

# Αναφορά Ασκήσεων Επεξεργασίας Εικόνας

Μέρος Β

Κωνσταντίνος Κοτορένης 1053750



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

# Περιεχόμενα

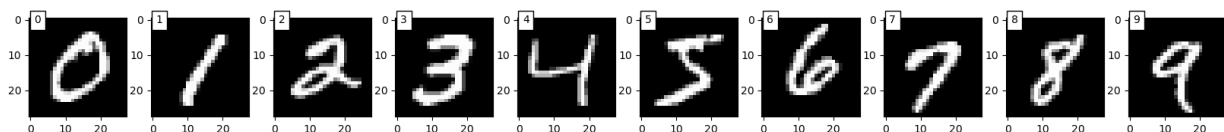
Περιεχόμενα	2
Μέρος Α: Κατηγοριοποίηση Εικόνων με χρήση Συνελικτικών Νευρωνικών Δικτύων (Convolutional Neural Networks - CNN)	3
Μέρος Β: Κατηγοριοποίηση Εικόνων με χρήση Histogram of Oriented Gradients και Support Vector Machines	8
Κώδικας	9

# Μέρος Α: Κατηγοριοποίηση Εικόνων με χρήση Συνελικτικών Νευρωνικών Δικτύων (Convolutional Neural Networks - CNN)

1. Αρχικά, κατεβάστε το MNIST από τον παρακάτω σύνδεσμο <http://yann.lecun.com/exdb/mnist/> και απεικονίστε σε ένα figure μία εικόνα από κάθε κλάση. Οι εικόνες είναι χωρισμένες σε δεδομένα εκπαίδευσης (training data) και δεδομένα ελέγχου (testing data).
2. Υλοποιήστε την επαναληπτική διαδικασία του αλγορίθμου Stochastic Gradient Descent για την εκπαίδευση ενός NN το οποίο ελαχιστοποιεί την συνάρτηση κόστους (2). Πιο συγκεκριμένα, θα πρέπει να υλοποιήσετε τα παρακάτω βήματα:
  - a. Χωρίστε τα δεδομένα εκπαίδευσης σε μη επικαλυπτόμενα υποσύνολα (mini-batches) μεγέθους  $b$ .
  - b. Χρησιμοποιώντας σε κάθε επανάληψη ένα διαφορετικό mini-batch υπολογίστε μία (στοχαστική) προσέγγιση για το gradient  $\nabla W_t F(X; W_t)$
  - c. Χρησιμοποιώντας το gradient που υπολογίσατε στο προηγούμενο βήμα ανανεώστε τις παραμέτρους του NN σύμφωνα με την εξίσωση του SGD.

Τα παραπάνω βήματα (α)-(c) επαναλαμβάνονται μέχρι να συγκλίνει ο SGD σε κάποιο τοπικό ελάχιστο της συνάρτησης (2). Ακόμη, προκειμένου να εξασφαλιστεί η σύγκλιση του SGD είναι χρήσιμο ο SGD να επεξεργαστεί πολλές φορές το σύνολο δεδομένων. Το κάθε «πέρασμα» των δεδομένων σηματοδοτεί μία εποχή (epoch). Συνεπώς, θα είναι χρήσιμο να χρησιμοποιήσετε αρκετές εποχές προκειμένου να εξασφαλιστεί η σύγκλιση του SGD.

3. Χρησιμοποιώντας το σύνολο δεδομένων ελέγχου (testing dataset) του MNIST στο τέλος κάθε εποχής υπολογίστε το σφάλμα κατηγοριοποίησης (loss) και την ακρίβεια (accuracy) του μοντέλου σας και απεικονίστε τα σε δύο διαφορετικές γραφικές παραστάσεις.
4. Υπολογίστε (μέσω συνάρτησης που θα κατασκευάσετε εσείς) τον confusion matrix με βάση τα αποτελέσματα που εξάγατε.



Εικόνα Α1: [α-ι] Παρουσιάζονται ένα παραδείγματα κάθε κλάσης των χειρόγραφων αριθμών από το σύνολο δεδομένων για την εκπαίδευση του νευρωνικού δικτύου.

