

# 3<sup>η</sup> Υποχρεωτική Εργασία

Ιανουάριος 2024

Γρηγορίου Στέργιος 9564 (THMMY)

## **Εισαγωγή:**

Σε αντίθεση με τις πρώτες 2 εργασίες, και κυρίως εξαιτίας έλλειψης παραλληλισμού και `gpu acceleration`, δεν διενεργήθηκε εκτενής αναζήτηση καταλληλότερων παραμέτρων για το μοντέλο κατηγοριοποίησης και των 10 κλάσεων, αλλά η προσοχή εστιάστηκε σε ένα ζευγάρι αυτών (συγκεκριμένα `auto vs truck`). Παρατίθενται αποτελέσματα για το καλύτερο μοντέλο κατηγοριοποίησης που βρέθηκε με 3 διαφορετικές στρατηγικές για τις 10 κλάσεις. Ωστόσο η κυρίως ανάλυση αφορά την δυαδική κατηγοριοποίηση.

Τέλος σε αντίθεση με τις προηγούμενες 2 εργασίες, σε αυτή ο αλγόριθμος χρησιμοποιεί μόνο την βιβλιοθήκη `NumPy` και την κλάση `KMeans` της `scikit-learn` για το κυρίως μέρος του μοντέλου.

## **Σύνολο Δεδομένων και Προεπεξεργασία:**

Στην εργασία χρησιμοποιήθηκε η βάση δεδομένων `CIFAR-10` που αποτελείται από 60.000 εικόνες `32x32x3 uint8 (RGB)`. Οι πρώτες 50.000 εικόνες απαρτίζουν το σύνολο εκπαίδευσης και οι υπόλοιπες 10.000 το σύνολο δοκιμής. Τα δεδομένα (οι εικόνες) χωρίζονται σε 10 κλάσεις `{airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck}` και γενικά η βάση είναι απολύτως ισορροπημένη (κάθε κλάση έχει 5000 παρατηρήσεις στο σύνολο εκπαίδευσης και 1000 στο σύνολο δοκιμής).

Σε αυτή την εργασία, όπως και στη δεύτερη, διενεργήθηκε προεπεξεργασία των δεδομένων. Η προεπεξεργασία αποτελείται από μετατροπή των τιμών κάθε πίξελ σε τυπική κανονική κατανομή, κι έπειτα ανάλυση κύριων συνιστωσών κατά την οποία κρατήθηκαν οι κύριες συνιστώσες που εξηγούν το 95% της διακύμανσης των πίξελ. Για την PCA χρησιμοποιήθηκε η αντίστοιχη κλάση της `scikit-learn`. Το αποτέλεσμα ήταν ότι ο νέος χώρος χαρακτηριστικών είχε μόλις 221 από 3072 διαστάσεις και τα χαρακτηριστικά όλα είχαν μέση τιμή 0. Για την δυαδική κατηγοριοποίηση, στο μεγάλο δείγμα δεν έγινε ξεχωριστή προεπεξεργασία, αλλά χρησιμοποιήθηκε η PCA που έγινε σε ολόκληρα τα δεδομένα, για να είναι πιο αντιπροσωπευτικά τα αποτελέσματα ως προς το πρόβλημα των 10 κλάσεων. Στο δεύτερο τεστ ωστόσο, που διατηρήθηκε μικρότερο δείγμα, έγινε ξεχωριστή προεπεξεργασία, ακολουθώντας τα ίδια βήματα.

## **Κώδικας:**

Για την ανάγνωση των δεδομένων χρησιμοποιήθηκαν οι ήδη υλοποιημένες (από την ενδιάμεση εργασία) συναρτήσεις *unpickle(file)* και *load\_data(path)*, ενώ δίνεται η δυνατότητα να χρησιμοποιηθεί η *load\_cifar10()* του keras.

Για το RBF δίκτυο, δημιουργήθηκαν τρεις κλάσεις.

