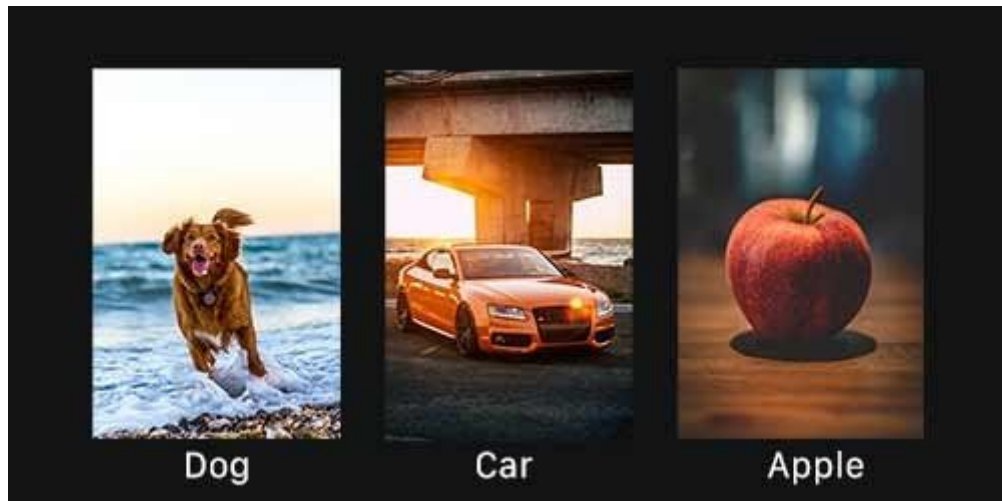


**Πανεπιστήμιο δυτικής Αττικής
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Πληροφορικής**



**Αναγνώριση προτύπων & Μηχανική Μάθηση
Εργασία 3
Σφυριδάκη Αγγελική
cs151036
cs151036@uniwa.gr**

CNN - 1ο Μέρος	2
Σκοπός εργασίας	2
Επεξήγηση κώδικα	2
Αποτελέσματα	3
Απόδοση	3
Αποτελέσματα με εικόνες	4
LSTM - Μέρος 2	5
Σκοπός εργασίας	5
Επεξήγηση κώδικα	5
Συναρτήσεις	6
Generate_data:	6
Random_sum_pairs:	6
Random_sub_pairs:	6
Sum_to_string:	6
Sub_to_string:	6
Integer_encode:	7
One_hot_encode:	7
Read_data:	7
Invert:	7
insert:	8
Αποτελέσματα-Εισόδη χρήστη:	8
Απόδοση	8

CNN - 1ο Μέρος

Σκοπός εργασίας

Στο πρώτο μέρος της εργασίας αναπτύσσουμε ένα δίκτυο CNN με το οποίο προσπαθούμε να κατηγοριοποιήσουμε τα δεδομένα εισόδου σε δέκα κατηγορίες. Ο κώδικας χωρίζεται σε δύο αρχεία. Στο πρώτο διαβάζουμε τα δεδομένα και εκπαιδεύουμε το μοντέλο ενώ στον δεύτερο ελέγχουμε την απόδοσή του.

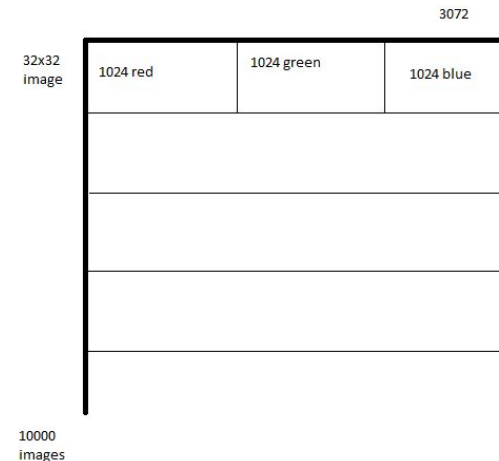
Επεξήγηση κώδικα

Διαβάζουμε όλες τις υπάρχουσες κλάσεις από το αρχείο `batches.meta` και στη συνέχεια τα δεδομένα από το πρώτο `data_batch`. Αρχικά, είναι μορφοποιημένα ως ένα dictionary όπου οι εγγραφές `data` είναι τα δεδομένα ενώ οι αντιστοιχίες τους στα `labels`.