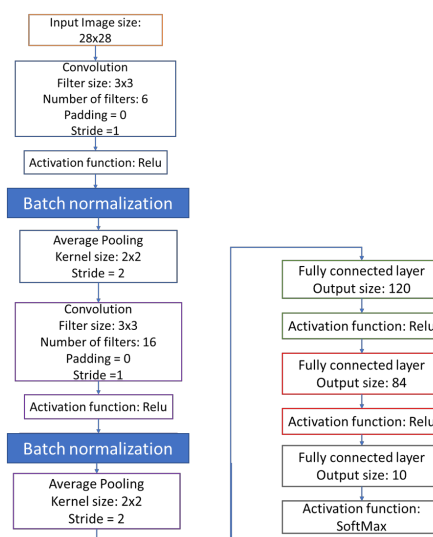
Για το πρώτο ερώτημα κατέβασα τα δεδομένα από την ιστοσελίδα και παρουσιάζω μία εικόνα από κάθε κλάση στην παραπάνω εικόνα (Εικόνα A1). Για να το κάνω αυτό έψαξα με τα labels τους αριθμούς από 0 μέχρι 9 και μετά τους ταξινόμησα σε αύξουσα σειρά.

Για τα επόμενα ερωτήματα κατάλαβα ότι μπορώ να προσαρμόσω μέσα στον κώδικα μου αν δεν υπάρχει το σύνολο των δεδομένων να το κατεβάζει πριν το τρέξει με την παρακάτω εντολή (κώδικας A1):

```
train_dataset = datasets.MNIST(root='./data', train=True,
transform=transform, download=True)
test_dataset = datasets.MNIST(root='./data', train=False,
transform=transform, download=True)
```

Κώδικας A1 : Εντολές για να φορτωθούν τα σύνολα δεδομένων για εκπαίδευση και έλεγχο του νευρωνικού δικτύου. Αν μπει το "download=True" τότε κατεβάζει τα δεδομένα αν δεν υπάρχουν.

Σύμφωνα με την εκφώνηση η δομή του νευρωνικού δικτύου φαίνεται στην εικόνα A2. Για την υλοποίηση του νευρωνικού δικτύου χρησιμοποιήθηκε η γλώσσα προγραμματισμού python και η βιβλιοθήκη pytorch.



Εικόνα A2: Αρχιτεκτονική του Συνελικτικού Νευρωνικού Δικτύου

Για την εκπαίδευση του νευρωνικού δικτύου δημιούργησα 10 εποχές με ρυθμό εκμάθησης 0.001 και batch size να είναι 64. Για να χρησιμοποιήσω διαφορετικό mini-batch κάθε φορά χρησιμοποιώ την εντολή shuffle όταν καλώ το train\_loader (κώδικας A2)

```
train_loader = torch.utils.data.DataLoader(train_dataset,
batch_size=batch_size, shuffle=True)
```