Η πρώτη *RBFLayer*, για δεδομένα κέντρα, αναπαριστά το κρυφό στρώμα του δικτύου. Ενώ, δίνεται η δυνατότητα να εκπαιδευτεί μέσω του αλγορίθμου RLS (Recursive Least Squares) το στρώμα εξόδου για δυαδική κατηγοριοποίηση (ένας κόμβος εξόδου). Συμπεριλαμβάνει, λοιπόν, τρεις μεθόδους. Η πρώτη *RLSstep* για δεδομένα εισόδου υπολογίζει με βάση τα κέντρα του αντικειμένου, τις εξόδους του κρυφού στρώματος. Η δεύτερη *fit* εκπαιδεύει τα βάρη των συναρτήσεων βάσης με τον αλγόριθμο RLS. Οι πράξεις πινάκων γίνονται με τη χρήση της NumPy. Τέλος η τρίτη, επιστρέφει τις προβλέψεις για δεδομένη είσοδο, είτε ως πραγματικό αριθμό είτε ως δυαδική ετικέτα (-1,1).

Η δεύτερη *RBFnn*, για δεδομένο αριθμό κέντρων και στρατηγικής κατηγοριοποίησης, εκπαιδεύει ένα δίκτυο RBF, με τον συνδυασμό kmeans, RLS, να λύνει ένα πρόβλημα κατηγοριοποίησης N κλάσεων. Ουσιαστικά εκπαιδεύει τα κέντρα του κρυφού στρώματος με την KMeans της scikit-learn, κι έπειτα για κάθε κόμβο εξόδου εκπαιδεύει τα αντίστοιχα βάρη με τον αλγόριθμο RLS. Κάθε κόμβος εξόδου αντιστοιχεί σε ένα πρόβλημα δυαδικής κατηγοριοποίησης το οποίο ορίζεται από τον πίνακα του σχήματος κατηγοριοποίησης. Τα πιθανά σχήματα κατηγοριοποίησης είναι: ένα ενάντια όλων, ένα προς ένα και σχήμα που ορίζεται με δυαδικό πίνακα N σειρών. Η κλάση έχει μεθόδους *fit* και *predict* που με βάση το σχήμα εκπαιδεύουν τους κόμβους εξόδου και αποφασίζουν για την τελική πρόβλεψη του δικτύου αντίστοιχα. Όσον αφορά το κομμάτι της πρόβλεψης δίνεται η δυνατότητα, αντί για κλασσικό hard voting στην μέθοδο one vs one, να γίνει πρόβλεψη μέσω 2 ακόμα πυκνά συνδεδεμένων στρωμάτων στην έξοδο του RBF στρώματος.

Η τρίτη *binary\_model* ουσιαστικά ενσωματώνει τα δύο κομμάτια της εκπαίδευσης (κέντρα κρυφού στρώματος, βάρη συναρτήσεων βάσης) σε ένα μοντέλο, αποκλειστικά για δυαδική κατηγοριοποίηση. Τα κέντρα μπορούν να εκπαιδευτούν με kmeans ή να επιλεγούν τυχαία από το σύνολο εκπαίδευσης, ενώ τα βάρη εκπαιδεύονται με τον αλγόριθμο RLS.

Τέλος για την εξαγωγή αποτελεσμάτων και την αξιολόγηση των μοντέλων, χρησιμοποιήθηκε η *evaluate\_model* της δεύτερης εργασίας, ενώ στα αρχικά πειράματα χρησιμοποιήθηκε η *optimal\_k* για την εκτίμηση του κατάλληλου αριθμού κέντρων για την kmeans.

## **Μέθοδος:**

Για τον διαχωρισμό ολόκληρης της βάσης δεδομένων (10 κλάσεις) δοκιμάστηκε (σχεδόν) κάθε πιθανή στρατηγική:

- One vs Rest με σταθερά κέντρα και εκπαίδευση βαρών με RLS ή back-propagation
- One vs One με σταθερά κέντρα και εκπαίδευση βαρών με RLS ή back-propagation
- Error Correction Classification Scheme με κώδικα  $d_{min} = 7$  με 25 δυαδικούς classifiers.
- One vs One με RLS και μετά back propagation στις πραγματικές ( $\in \mathbb{R}$ ) προβλέψεις.
- Σε προηγούμενη έκδοση της RBFLayer δοκιμάστηκε, η έξοδος να έχει 10 κόμβους με one-hot encoding και να εκπαιδευτούν τα βάρη με RLS.

