CS57800: Statistical Machine Learning

Homework 2

Yu-Jung Chou

chou63@purdue.edu

Due: Oct 9, 2018 on Tuesday

1 Implementation

Before training the classifier, pre-process the training set. Pre-process includes: shuffle the data, limit the size of training data sets, reshape data from 2D to 1D , round the value, and add the bias to the end of the data (winnow algorithm doesn't have to add the bias).

In the training process, we will have ten sub-classifiers for classifying whether the input should be labeled as the corresponding class (digit in MNIST) or not.

In perceptron training process, the initialization includes randomization of weights and bias, and they are updated when the current prediction is wrong, which is not true positive or true negative. On the other hand, winnow algorithm initialize the weights to 1, and update the weights when make a false prediction. The weights will be promoted for false negative or demoted for false positive by the multiplication factor.

Finally, the testing process, we used the trained 10 classifiers to get the 10 activation values of the test input, and labeled it as the one classifier with the max activation.

2 Vanilla / Basic Perceptron

2.1 Vanilla Perceptron Algorithm

Algorithm 1: Vanilla Perceptron Training Algorithm$(D, MaxIter)$

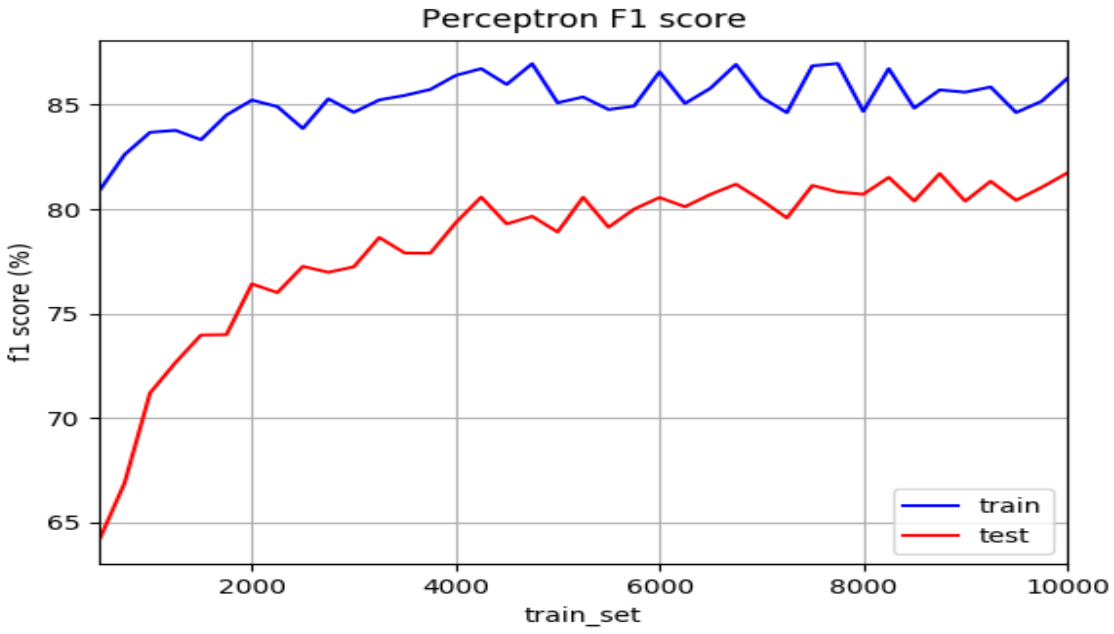
$W \leftarrow [randomInt_1, randomInt_2, ...randomInt_D]$;
$b \leftarrow randomInt$;
for $iter = 1$ .. $MaxIter$ **do**
  for $(x, y) \in D$ **do**
    activation $= W \cdot x + b$;
    if $y \times activation \leq 0$ **then**
      $W \leftarrow W + y \times x$;
      $b \leftarrow b + y$;
    else
      continue ;
return (W, b);

Algorithm 2: Vanilla Perceptron Classify Algorithm(W, b, x)

$activation = W \cdot x + b$;
return SIGN$(activation)$;