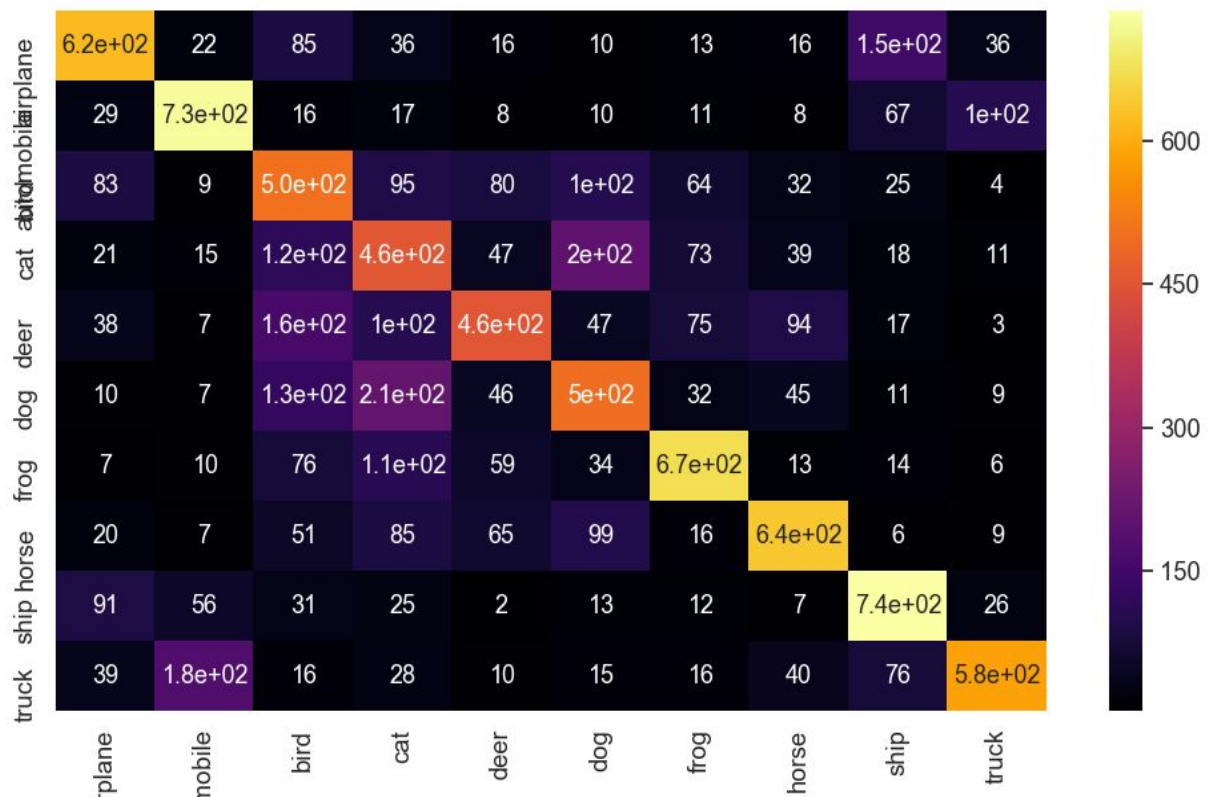
Περιέχει 10000 εικόνες, μια λίστα για κάθε εικόνα. Κάθε εικόνα αποτελείται από 3072 bytes: 1024 κόκκινα, 1024 πράσινα και 1024 μπλε στη σειρά. Για την σύνθεση κάθε pixel πρέπει να πάρουμε τα αντίστοιχα 3 χρώματα (κόκκινο, πράσινο, μπλε) οπότε με την εντολή `np.reshape(np.transpose(np.reshape(i, (3, 1024))), (32, 32, 3))` μετατρέπουμε τα 3072 bytes σε 3 των 1024 ώστε να απεικονίσουμε τα χρώματα σε τρεις λίστες. Στη συνέχεια ενωπιοούμε τα τρία αντίστοιχα bytes ώστε να προκύψουν τα pixels. Τέλος χωρίζουμε την εικόνα σε $32 \times 32 \times 3$ ώστε να αποκτήσουμε ακριβώς την εικόνα στις διαστάσεις και την μορφή που την επιθυμούμε.