Για όλα τα παραπάνω, τα αποτελέσματα δεν ήταν καλύτερα από τα benchmark μοντέλα (παρά το γεγονός πως στη βιβλιογραφία αναφέρεται ακρίβεια  $>50\%$  για όλη τη cifar-10). Για αυτό, μιας και η πολυπλοκότητα του συγκεκριμένου RLS είναι  $O(n^3)$ , εστιάστηκε η προσοχή σε δυαδική κατηγοριοποίηση, όπου ήταν δυνατό να αυξηθούν αρκετά τα κέντρα (σε σχέση με το σύνολο). Στο τέλος, ωστόσο, παρουσιάζονται και αποτελέσματα για τις 10 κλάσεις με μικρό αριθμό κέντρων.

Για τον διαχωρισμό των κλάσεων αυτοκίνητο και φορτηγό, δοκιμάστηκε μια σειρά από παραμέτρους, με δύο τρόπους εκπαίδευσης του κρυφού στρώματος και εκπαίδευση των βαρών με RLS. Τα αποτελέσματα φαίνονται παρακάτω. Γενικά, παρατηρήθηκε ταχεία σύγκλιση σε κάθε περίπτωση (από άποψη εποχών), οπότε τα μοντέλα είναι εκπαιδευμένα για 2 εποχές. Επίσης δοκιμάστηκαν ακραίες τιμές της παραμέτρου ομαλοποίησης, αλλά στο τέλος επιλέχθηκε η τιμή 0.25. Δεν παρατηρήθηκε ιδιαίτερη μεταβολή της επίδοσης, ίσως λίγο αργότερη σύγκλιση.

Γίνανε δοκιμές για μέγεθος δείγματος 10000 και για μέγεθος δείγματος 2000. Ο λόγος ήταν για να παρατηρηθεί η χωρητικότητα του μοντέλου, απέναντι στα benchmark μοντέλα, καθώς για μεγάλες τιμές κρυφών νευρώνων, ο αλγόριθμος (τουλάχιστον όπως έχει υλοποιηθεί) είναι υπερβολικά χρονοβόρος.

### **Αποτελέσματα:**

Παρακάτω φαίνονται οι μετρικές για διάφορες τιμές του πλήθους νευρώνων στο κρυφό επίπεδο, πάνω για κέντρα υπολογισμένα με kmeans και κάτω για τυχαία κέντρα από το σύνολο εκπαίδευσης. Το σύνολο εκπαίδευσης αποτελείται από 10.000 παρατηρήσεις και το σύνολο δοκιμής από 2.000 (απολύτως ισορροπημένο μεταξύ των κλάσεων). Θεωρούμε κλάση 1 (positive) το αυτοκίνητο. Για σύγκριση οι αντίστοιχες μετρικές για τα benchmark μοντέλα.