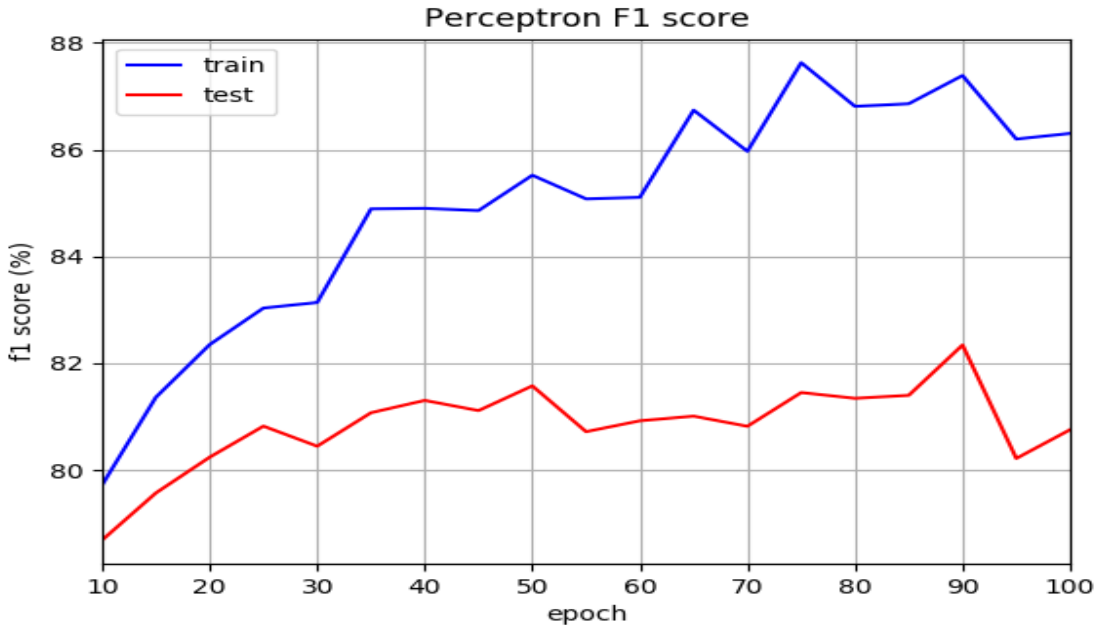
2.2 Effect of size of training set in learning

Vary the size of training set from 500 to 10000 with a step of 250;
set default number of epoch 50 and default learning rate 0.001.



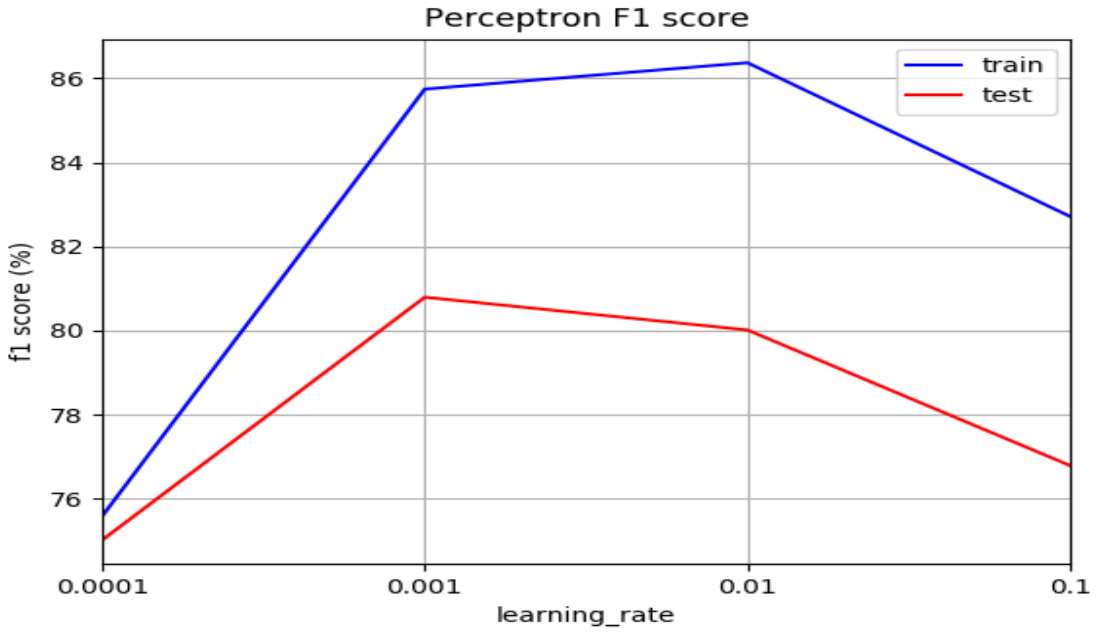
2.3 Effect of number of epoch in learning

Vary the number of epoch from 10 to 100 with step of 5;
set default size of training set 10k and default learning rate 0.001.



2.4 Effect of learning rate in learning

Vary the learning rate for the values 0.0001, 0.001, 0.01, 0.1;
set default number of epoch 50 and default size of training set 10k.



2.5 Observation

When the size of the training set is taken as the hyper-parameter, the f score converged after the size is over 6k.

