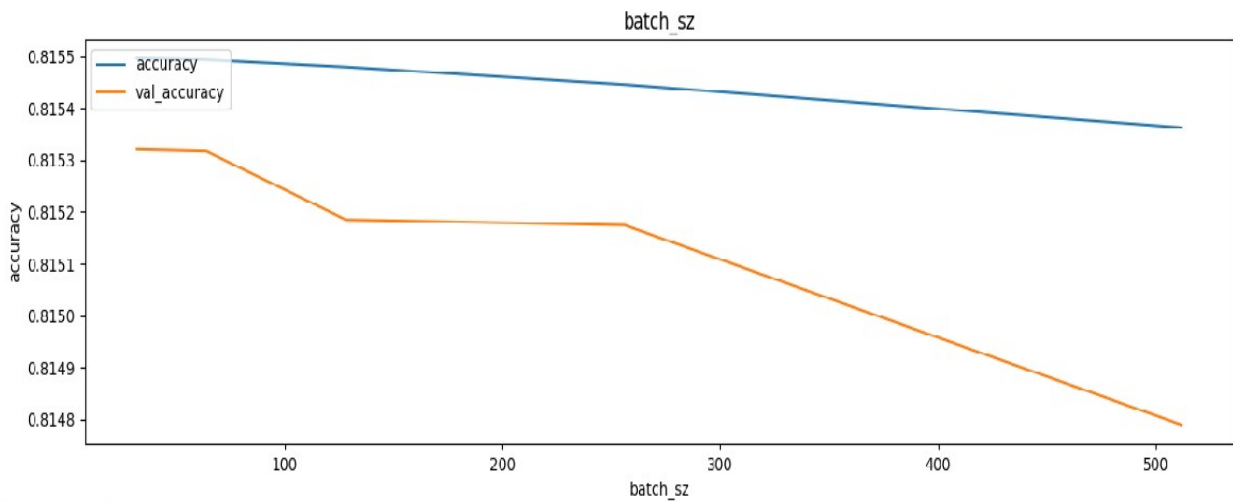
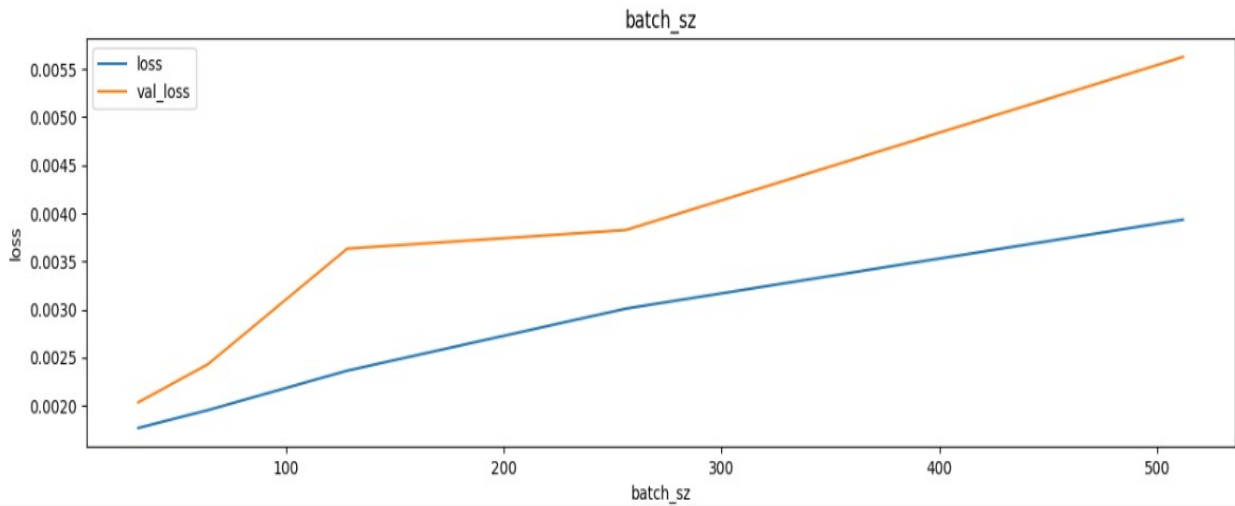
(κρατώντας σταθερές τιμές layers = 5, batch\_size = 64, filters\_size = 3, filters\_num = 16)





