

Νευρωνικά Δίκτυα - Βαθιά Μάθηση

Τμήμα Πληροφορικής ΑΠΘ 7ο εξάμηνο

Εργασία 2 - Support Vector Machine

Αλεξία Νταντουρή ΑΕΜ: 3871

2η Εργασία

Να γραφεί πρόγραμμα σε οποιαδήποτε γλώσσα προγραμματισμού το οποίο να υλοποιεί ένα **Support Vector Machine** που θα εκπαιδεύει για να επιλύει ένα από τα παρακάτω προβλήματα:

1. Διαχωρισμό 2 ή όλων των κλάσεων που υπάρχουν στις Cifar-10 ή SVHN
2. ή να επιλύει οποιοδήποτε πρόβλημα κατηγοριοποίησης πολλών κλάσεων

Σημείωση

Προτείνεται να διαβαστεί το παρόν report παράλληλα με *python notebook*.

Με **πράσινο** μαρκάρω τις γραμμές των πινάκων με το υψηλότερο test accuracy και με **κίτρινο** τις γραμμές με το δεύτερο υψηλότερο test accuracy.

1. Testing different kernels

With PCA:

	Training Time	Train Accuracy	Test Accuracy
Linear	1694.97	0.42032	0.4075
Polynomial	570.09	0.63514	0.4647
Sigmoid	466.35	0.22366	0.2235
RBF	405.14	0.65472	0.5401

Φαίνεται πως ο πυρήνας **RBF** είναι ο **γρηγορότερος** και ο **πιο ακριβής** τόσο για το training όσο και για το test set.

Χωρίς PCA, η εκπαίδευση ενός SVM με RBF kernel (ο οποίος είναι και ο πιο γρήγορος πυρήνας) πήρε περίπου 92 λεπτά, το οποίο είναι πολύ αργό για πειραματισμό με διάφορες υπερπαραμέτρους. Έτσι, ο πειραματισμός με διάφορες υπερπαραμέτρους έγινε στα δεδομένα μετά από PCA.

2. Testing different degrees

For the polynomial kernel:

	Training Time	Train Accuracy	Test Accuracy
deg = 1	462.56	0.4181	0.4111
deg = 2	438.24	0.54444	0.4731
deg = 3 (default)	568.23	0.63514	0.4647
deg = 4	702.01	0.60616	0.3919

Για τον **πολυωνυμικό** πυρήνα, ο βαθμός **deg=2** δίνει το υψηλότερο test accuracy, ενώ για **deg=3** έχουμε το υψηλότερο training accuracy.

3. Testing different C values

For the RBF kernel:

	Training Time	Train Accuracy	Test Accuracy
C = 0.01	645.64	0.35946	0.3644
C = 0.1	412.11	0.47696	0.4605
C = 1 (default)	366.03	0.65472	0.5401
C = 2	361.56	0.73614	0.5536
C = 10	458.86	0.9216	0.5623
C = 20	567.77	0.96436	0.5555
C = 50	668.88	0.98974	0.5498

Χρησιμοποιώντας τον **RBF** kernel, έχουμε για **C=10** το υψηλότερο accuracy τόσο για το training όσο και για το test set.

4. Testing different gamma values

For the RBF kernel:

	Training Time	Train Accuracy	Test Accuracy
gamma = scale (0.00583) (default)	445.73	0.65472	0.5401
gamma = auto (0.010101)	459.74	0.75644	0.5523

gamma = 0.02	754.62	0.89596	0.5472
gamma = 0.1	1774.66	0.99842	0.2694

Για τον **RBF** πυρήνα, το **gamma='auto'** δίνει το υψηλότερο test accuracy, ενώ για **gamma=0.02** έχουμε το δεύτερο υψηλότερο test accuracy και training accuracy μεγαλύτερο σε σύγκριση με **gamma='auto'**.

Για μεγάλο gamma=0.1, βλέπουμε πως οδηγούμαστε σε overfitting, καθώς το train accuracy αυξάνεται πολύ, ενώ το test accuracy πέφτει δραματικά.