When epoch is taken as the hyper-parameter, the f score converged after the epoch is over 75.

When the learning rate is taken as the hyper-parameter, the f score increases as the learning rate increases between learning rate=0.0001 and 0.001. But it overfits when the learning rate is 0.01, and underfits when the learning rate is 0.1.

3 Average Perceptron

3.1 Average Perceptron Algorithm

Algorithm 3: Average Perceptron Training Algorithm($D, MaxIter$)

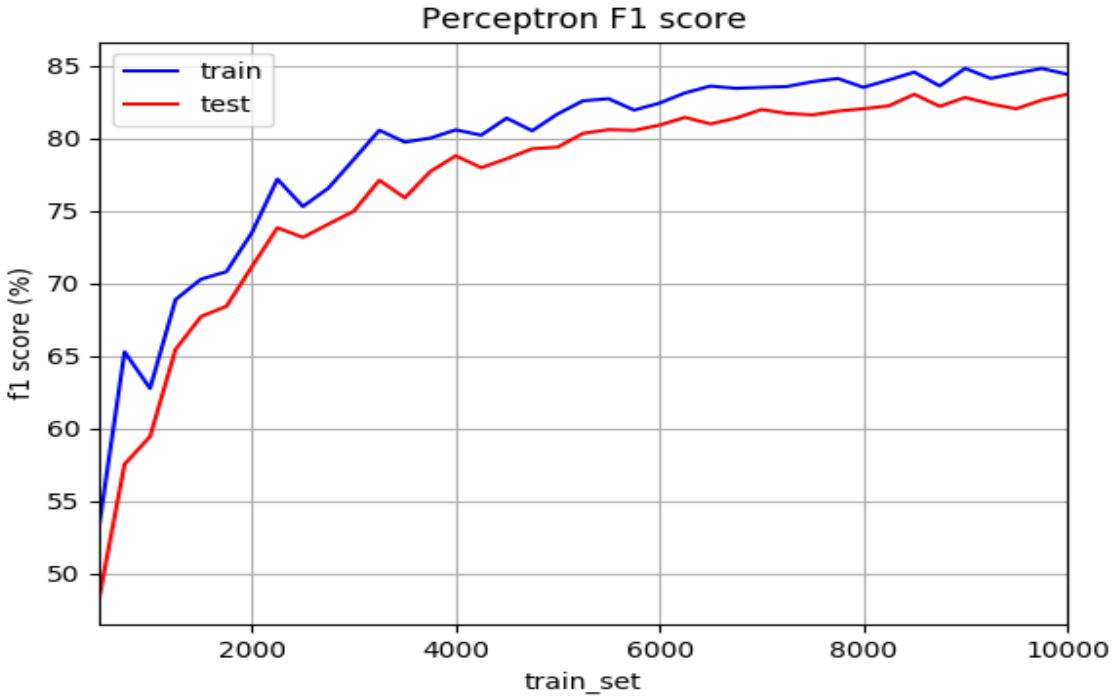
$W \leftarrow [randomInt_1, randomInt_2, ...randomInt_D]$;
$b \leftarrow randomInt$;
$W_{average} \leftarrow$[0 for all d in 1..D] ;
$b_{average} \leftarrow 0$;
for *iter = 1 .. MaxIter* **do**
  for $(x, y) \in D$ **do**
    activation $= W \cdot x + b$;
    if $y \times activation \leq 0$ **then**
      $W \leftarrow W + y \times x$;
      $b \leftarrow b + y$;
      $W_{average} \leftarrow W_{average} + W$;
      $b_{average} \leftarrow b_{average} + b$;
    else
      $W_{average} \leftarrow W_{average} + W$;
      $b_{average} \leftarrow b_{average} + b$;
return $(W_{average}, b_{average})$;

Algorithm 4: Average Perceptron Classify Algorithm($W_{average}, b_{average}, x$)

$activation = W_{average} \cdot x + b_{average}$;
return SIGN($activation$);