w/ K-means

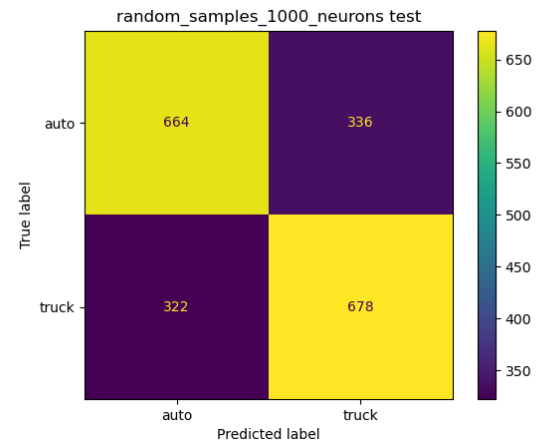
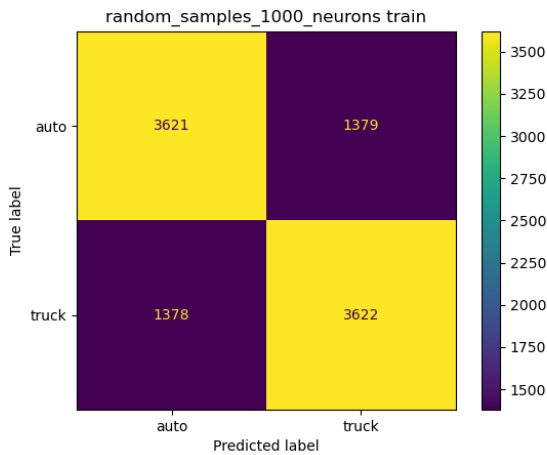
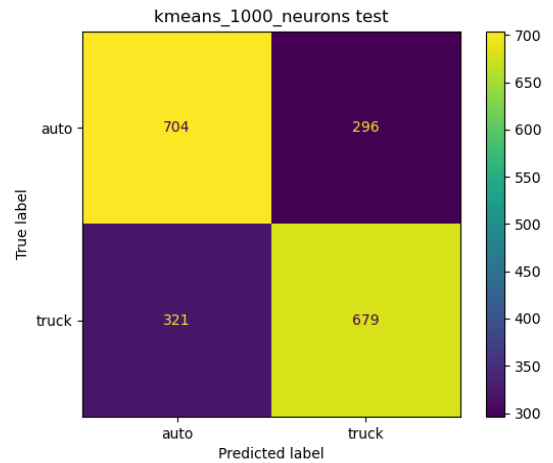
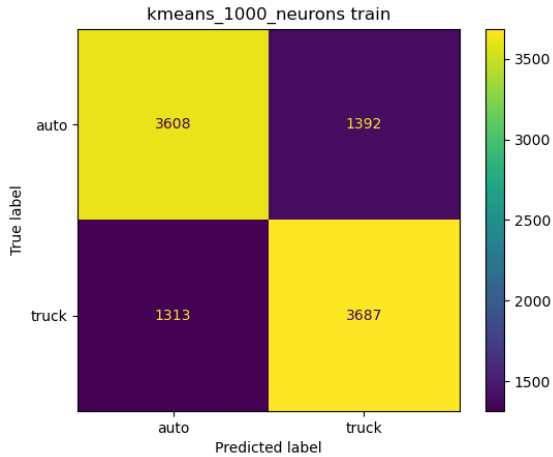
Neurons	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time (s)
<b>17</b>	64.89	65.75	64.11	65.1	1.83
<b>40</b>	66.02	66.3	65.71	65.99	2.49
<b>60</b>	66.46	66.65	65.52	65.42	3.07
<b>99</b>	67.62	67.05	66.78	66.11	4.39
<b>120</b>	68.77	68	68.08	67.38	5.85
<b>250</b>	69.99	68.45	69.95	68.62	23.42
<b>500</b>	71.35	68.85	68.8	68.8	199.17
<b>1000</b>	<b>72.95</b>	<b>69.15</b>	<b>72.73</b>	<b>69.53</b>	2837.95

w/ random centers

Neurons	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time(s)
<b>17</b>	64.02	64.25	61.99	62.31	1.46
<b>40</b>	66.05	66.15	65.17	64.83	2.02
<b>60</b>	65.83	64.5	66.41	65.8	2.54
<b>99</b>	67.04	67.5	65.76	66.18	3.96
<b>120</b>	67.88	67.05	67.39	66.57	5.05
<b>250</b>	68	66.3	68.96	<b>67.53</b>	23.9
<b>500</b>	69.89	67.2	69.32	66.6	200.4
<b>1000</b>	<b>70.12</b>	<b>67.31</b>	<b>69.93</b>	66.3	2838.45

/	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time (s)
<b>NN-1</b>	100	69.4	100	70.06	1.1
<b>NN-3</b>	86.12	71.45	86.32	72.22	1.33
<b>NC</b>	63.43	64.15	63.02	63.58	0.01

Παρακάτω φαίνονται οι πίνακες σύγχυσης (στάδιο εκπαίδευσης και δοκιμής) για τα καλύτερα μοντέλα σε κάθε μέθοδο εκπαίδευσης (1000 νευρώνες/kmeans και τυχαία επιλογή).