### Δοκιμές για batch size:

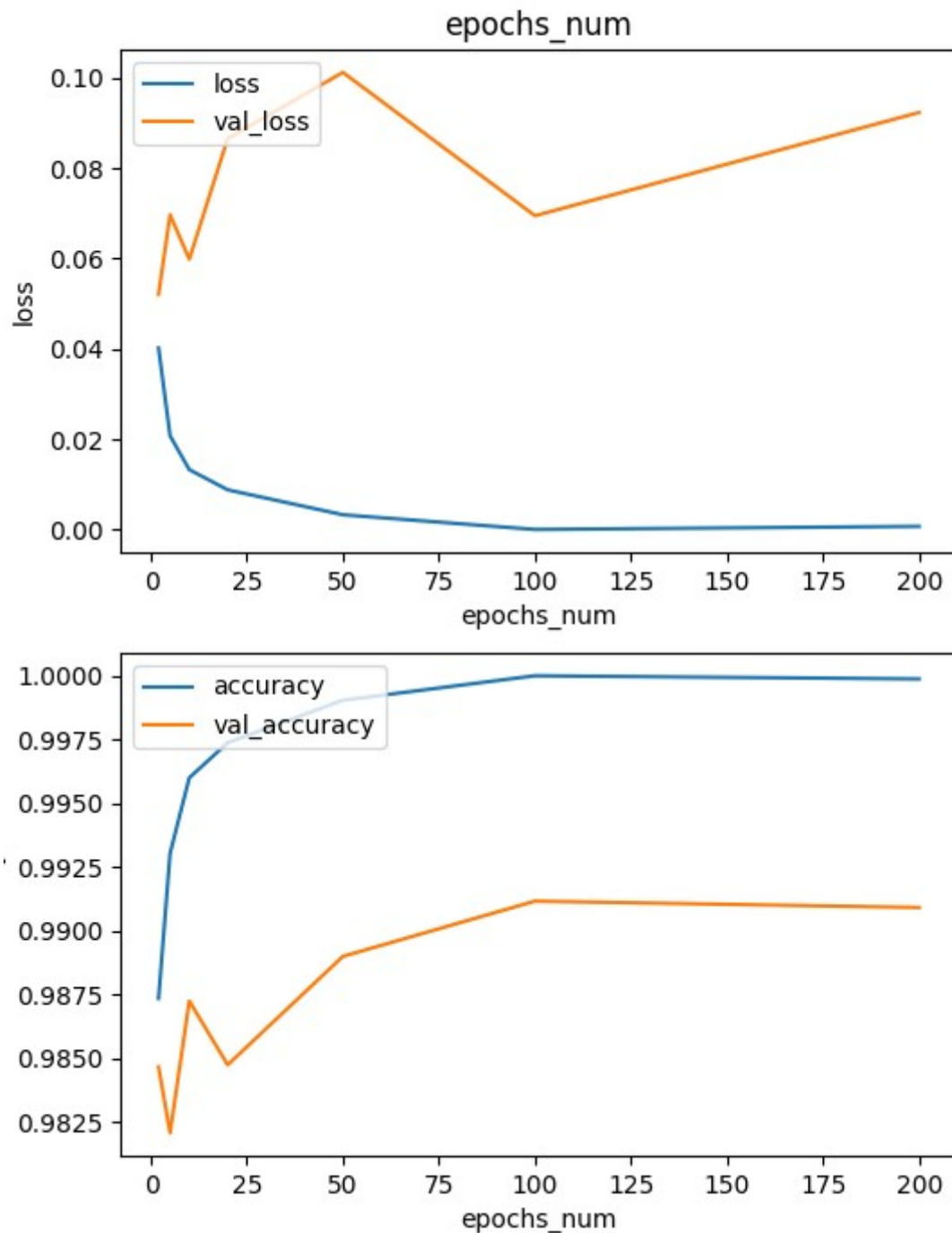
(κρατώντας σταθερές τιμές layers = 5, epochs = 20, filters\_size = 3, filters\_num = 16)



- CLASSIFICATION

Δοκιμές για αριθμό epochs:

(κρατώντας σταθερές τιμές batch\_size = 64 και neurons\_fc = 256)



Πίνακας για 10 epochs:

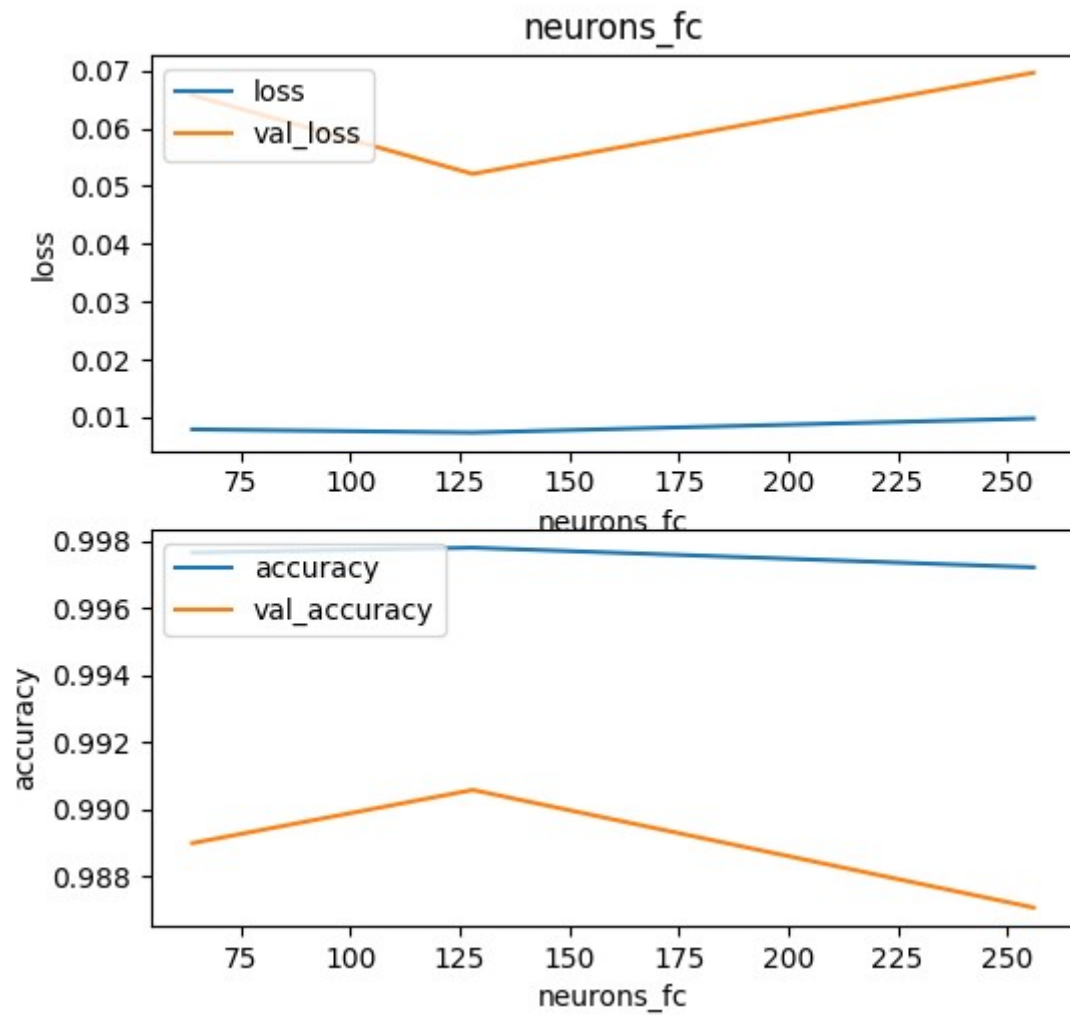
	precision	recall	f1-score	support
0	0.96	1.00	0.98	980
1	1.00	0.98	0.99	1135
2	0.99	0.99	0.99	1032
3	0.99	0.98	0.99	1010
4	0.99	0.99	0.99	982
5	0.99	0.97	0.98	892
6	0.98	0.98	0.98	958
7	0.99	0.99	0.99	1028
8	0.98	0.97	0.98	974
9	0.97	0.98	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9824				
INCORRECT:				
176				

Πίνακας για 50 epochs:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	980
1	1.00	0.99	1.00	1135
2	0.99	0.99	0.99	1032
3	0.99	1.00	0.99	1010
4	0.99	0.99	0.99	982
5	0.99	1.00	0.99	892
6	0.99	0.99	0.99	958
7	1.00	0.99	0.99	1028
8	0.99	0.99	0.99	974
9	0.99	0.99	0.99	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9932				
INCORRECT:				
68				