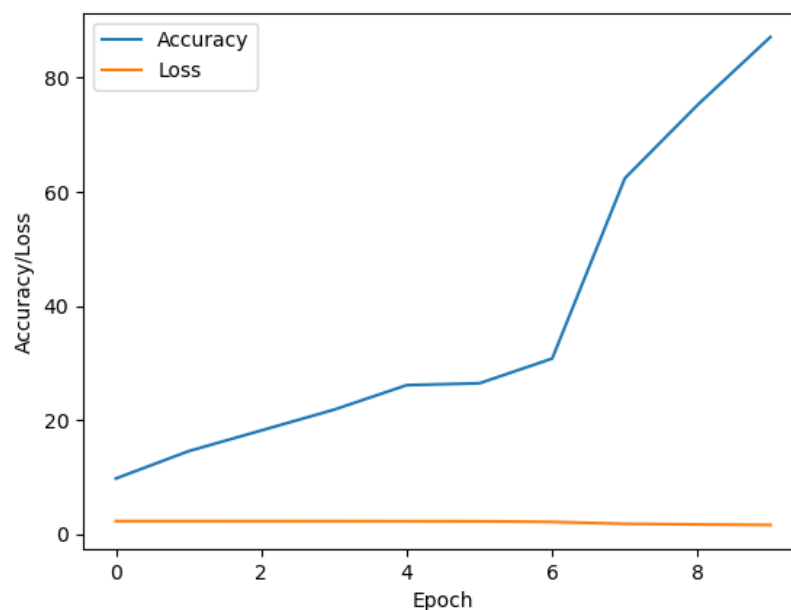
Κώδικας A2 : Εντολή δημιουργίας batch

Παρακάτω βλέπουμε την έξοδο του προγράμματος στο τερματικό (Εικόνα A2). Παρατηρούμε ότι με το πέρασμα κάθε εποχής βελτιώνεται πολύ η ακρίβεια και στη 10η εποχή φτάνει το 87.14%. Αντίστοιχα το σφάλμα κατηγοριοποίησης όλο και μειώνεται, κάτι που είναι αναμενόμενο εφόσον συγκλίνει το νευρωνικό. Τα νούμερα αυτά παρουσιάζονται και στη γραφική παράσταση της εικόνας A3.

```
Epoch [1/10]: test accuracy of the model on the test images: 9.77 %
Epoch [1/10]: loss of the model: 2.3006
Epoch [2/10]: test accuracy of the model on the test images: 14.56 %
Epoch [2/10]: loss of the model: 2.2990
Epoch [3/10]: test accuracy of the model on the test images: 18.19 %
Epoch [3/10]: loss of the model: 2.2966
Epoch [4/10]: test accuracy of the model on the test images: 21.83 %
Epoch [4/10]: loss of the model: 2.2933
Epoch [5/10]: test accuracy of the model on the test images: 26.13 %
Epoch [5/10]: loss of the model: 2.2872
Epoch [6/10]: test accuracy of the model on the test images: 26.46 %
Epoch [6/10]: loss of the model: 2.2691
Epoch [7/10]: test accuracy of the model on the test images: 30.79 %
Epoch [7/10]: loss of the model: 2.1635
Epoch [8/10]: test accuracy of the model on the test images: 62.4 %
Epoch [8/10]: loss of the model: 1.8272
Epoch [9/10]: test accuracy of the model on the test images: 75.25 %
Epoch [9/10]: loss of the model: 1.7115
Epoch [10/10]: test accuracy of the model on the test images: 87.14 %
Epoch [10/10]: loss of the model: 1.6325
```

	0	1	2	3	4	5	6	7	8	9
0	935	2	0	2	3	16	11	2	8	1
1	0	1115	2	5	0	1	4	0	8	0
2	18	7	911	11	24	7	13	22	15	4
3	1	10	30	876	1	40	5	14	21	12
4	2	3	5	0	862	5	37	5	7	56
5	10	13	10	70	32	712	18	3	18	6
6	13	4	4	0	21	33	876	5	2	0
7	3	22	33	2	11	4	0	871	10	72
8	28	6	19	62	17	65	11	11	717	38
9	13	9	6	10	62	19	2	32	17	839

Εικόνα A2: Ακρίβεια και σφάλμα κατηγοριοποίησης ανά εποχή και ο Confusion Matrix (10 εποχές).



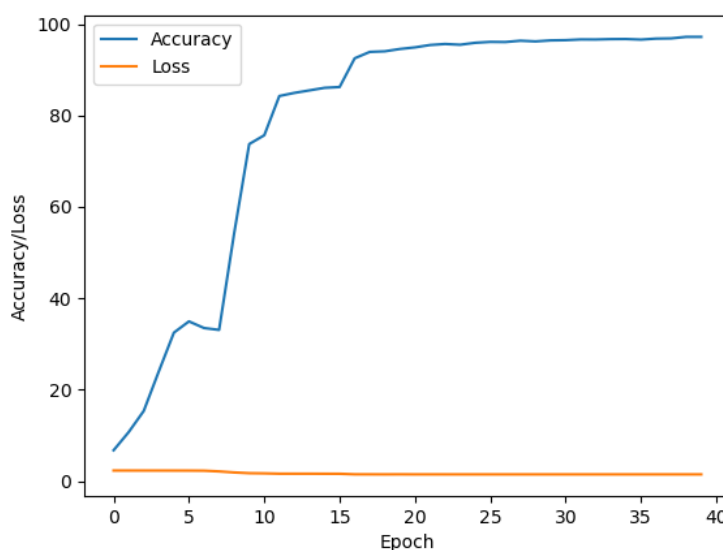
Εικόνα A3: Γραφική παράσταση ακρίβειας και σφάλματος κατηγοριοποίησης ως προς κάθε εποχή (10 εποχές)

Στην εικόνα A2 και στον πίνακα A1 παρουσιάζεται και ο Confusion Matrix μετά την εκπαίδευση του νευρωνικού όπου σαν είσοδο έχουμε εικόνες από το training set. Οι γραμμές είναι η είσοδος ενώ οι στήλες δηλώνουν την κλάση που κατηγοριοποιήθηκε η είσοδος. Για παράδειγμα στη γραμμή με είσοδο "4" παρατηρούμε ότι 862 από τις εισόδους ταξινομήθηκαν σωστά σαν "4" και δεξιά αριστερά βλέπουμε που αλλού ταξινομήθηκαν οι εισοδοί "4". Στον Confusion Matrix θέλουμε η διαγώνιος να έχει τη μεγαλύτερη τιμή και τα υπόλοιπα κελιά να έχουν όσο χαμηλότερη τιμή γίνεται έτσι ώστε η κατηγοριοποίηση που γίνεται να είναι επιτυχής.