Παρακάτω φαίνονται οι μετρικές για διάφορες τιμές του πλήθους νευρώνων στο κρυφό επίπεδο, πάνω για κέντρα υπολογισμένα με kmeans και κάτω για τυχαία κέντρα από το σύνολο εκπαίδευσης. Το σύνολο εκπαίδευσης αποτελείται από 2.000 παρατηρήσεις και το σύνολο δοκιμής από 2.000 (απολύτως ισορροπημένο μεταξύ των κλάσεων). Θεωρούμε κλάση 1 (positive) το αυτοκίνητο. Για σύγκριση οι αντίστοιχες μετρικές για τα benchmark μοντέλα.

/	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time (s)
<b>NN-1</b>	100	64.1	100	65.18	0.33
<b>NN-3</b>	83.8	66.55	83.83	67.7	0.144
<b>NC</b>	65.75	63.4	64.81	63.36	0.09

w/ kmeans

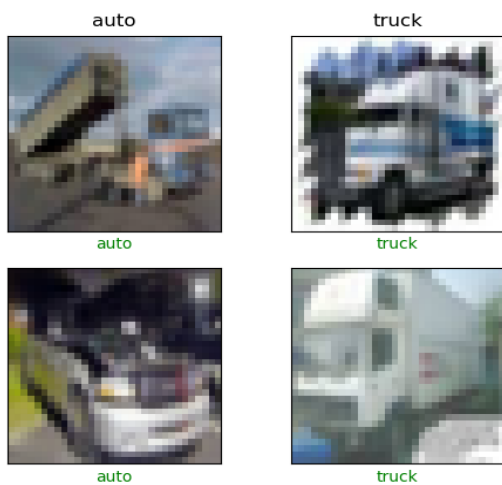
Neurons	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time
100	70.3	66.45	68.03	64.7	2.52
235	73.5	<b>67.45</b>	72.99	<b>66.66</b>	4.15
250	74.4	67.05	73.82	66.46	10.02
500	77.75	64.9	77.26	63.55	59.01
1000	<b>84.35</b>	64.65	<b>84.39</b>	65.16	724.53

w/ random centers

Neurons	Train Accuracy	Test Accuracy	Train F1-score	Test F1-score	Training time
100	71.45	<b>66.2</b>	70.97	<b>66.37</b>	1.37
250	73.05	66.1	71.76	65.58	10.75
500	77.75	65.45	77.19	64.51	41.47
1000	<b>85.2</b>	65.35	<b>85.23</b>	65.29	487.64

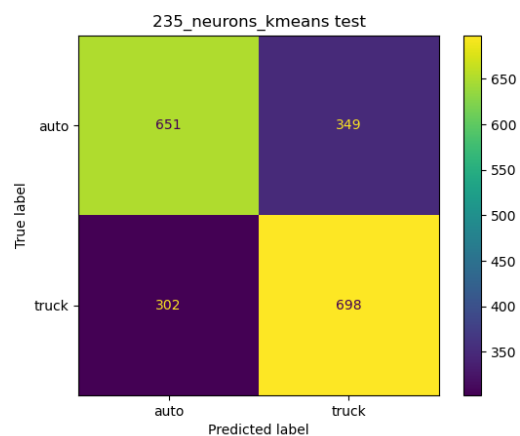
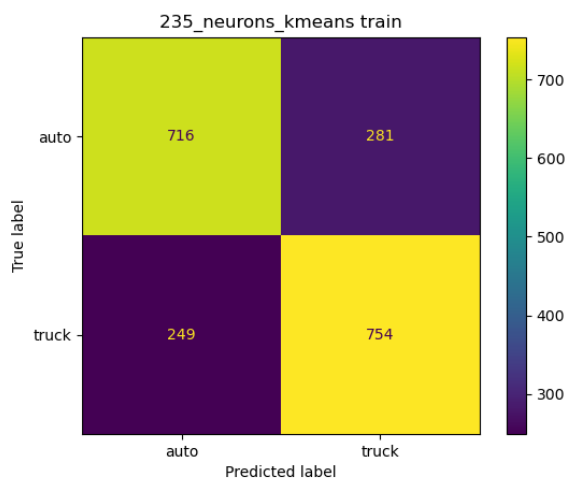
Παρακάτω φαίνονται οι πίνακες σύγχυσης (στάδιο εκπαίδευσης και δοκιμής) για το καλύτερο μοντέλο, μαζί με τα αντίστοιχα παραδείγματα.