Δοκιμές για αριθμό neurons στο FC layer:

(κρατώντας σταθερές τιμές epochs = 20 και batch\_size = 64)



Πίνακας για 64 FC layers:

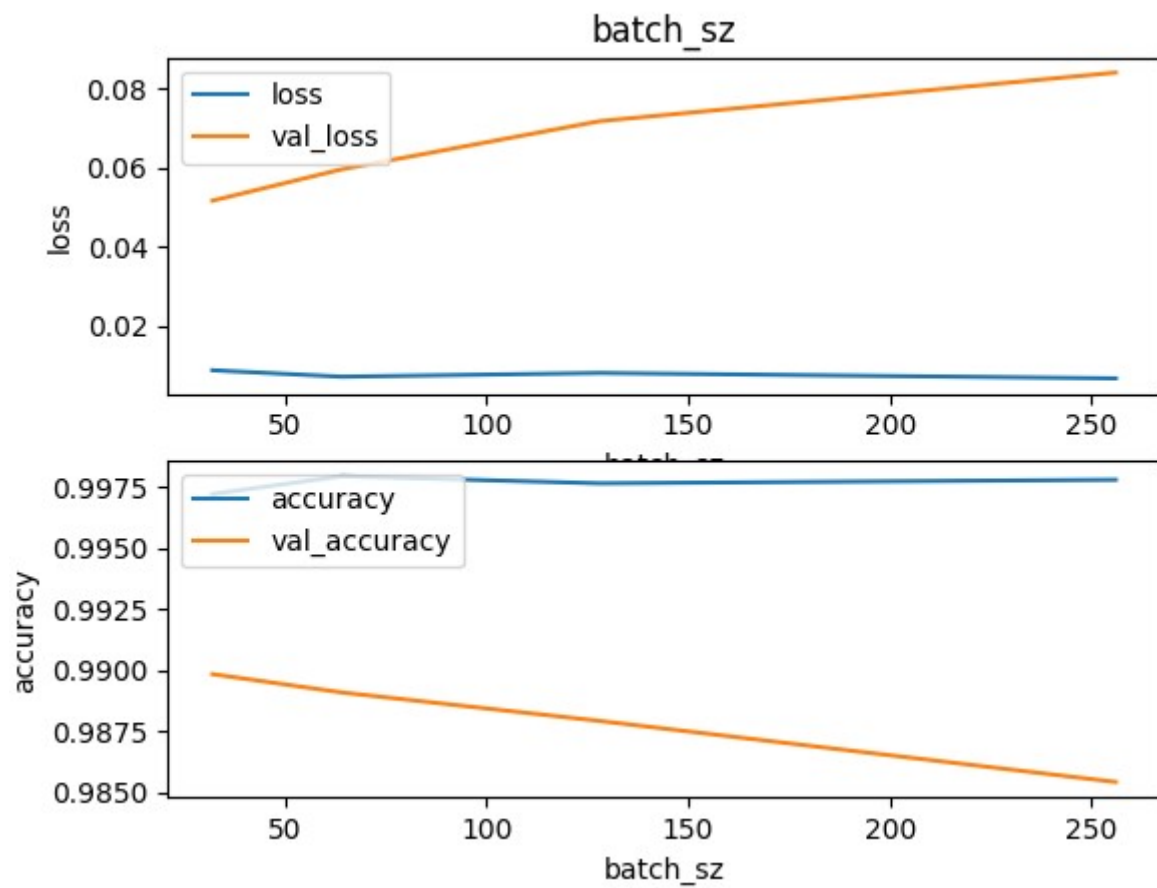
	precision	recall	f1-score	support
0	0.99	1.00	0.99	980
1	0.99	0.99	0.99	1135
2	0.97	0.99	0.98	1032
3	0.99	0.98	0.99	1010
4	0.99	0.99	0.99	982
5	0.99	0.99	0.99	892
6	0.99	0.99	0.99	958
7	1.00	0.97	0.98	1028
8	0.98	0.99	0.99	974
9	0.98	0.98	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9873				
INCORRECT:				
127				