	0	1	2	3	4	5	6	7	8	9
0	935	2	0	2	3	16	11	2	8	1
1	0	1115	2	5	0	1	4	0	8	0
2	18	7	911	11	24	7	13	22	15	4
3	1	10	30	876	1	40	5	14	21	12
4	2	3	5	0	862	5	37	5	7	56
5	10	13	10	70	32	712	18	3	18	6
6	13	4	4	0	21	33	876	5	2	0
7	3	22	33	2	11	4	0	871	10	72
8	28	6	19	62	17	65	11	11	717	38
9	13	9	6	10	62	19	2	32	17	839

Πίνακας A1: Confusion Matrix

Δοκίμασα να αυξήσω τον αριθμό των εποχών, το έκανα 40, και μετά την εκπαίδευση κατάφερε να φτάσει 97,21% ακρίβεια. Παρατηρώ ότι σχεδόν κοντά στην εποχή 20 το νευρωνικό ήδη ξεκινάει να συγκλίνει σε ένα καλό ποσοστό ακρίβειας (εικόνα A4).



Εικόνα A4: Γραφική παράσταση ακρίβειας και σφάλματος κατηγοριοποίησης ως προς κάθε εποχή (40 εποχές)

Παρακάτω φαίνεται δείγμα της εξόδου στο τερματικό μαζί με τον Confusion Matrix για τις 40 εποχές (εικόνα A5). Ο Confusion Matrix αυτός φαίνεται ότι έχει πολύ λιγότερα λάθη σε σχέση με αυτόν των 10 εποχών και κοντά στα 1000 δείγματα κατηγοριοποιούνται στην σωστή κλάση ενώ στον προηγούμενο ήταν γύρω στα 800 δείγματα.

```
Epoch [1/40]: test accuracy of the model on the test images: 6.74 %
Epoch [1/40]: loss of the model: 2.3015
Epoch [2/40]: test accuracy of the model on the test images: 10.71 %
Epoch [2/40]: loss of the model: 2.2996
Epoch [3/40]: test accuracy of the model on the test images: 15.35 %
Epoch [3/40]: loss of the model: 2.2973
Epoch [4/40]: test accuracy of the model on the test images: 24.01 %
Epoch [4/40]: loss of the model: 2.2946
Epoch [5/40]: test accuracy of the model on the test images: 32.48 %
Epoch [5/40]: loss of the model: 2.2908
Epoch [6/40]: test accuracy of the model on the test images: 34.94 %
Epoch [6/40]: loss of the model: 2.2843
Epoch [7/40]: test accuracy of the model on the test images: 33.49 %
Epoch [7/40]: loss of the model: 2.2652
Epoch [8/40]: test accuracy of the model on the test images: 33.08 %
Epoch [8/40]: loss of the model: 2.1090
Epoch [9/40]: test accuracy of the model on the test images: 54.21 %
Epoch [9/40]: loss of the model: 1.8802
Epoch [10/40]: test accuracy of the model on the test images: 73.75 %
Epoch [10/40]: loss of the model: 1.7215

Epoch [36/40]: test accuracy of the model on the test images: 96.62 %
Epoch [36/40]: loss of the model: 1.4612
Epoch [37/40]: test accuracy of the model on the test images: 96.83 %
Epoch [37/40]: loss of the model: 1.4612
Epoch [38/40]: test accuracy of the model on the test images: 96.88 %
Epoch [38/40]: loss of the model: 1.4612
Epoch [39/40]: test accuracy of the model on the test images: 97.21 %
Epoch [39/40]: loss of the model: 1.4612
Epoch [40/40]: test accuracy of the model on the test images: 97.21 %
Epoch [40/40]: loss of the model: 1.4612
```

	0	1	2	3	4	5	6	7	8	9
973	0	0	0	0	2	2	1	2	0	
0	1118	3	2	1	0	4	1	6	0	
3	0	1009	1	2	0	1	9	7	0	
2	1	6	969	0	12	0	9	8	3	
2	0	3	0	948	1	5	1	3	19	
4	0	0	3	0	870	4	3	6	2	
8	2	1	0	5	7	934	0	1	0	
1	5	12	2	0	0	0	999	2	7	
6	1	4	3	3	5	5	5	940	2	
4	5	2	8	9	4	1	11	4	961	

Εικόνα A5: Ακρίβεια και σφάλμα κατηγοριοποίησης ανά εποχή και ο Confusion Matrix (40 εποχές).