5. Testing different coef0 values

For the polynomial kernel:

	Training Time	Train Accuracy	Test Accuracy
coef0 = -1.0	341.85	0.16496	0.1626
coef0 = 0.0 (default)	614.83	0.63514	0.4647
coef0 = 1.0	417.96	0.77496	0.549
coef0 = 2.0	515.31	0.80234	0.5496
coef0 = 5.0	1005.01	0.84216	0.5414

Παρατηρούμε πως για coef0=2.0 έχουμε το υψηλότερο train και test accuracy.

6. Randomized Search

Το randomized search με 5 επαναλήψεις, έδωσε ως καλύτερο μοντέλο ένα SVM με RBF kernel, C=1 και gamma='scale'. Αυτό το μοντέλο πιάνει 80.234% train accuracy και 54.96% test accuracy.

7. Voting Ensemble

Χρησιμοποιούμε 3 SVMs και τα περνάμε από έναν Voting Classifier για να δούμε αν ο Voting Classifier θα δώσει καλύτερα αποτελέσματα σε σχέση με τη χρήση ενός μοναδικού SVM.

	Training Time	Train Accuracy	Test Accuracy
Hard Voting	983.17	0.89246	0.5602
Soft Voting	5053.36	0.90184	0.5695

Το soft voting δίνει υψηλότερο accuracy, τόσο για το training όσο και για το test set σε σχέση με το hard voting. Βέβαια, ο χρόνος της εκπαίδευσης με soft voting είναι πολύ πιο μεγάλος σε σχέση με του hard voting.

Με τη χρήση του soft voting πιάνουμε υψηλότερο accuracy από το να χρησιμοποιούσαμε ένα μοναδικό SVM.

8. The best SVM

Το καλύτερο SVM που βρήκα όσον αφορά το test accuracy είναι ένα SVM με RBF kernel, $C=2$ και $\gamma=\text{'auto'}$. Αυτό το μοντέλο πιάνει 86.304% train accuracy και 56.36% test accuracy.

9. Passing the output of the last hidden layer of the CNN to the SVM

Χρησιμοποιούμε το συνελικτικό νευρωνικό δίκτυο (CNN) από την προηγούμενη εργασία, παίρνουμε την έξοδο του τελευταίου κρυφού επιπέδου του και την περνάμε από ένα SVM για να δούμε αν θα δώσει καλύτερα αποτελέσματα σε σχέση με τη χρήση μόνο του CNN.

	Training Time	Train Accuracy	Test Accuracy
CNN	745	84.02%	71.9%
SVM	104	86.304%	56.36%
CNN + SVM	849	88.128%	72.4%

Βλέπουμε πως με τον συνδυασμό των δύο μοντέλων παίρνουμε μεγαλύτερο training (88.128%) και test accuracy (72.4%).

10. Comparing Results

Ο κατηγοριοποιητής πλησιέστερου γείτονα (KNN) δεν έχει φάση εκπαίδευσης, γι' αυτό στις στήλες των KNN έβαλα τους χρόνους πρόβλεψης των train και test sets.

	NCC	KNN 1	KNN 3	FC MLP	CNN	SVM
Training time	0.7	574 (train set)	556 (train set)	53	745	337
		129 (test set)	109 (test set)			
Train accuracy	26.968%	100%	57.904%	77.094%	84.02%	86.304%
Test accuracy	27.74%	35.39%	33.03%	56.03%	71.9%	56.36%

Το SVM πιάνει υψηλότερο accuracy σε σχέση με τα υπόλοιπα μοντέλα εκτός από το συνελκτικό νευρωνικό δίκτυο.

	NCC	KNN 1	KNN 3	FC MLP	CNN	SVM	SVM VOTING	CNN passed to SVM
Training time	0.7	574 (train set)	556 (train set)	53	745	337	5053	915
		129 (test set)	109 (test set)					
Train accuracy	26.968%	100%	57.904%	77.094%	84.02%	86.304%	90.184%	88.128%
Test accuracy	27.74%	35.39%	33.03%	56.03%	71.9%	56.36%	56.95%	72.4%

Με το συνδυασμό του CNN με το SVM παίρνουμε μεγαλύτερο accuracy από το να χρησιμοποιούσαμε το CNN μόνο του.