Πίνακας για 256 FC layers:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	980
1	1.00	0.99	1.00	1135
2	0.99	0.99	0.99	1032
3	1.00	1.00	1.00	1010
4	0.99	0.99	0.99	982
5	0.99	0.99	0.99	892
6	0.99	0.99	0.99	958
7	0.99	0.99	0.99	1028
8	0.99	1.00	0.99	974
9	0.99	0.98	0.99	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9920				
INCORRECT:				
80				

Δοκιμές για batch size:

(κρατώντας σταθερές τιμές epochs = 20 και neurons\_fc = 256)



Πίνακας για batch\_size 32

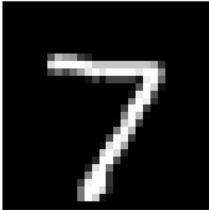
	precision	recall	f1-score	support
0	0.99	1.00	0.99	980
1	0.99	1.00	0.99	1135
2	0.99	0.99	0.99	1032
3	0.99	0.99	0.99	1010
4	0.99	0.99	0.99	982
5	0.99	0.98	0.99	892
6	1.00	0.99	0.99	958
7	0.99	0.99	0.99	1028
8	0.99	0.99	0.99	974
9	0.98	0.98	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9904				
INCORRECT:				
96				

Πίνακας για batch\_size 256:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	980
1	1.00	0.99	1.00	1135
2	0.99	0.99	0.99	1032
3	0.97	1.00	0.98	1010
4	1.00	0.99	0.99	982
5	1.00	0.98	0.99	892
6	1.00	0.99	0.99	958
7	0.99	0.98	0.99	1028
8	0.98	0.99	0.98	974
9	0.98	0.99	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000
TO REPEAT THE EXPERIMENT PRESS 1.				
TO CLASSIFY IMAGES PRESS 3.				
3				
CORRECT:				
9883				
INCORRECT:				
117				

## Correct found (for 200 epochs):

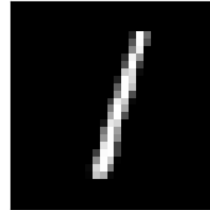
Predicted 7, Class 7



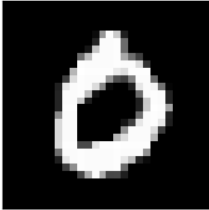
Predicted 2, Class 2



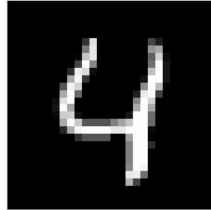
Predicted 1, Class 1



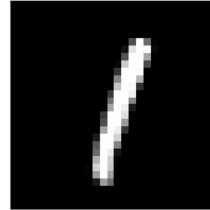
Predicted 0, Class 0



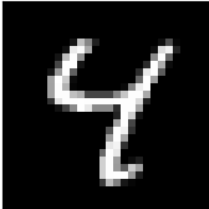
Predicted 4, Class 4



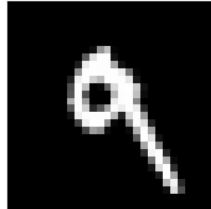
Predicted 1, Class 1



Predicted 4, Class 4



Predicted 9, Class 9

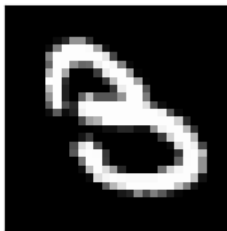


Predicted 5, Class 5

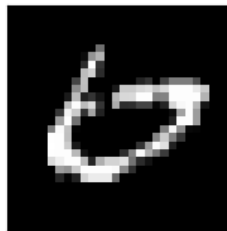


## Incorrect found (for 200 epochs):

Predicted 5, Class 3



Predicted 5, Class 6



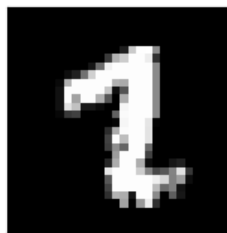
Predicted 5, Class 9



Predicted 5, Class 6



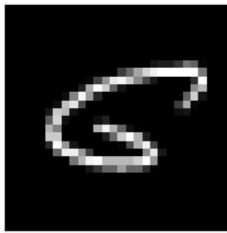
Predicted 5, Class 2



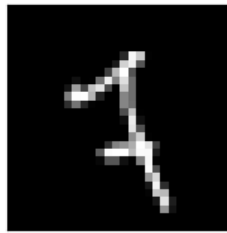
Predicted 5, Class 8



Predicted 5, Class 6



Predicted 5, Class 7



Predicted 5, Class 9