Classification Examples 235\_neurons\_kmeans train



Classification Examples 235\_neurons\_kmeans test

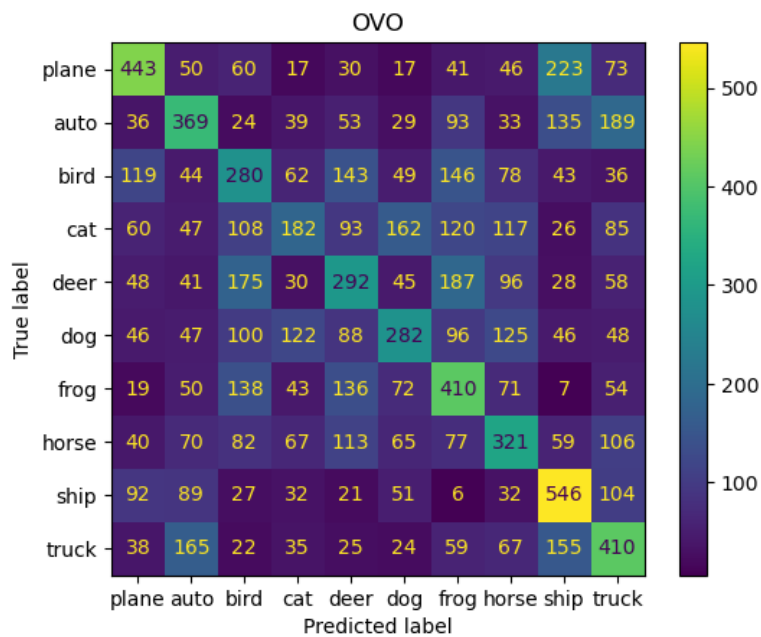




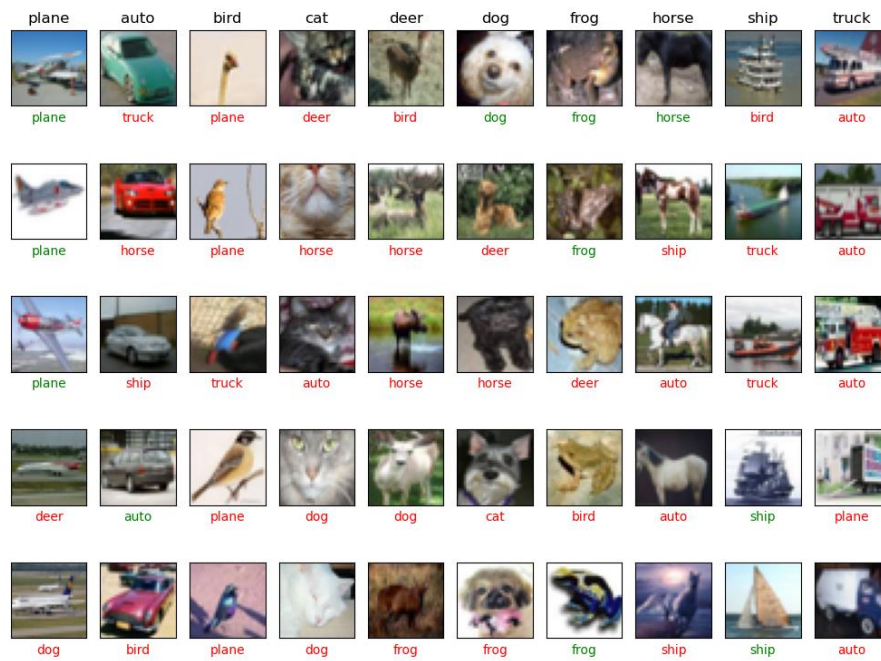
## Multiclass

Παρακάτω φαίνονται τα αποτελέσματα για τις 10 κλάσεις, με 99 νευρώνες και one vs one.