Ακολούθως χωρίζουμε το σύνολο σε `train` και `validation` τεστ έτσι ώστε το `validation set` να αποτελείται από το 3%. Παρακάτω, χωρίζουμε το σύνολο των εικόνων ανά κατηγορία και εμφανίζουμε τυχαία 4 από κάθε είδος. Επιτέλους καταλλήλουμε στην εκπαίδευση του μοντέλου, όπου αρχικά καθορίζουμε την αρχιτεκτονική του δημιουργώντας 2 στρώματα νευρώνων. Προσθέτουμε την δυνατότητα στο δίκτυο να αναγνωρίζει τα μοτίβα ασχέτως μεγέθους και το εκπαιδεύουμε για 40 εποχές. Μετά το πέρας της εκπαίδευσης αποθηκεύουμε το μοντέλο ως `CNN_1.h5`.

Στο αρχείο `TestTheModel` φορτώνουμε το `test_batch`, το μορφοποιούμε ακριβώς όπως στο προηγούμενο κώδικα και το δίνουμε σαν είσοδο στο CNN μοντέλο, το οποίο επίσης φορτώνουμε. Το μοντέλο προβλέπει την κατηγορία κάθε εικόνας και ακολούθως υπολογίζει όλα τα `metric scores` (`accuracy`, `precision`, `recall`, `f1-score`) για το `test set`. Επίσης, εμφανίζει ένα συνολικό `confusion matrix` καθώς και 4 τυχαίες εικόνες ανά κατηγορία, με βάση την πρόβλεψη του CNN.



Αποτελέσματα



Παραπάνω βλέπουμε τον συνολικό confusion matrix που προκύπτει. Μπορούμε να δούμε ότι εντοπίζει τα αεροπλάνα αν και κάποια τα έχει θεωρήσει ως πλοία. Τα αυτοκίνητα επίσης εντοπίζονται αρκετά σωστά αλλά παρατηρούμε ότι λανθασμένα κατηγοριοποιεί αρκετά αυτοκίνητα ως φορτηγά. Στα πτηνά δυσκολεύεται αρκετά καθώς μπερδεύεται με τους σκύλους(fn) και γενικότερα κατηγοριοποιεί τα πτηνά σε άλλες κλάσεις(fr). Στις γάτες επίσης δεν ήταν αποτελεσματικό καθώς τις είδε ως σκύλους και πουλιά(fn), ενώ θεώρησε αρκετά μικρά τετράποδα ως γάτες. Παρακάτω, δεν βρήκε πολλά ελάφια διότι τα απέδωσε σε άλλες κατηγορίες αλλά καταφέρνει να μην θεωρεί τις άλλες κατηγορίες ελάφια. Τους σκύλους, τους μπερδεύει με γάτες και πτηνά. Τα βατραχάκια τα βρίσκει σχετικά ορθά αν εξαιρέσουμε ότι πολλά τα θεωρεί γάτες. Τα άλογα είναι σωστά. Τέλος τα φορτηγά τα μπερδεύει με τα αυτοκίνητα. Γενικότερα μπερδεύει όλα τα μικρά ζώα μεταξύ τους και τα αυτοκίνητα με τα φορτηγά.