## Μέρος Β: Κατηγοριοποίηση Εικόνων με χρήση Histogram of Oriented Gradients και Support Vector Machines

1. Υπολογίστε τα χαρακτηριστικά HOG του συνόλου δεδομένων εκπαίδευσης με έτοιμη συνάρτηση της MATLAB. Στη συνέχεια, εκπαιδεύστε ένα SVM με τα χαρακτηριστικά αυτά.
2. Με την χρήση του εκπαιδευμένου SVM, παρουσιάστε ενδεικτικά αποτελέσματα κατηγοριοποίησης για το σύνολο των δεδομένων ελέγχου. Πειραματιστείτε με διαφορετικό μέγεθος patch για τα HOGs, τόσο κατά την εκπαίδευση όσο και κατά τον έλεγχο.
3. Υπολογίστε (μέσω συνάρτησης που θα κατασκευάσετε εσείς) τον confusion matrix με βάση τα αποτελέσματα που εξάγατε. Συγκρίνεται τα αποτελέσματα με τα αποτελέσματα που προέκυψαν με την χρήση του Συνελικτικού δικτύου από το Α μέρος.
4. Bonus: Υπολογίστε τα HOG χωρίς την έτοιμη συνάρτηση της MATLAB

Για την υλοποίηση της άσκησης χρησιμοποίησα τη python και την έτοιμη εντολή της python για το Histogram of Oriented Gradients (κώδικας B1). Επίσης για να φορτώσω τα δεδομένα χρησιμοποίησα τον τρόπο που ακολούθησα στην άσκηση A1 γιατί ο άλλος τρόπος απαιτούσε τη pytorch κάτι που δεν χρειάζομαι τώρα.

```
hog_features_train = []
for image in features_train:
    hog_features_train.append(hog(image.reshape((28, 28)),
    orientations=9, pixels_per_cell=(14, 14), cells_per_block=(1, 1),
    visualize=False))
```

Κώδικας B1: Συνάρτηση HOG της python μέσα σε επανάληψη για τον υπολογισμό των χαρακτηριστικών όλων των εικόνων του training set.

Για τα SVM υπάρχει μία εντολή για έτοιμα classification report (κώδικας B2) τα οποία δείχνουν την ακρίβεια για κάθε κλάση. Παρατηρούμε ότι η ακρίβεια (precision) είναι πάνω από 80% σε κάθε κλάση και η συνολική ακρίβεια (accuracy) είναι 89% (εικόνα B1). Το report επίσης μας δίνει το recall, που είναι τα σωστά/(σωστά + false negatives), το f1-score, που είναι ο μέσος της ακρίβειας και του recall, και το support, που είναι τα δείγματα που ήταν σωστά (true positives and true negatives)

```
metrics.classification_report(labels_test, predicted_labels)
```

Κώδικας B2: Εντολή classification\_report



Classification Report:									
	precision	recall	f1-score	support					
0	0.87	0.94	0.90	980					
1	0.98	0.96	0.97	1135					
2	0.86	0.91	0.88	1032					
3	0.83	0.87	0.85	1010					
4	0.88	0.86	0.87	982					
5	0.91	0.88	0.90	892					
6	0.89	0.90	0.89	958					
7	0.93	0.88	0.90	1028					
8	0.88	0.86	0.87	974					
9	0.82	0.78	0.80	1009					
accuracy			0.89	10000					
macro avg	0.88	0.88	0.88	10000					
weighted avg	0.89	0.89	0.89	10000					
Confusion Matrix:									
0	1	2	3	4	5	6	7	8	9
920	2	3	1	2	0	5	6	5	36
8	1089	3	1	15	0	8	5	2	4
8	1	934	52	17	1	2	9	4	4
3	0	56	883	8	12	1	5	15	27
16	5	20	12	845	3	34	7	14	26
1	1	3	38	5	789	15	0	26	14
29	4	1	2	20	18	862	0	14	8
3	5	50	14	8	1	0	904	9	34
9	1	8	20	22	21	32	9	836	16
57	6	5	43	21	18	14	28	27	790

Εικόνα B1: Classification Report και Confusion Matrix

Τέλος παρατηρώ ότι ο Confusion Matrix αυτής της κατηγοριοποίησης είναι πιο κοντά σε αυτόν των 10 εποχών του CNN παρά σε αυτόν των 40 εποχών. Γενικά δείγματα που έχει κατηγοριοποιήσει σωστά είναι γύρω στα 800 με 900. Επίσης οι 10 εποχές είχαν παρόμοια ακρίβεια με το συγκεκριμένο SVM.

## Κώδικας

Όλα τα ερωτήματα έγιναν με τη γλώσσα python και ο κώδικας είναι αναρτημένος στο παρακάτω σύνδεσμο στο GitHub:

<https://github.com/kostaskoto/Digital-Image-Processing-Neural-Network>