Accuracy: 35.35%

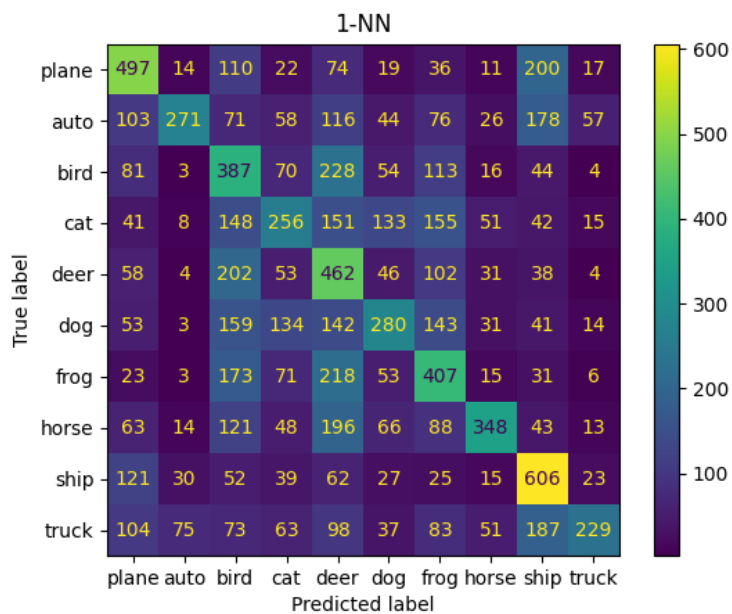


Classification Examples OVO



Για σύγκριση το καλύτερο απλό μοντέλο (πλησιέστερος γείτονας).

Accuracy: 37.43%





## **Σχόλια-Συμπεράσματα:**

Αρχικά το μοντέλο, παρά την υψηλή υπολογιστική πολυπλοκότητά του δεν φαίνεται να υπερτερεί των πιο απλών μοντέλων σύγκρισης. Ακόμα και στο δυαδικό πρόβλημα είναι οριακά καλύτερο, αλλά μόνο όταν οι κρυφοί νευρώνες είναι σχετικά πολλοί (σε σχέση με το μέγεθος του δείγματος).

Στα πιο τεχνικά, παρόλο που η παράμετρος ομαλοποίησης δεν φάνηκε να κάνει μεγάλη διαφορά στην επίδοση του μοντέλου, είναι ξεκάθαρο ότι το μέγεθος του κρυφού επιπέδου είναι αυτό που καθορίζει την χωρητικότητα του μοντέλου. Από τους αντίστοιχους πίνακες γίνεται ξεκάθαρο πώς για αριθμό κέντρων πάνω από 10-15% του μεγέθους του δείγματος το μοντέλο αρχίζει να μαθαίνει καλύτερα το σύνολο και από underspecified σιγά σιγά κάνει overfitting. Δυστυχώς η υλοποίησή μου δεν μου επιτρέπει να το εξετάσω σε όλο το dataset όπως θα ήθελα και δεν μπορώ να βγάλω σαφή συμπεράσματα.

Ένα ακόμα άξιο αναφοράς σημείο, είναι η θετική επίδραση της κανονικοποίησης των εξόδων του κρυφού επιπέδου, στην απόδοση του μοντέλου. Ουσιαστικά η gaussian έξοδος μετατρέπεται σε σιγμοειδή, και από τα πειράματα που διενεργήθηκαν φαίνεται να βοηθάει.

Τέλος από τους πίνακες, είναι επίσης εμφανές πως η υλοποίηση του RLS είναι σχεδόν  $O(n^4)$ , ως προς το μέγεθος του κρυφού επιπέδου, γεγονός για το οποίο φταίει ο αντίστροφος πίνακας συσχέτισης.

## **Παράρτημα:**

Στο .zip αρχείο υπάρχει αυτή η αναφορά, το αρχείο κώδικα Grigoriou\_9564\_3.py καθώς κι ένα αρχείο Grigoriou\_9564\_3.ipynb στο οποίο υπάρχουν σε σχόλια τα διάφορα πειράματα που έγιναν, καθώς και χρήσεις της κλάσης RBFnn που στο .py αρχείο δεν χρησιμοποιείται (κατέληξα να ασχοληθώ με δυαδική κατηγοριοποίηση).