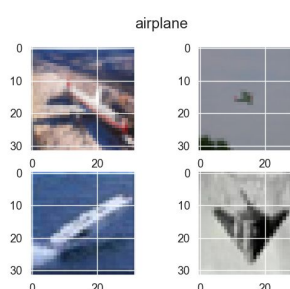
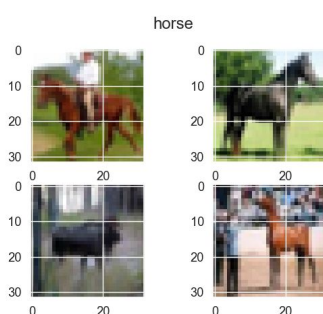
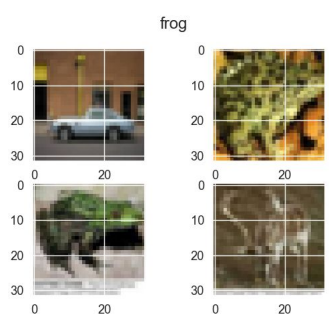
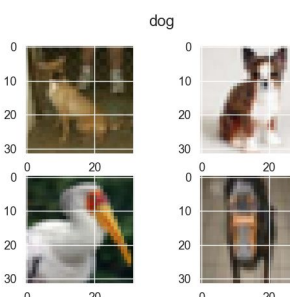
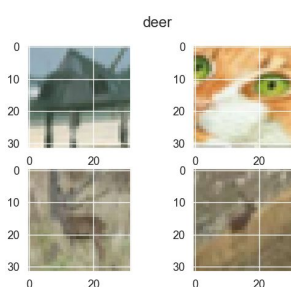
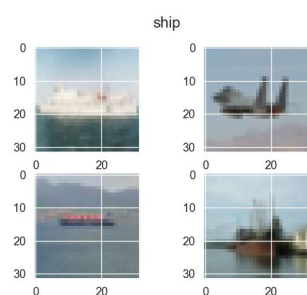
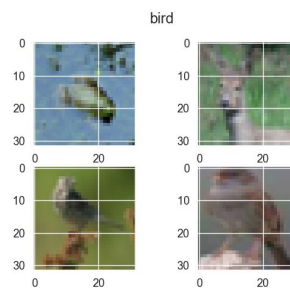
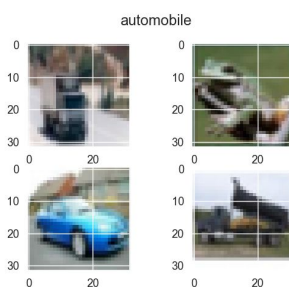
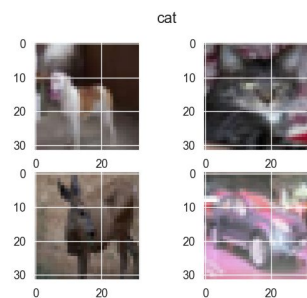
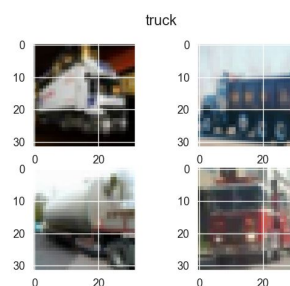
Απόδοση

- precision_score 0.5995
- recall_score 0.5903
- f1_score 0.5923
- accuracy_score 0.5903

Να τονίσουμε ότι τα αποτελέσματα δεν είναι αρκετά καλά. Εάν είχαμε περισσότερα δεδομένα θα ήταν πολύ καλύτερα.

Αποτελέσματα με εικόνες

Παραδείγματα εικόνων έπειτα από την πρόβλεψη του μοντέλου



LSTM - Μέρος 2

Σκοπός εργασίας

Στο δεύτερο μέρος της εργασίας αναπτύσσουμε ένα δίκτυο LSTM το οποίο μπορεί να μοντελοποιήσει την πρόσθεση και την αφαίρεση μεταξύ ακεραίων αριθμών. Ο κώδικας χωρίζεται σε δύο αρχεία. Στο πρώτο δημιουργούμε τα δεδομένα και εκπαιδεύουμε το μοντέλο ενώ στον δεύτερο ελέγχουμε την απόδοσή του και μπορεί ο χρήστης να ελέγξει την λειτουργικότητα του προγράμματος.

Επεξήγηση κώδικα

Το πρώτο βήμα είναι να θέσουμε το αποδεκτό αλφάβητο και το μέγιστο μήκος της συμβολοσειράς εισόδου και εξόδου το οποίο θα εξηγήσω αργότερα γιατί το χρειαζόμαστε. Ορίζουμε τις εποχές και παραμετροποιούμε το μοντέλο. Για κάθε εποχή δημιουργούμε καινούργια δεδομένα εκπαίδευσης και κάνουμε κάθε φορά fit. Στο τέλος αποθηκεύουμε το μοντέλο LSTM_1.h5.

Όσον αφορά τα δεδομένα, δημιουργούμε τυχαία ζευγάρια αριθμών τα αθροίζουμε/αφαιρούμε αναλόγως και ακολούθως δημιουργούμε τις αντίστοιχες συμβολοσειρές έτσι ώστε πάντα να προκύπτει το μέγιστο μήκος. Στη συνέχεια, κωδικοποιούμε την συμβολοσειρά σε μια αριθμητική ακολουθία που προκύπτει από το αλφάβητο και τέλος μετατρέπουμε την ακολουθία στα αντίστοιχα 0,1. Κατ' αυτόν τον τρόπο προκύπτουν τα δεδομένα εκπαίδευσης (εισόδου με τα αντίστοιχα αποτελέσματα) για το μοντέλο.

