Kaggle Project (Classifying Movie Reviews)

UID: u1208099

Kaggle Username: SrivathsanGr

Full Name: SRIVATHSAN GOMADAM RAMESH

- Algorithms Used and their Accuracies:

I. Averaged Perceptron:

(a) The best hyper-parameters:
Ŋ (eta) = 0.01 gives the highest accuracy .

(b) The cross-validation accuracy for the best hyperparameter.
Cross-validation accuracy of chosen hyperparameter = 87.99 %

(c) The total number of updates the learning algorithm performs on the training set
20000 updates for avgW (Since update is made for every example irrespective of correctness in average perceptron). 1445 is chosen as the best update based on highest dev set accuracy.

(d) Test set accuracy:

87.58 %

(e) Kaggle Score:

0.87648

(f) Epoch with Best dev set accuracy:
⇨Best Accuracy Obtained at 19th epoch

(Since the weights and bias are initialized randomly, the above parameters are subject to show slight variations for each run)

II. Decaying the Learning Rate:

(a) The best hyper-parameters:
Ŋ (eta) = 1.0 gives the highest Accuracy.

(b) The cross-validation accuracy for the best hyperparameter.
Best Hyper Parameter Accuracy: 58.36

(c) Best Update:
1952 is chosen as the best update based on the highest dev set accuracy

(d) Kaggle Score:
0.87616

(e) Test set accuracy:
87.52 %

(f) Epoch with Best dev set accuracy

- Best Accuracy Obtained at 5th epoch

(Since the weights and bias are initialized randomly, the above parameters are subject to show slight variations for each run).

III. Averaged Logistic Regression:

(a) The best hyper-parameters:
Sigma2 = 10000
Gamma =0.1

(b) The cross-validation accuracy for the best hyperparameter.
Best Hyper Parameter Accuracy: 87.912

(c) Kaggle Score:
0. 87776

(d) Test set accuracy:
87.912 %

(e) Epoch with Best dev set accuracy

- Best Accuracy Obtained at 19th epoch

IV. Support Vector Machines:

(a) The best hyper-parameters:

Trade-Off: C = 10000

Gamma: 0.001.

(b) The cross-validation accuracy for the best hyperparameter.
Best Hyper Parameter Accuracy: 86.58

(c) Kaggle Score:
0. 87264

(d) Test set accuracy:
87.488 %

(e) Epoch with Best dev set accuracy

- Best Accuracy Obtained at 5th epoch

Shuffling is done for every epoch.

V. Logistic Regression:

(a) The best hyper-parameters:
Sigma2: 100000
Gamma: 0.1

(b) Kaggle Score:
0. 86704

(c) Test set accuracy:
87.056%

(d) Epoch with Best dev set accuracy

- Best Accuracy Obtained at 8th epoch

Shuffling is done for every epoch.

VI. Neural Net MultiLayer Perceptron:

For this algorithm, Neural Network Multi Layer Perceptron is used.
Sparse Vector is used to store the data for train set, test set and eval set.
Multi Layer Perceptron model parameters:
Number of Hidden Layers:

- 3 layers
- Activation Method Used: relu
- Solver Used for Weight Optimization: adam stochastic gradient based optimizer.
- Learning rate used: 0.001
- Regularization term Alpha(to prevent overfitting): 0.0001
- Shuffling is done on each iteration
- Momentum of gradient descent update: 0.9
- Confusion Matrix on Test Set:
  Classification Report

	precision	recall	f1-score	support
0	0.85	0.87	0.86	6188

1	0.87	0.85	0.86	6312
micro avg	0.86	0.86	0.86	12500
macro avg	0.86	0.86	0.86	12500
weighted avg	0.86	0.86	0.86	12500

Kaggle Score: 0.85984

Other Submissions:

VII. Averaged SVM:

Best Hyper Parameter Chosen:

(a) The best hyper-parameters:

Trade-off Parameter C: 90000

Gamma: 0.001

(b) Kaggle Score:

0. 84464

(c) Test set accuracy:

84.088 %

(d) Epoch with Best dev set accuracy

- Best Accuracy Obtained at 10^{th} epoch

Shuffling is done for every epoch.

VIII. K – Nearest Neighbors:

Chosen Values of k = 7;

Observation:

The k = 7,

Observation: The Algorithm is very inefficient for huge datasets and higher dimension spaces in terms of space and time complexity. Both Euclidian distance and hamming distance performs very poor

Misc: Other Algorithms tried which doesn't give good accuracy,

Bagged SVM over Logistic Regression : Trained 200 SVM and performed feature

transformation on dataset and ran Logistic Regression on that.

Bagged SVM over Perceptron: Trained 200 SVM by bagging and performed feature transformation on dataset and ran Perceptron over that.

- Any pre-processing or feature transformations you may have done

  Training Data is shuffled for randomization using random.shuffle function in python.
  Out of the shuffled training Data, 20% ~5000 rows were taken as dev set.
  The remaining 20,000 rows are shuffled and split into 5 fold CV splits for Cross Validation.
  Code is much optimized and the variable is destroyed dynamically when its out of scope to make it memory and space efficient.

- what works, what doesn't work
  Shuffling the dataset improved the result. Most of the times randomization of weight vector and bias yields better results rather than seeding it to a constant value.

  Increasing the number of epochs not necessarily increases the Kaggle/test accuracy.
  I have tried most of the algorithms thought in class as part of Kaggle project and realized that having chosen proper parameters, all algorithms perform within same range.

- How the project helped you to understand ML
  - ⇨ On trying out different algorithms, I realized that all the algorithms with properly chosen hyperparameters and epochs almost yields the same accuracy on test set.
  - ⇨ Naïve Bayes sometimes performs better than other classifiers when the dataset is skewed.
  - ⇨ Selection of hyperparameter, Randomization, Shuffling, initialization of weights does have impact in the result.
  - ⇨ This course really helped me to practically understand the mechanism of what happens in each learning algorithm, why we need to choose a algorithm and loss function associated with it. Could visibly observe the relation between theory and practical ML.
  - ⇨ Understood bagging and boosting. How set of weak classifier can be combined to form a strong learner using ensemble.
  - ⇨ In future, I would like to take it forward implementing different neural networks on the same dataset and check how good is that.