Παρακάτω διαβάζουμε το μοντέλο και θέτουμε το μέγεθος τεστ στο 3000 (30%). Δημιουργούμε τα δεδομένα ελέγχου ακριβώς με τον ίδιο τρόπο και υπολογίζουμε τιμές σύμφωνα με το μοντέλο. Ακολούθως, εκτελούμε την αντίστροφη διαδικασία μετατροπής από συνεχόμενα 0,1 σε χαρακτήρες αλφαβήτου. Εν τέλη, εμφανίζουμε μερικά παραδείγματα και προσφέρουμε την δυνατότητα στον χρήστη να πληκτρολογήσει διάφορες συμβολοσειρές.

Συναρτήσεις

- `Generate_data`:

Δημιουργεί τυχαία ζευγάρια αριθμών τα αθροίζει/αφαιρεί αναλόγως (`random_sum_pairs`). Ακολουθώς δημιουργεί τις αντίστοιχες συμβολοσειρές (`sum_to_string`) έτσι ώστε πάντα να προκύπτει το μέγιστο μήκος. Στη συνέχεια, κωδικοποιεί την συμβολοσειρά σε μια αριθμητική ακολουθία που προκύπτει από το αλφάβητο (`integer_encode`) και τέλος μετατρέπει την ακολουθία στα αντίστοιχα 0,1 (`one_hot_encode`). Κατ' αυτόν τον τρόπο επιστρέφει τα τελικά δεδομένα εκπαίδευσης, εισόδου και τα αντίστοιχα εξόδου, για το μοντέλο.

- `Random_sum_pairs`:

Δημιουργεί `n_range(1500)` τυχαία ζευγάρια αριθμών που ανήκουν στο εύρος 0-99. Επίσης υπολογίζει το άθροισμα τους. Τέλος επιστρέφει τους δυο αριθμούς σε μια λιστα και το άθροισμα τους.

- `Random_sub_pairs`:

Δημιουργεί όπως και η προηγούμενη `n_range(1500)` τυχαία ζευγάρια αριθμών που ανήκουν στο εύρος 0-99. Σε αντίθεση με την προηγούμενη υπολογίζει τη διαφορά τους. Τέλος επιστρέφει τους δυο αριθμούς σε μια λιστα και τη διαφορά τους.

- `Sum_to_string`:

Αρχικά υπολογίζει το μέγιστο δυνατό μήκος ενός αθροίσματος ($99+99=198...5$ ψηφία). Ακολουθώς προσθέτει το '+' μεταξύ των αριθμών καθώς και τόσα κενά στην αρχή της συμβολοσειρας ώστε να συμπληρώνουν ολοι μαζί οι χαρακτήρες το μέγιστο δυνατό μήκος. Το συγκεκριμένο βήμα είναι απαραίτητο διότι, δεν υπάρχει σύμβολο τερματισμού συμβολοσειρας, οπότε ο μόνος τρόπος για να γνωρίζει το πρόγραμμα που αρχίζει και που τελειώνει η πράξη είναι το σταθερό μήκος της. Παρόμοια διαδικασία εκτελούμε για την έξοδο γ.

- `Sub_to_string`:

Όπως και η προηγούμενη συνάντηση υπολογίζει το μέγιστο δυνατό μήκος ενός αθροίσματος ($99+99=198...5$ ψηφία). Ακολουθώς προσθέτει το '-' μεταξύ των αριθμών καθώς και τόσα κενά στην αρχή της συμβολοσειρας ώστε να συμπληρώνουν ολοι μαζί οι χαρακτήρες το μέγιστο δυνατό μήκος. Παρόμοια διαδικασία εκτελούμε για την έξοδο γ.

- **Integer_encode:**

Ο σκοπός της συνάρτησης είναι η αντιστοίχιση των χαρακτήρων. Αρχικά δημιουργεί ένα dictionary και αναθέτει σε κάθε γράμμα έναν αύξοντα αριθμό. Στη συνέχεια για κάθε χαρακτήρα στη συμβολοσειρά παραθέτει, σύμφωνα με το λεξικό, τον αντίστοιχο ακέραιο. Παρεμφερή πράττει και για την εξοδό.

- **One_hot_encode:**

Μετατρέπει κάθε νούμερο που προέκυψε από πριν σε μια ακολουθία 0,1. Να σημειωθεί ότι για κάθε χαρακτήρα όλα είναι 0 εκτός από εκείνο που βρίσκεται στη θέση που δείχνει ο ακέραιος. Δηλαδή το 5 θα έχει στην 5η θέση άσσο και τα υπόλοιπα θα είναι 0. Δεν είναι κώδικας συμπίεσης αλλά είναι ένας ευδιάκριτος κώδικας δηλαδή δεν συγχέει τους ακέραιους μεταξύ τους.

- **Read_data:**

Όπως η generate data αλλά μόνο για την εισοδό. Εκτελεί πρακτικά την μετατροπή των αριθμών που δίνει ο χρήστης σε 0,1 έτσι ώστε να δοθεί στο μοντέλο. Παίρνει το σύμβολο πράξης και αναλόγως το προσθέτει ή υπο αφαιρεί, φροντίζοντας παράλληλα να προσθέσει κατάλληλα κενά ώστε να προκύψει το απαραίτητο μήκος. Παρακάτω αντιστοιχεί τους χαρακτήρες σε ακέραιους και το μετατρέπει σε ακολουθία 0,1. Τέλος το προσφέρει σαν εισοδό στο μοντέλο ώστε αυτό να επιστρέψει την τιμή που υπολογίζει.

- **Invert:**

Κάνει ακριβώς την αντίθετη διαδικασία από το Integer_encode. Με το λεξικό αντιστοιχεί μια ακολουθία 0,1 σε έναν χαρακτήρα έτσι ώστε στο τέλος να προκύπτει η αρχική συμβολοσειρά.

- insert:

Γενικά διαβάζει μια συμβολοσειρα έως ότου αυτή να είναι 0. Ακολουθως ελέγχει εάν ταιριάζει στο regex, το οποίο δέχεται συμβολοσειρες που ξεκινάνε με αριθμο, ακολουθεί το σύμβολο της πράξης ή της αφαίρεσης και συνεχίζουν με αριθμό. (3+5,34-75,5752+4642) Επίσης μπορεί να ξεκινάει με έναν αριθμό και να μην ακολουθεί πράξη (6,43,373). Ως επακόλουθο μπορεί να δεχτεί συμβολοσειρες όπως 8φ,58+8γ, 4-7- και εφόσον δεν έχει στο τέλος \$ σταματάει στο νούμερο. Μόλις εντοπίσει τη φράση που ταιριάζει στο μοτίβο ξεχωρίζει τα ψηφία απο την πράξη ωστε μετέπειτα να ελέγξει οτι τα νούμερα ανήκουν στο επιτρεπτό όριο 1-99. Στη συνέχεια δημιουργεί μια λίστα 2 αντικειμένων όπου στο πρώτο βάζει το πρώτο ψηφίο που βρήκε και στο δεύτερο το 0 έτσι ώστε εάν ο χρήστης έχει δώσει μόνο ένα ψηφίο και τίποτα άλλο να εμφανίσει αυτό, αλλιώς προσθέτει το δεύτερο ψηφίο. εκτελεί την read_data και παίρνει ως απάντηση την τιμή που προέβλεψε το μοντελο.

Αποτελέσματα:

```
99-28, = 71
84-88, = -4
44-10, = 34
36-92, = -56
72-49, = 23
43-8, = 35
72-14, = 58
5-20, = -15
34-15, = 19
24-74, = -50
18-63, = -44
2-10, = -8
46-78, = -32
76-3, = 72
72-3, = 79
51-82, = -31
```

Εισόδη χρήση:

```
give some input or press 0 for exit6-0
6-0 = 6
give some input or press 0 for exit6
6 = 6
give some input or press 0 for exit6λ
6λ = 6
give some input or press 0 for exit6-2
6-2 = 4
give some input or press 0 for exit6-2-
6-2- = 4
give some input or press 0 for exit6-2-p
6-2-p = 4
give some input or press 0 for exit5+7
5+7 = 12
give some input or press 0 for exitλ
give some input or press 0 for exit50+99
50+99 = 149
give some input or press 0 for exit50-99
50-99 = -49
give some input or press 0 for exit0
```

Ενδεικτικά αποτελέσματα που εμφανίζει και παραδείγματα δοκιμών από χρήστη.

Απόδοση

- loss: 0.0677
- precision_score 0.9861
- f1_score 0.9833
- accuracy: 0.9853
- recall_score 0.983
- accuracy_score 0.9827

Τα αποτελέσματα είναι αρκετά καλά έως και τέλεια.