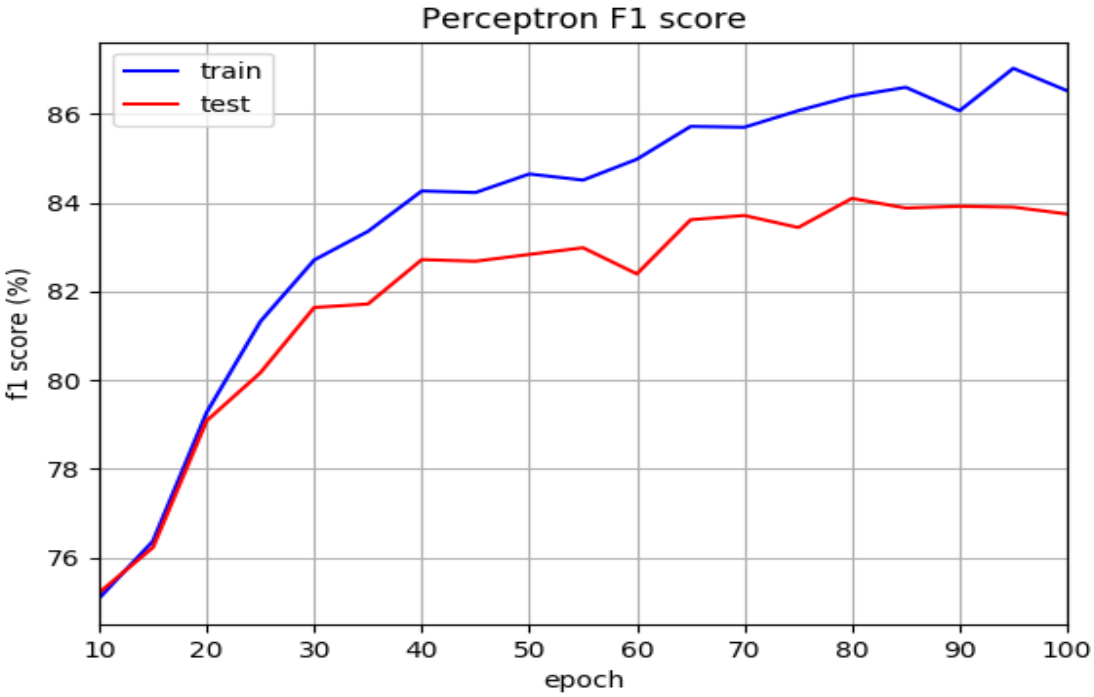
3.2 Effect of size of training set in learning

Vary the size of training set from 500 to 10000 with a step of 250;
set default number of epoch 50 and default learning rate 0.001.



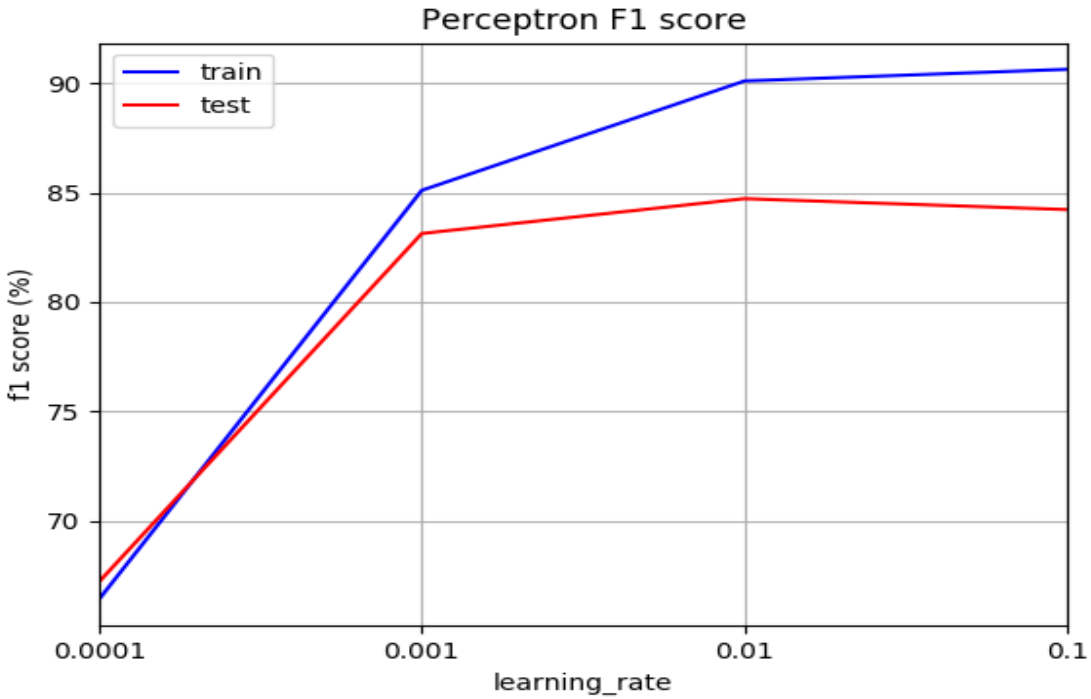
3.3 Effect of number of epoch in learning

Vary the number of epoch from 10 to 100 with step of 5;
set default size of training set 10k and default learning rate 0.001.



3.4 Effect of learning rate in learning

Vary the learning rate for the values 0.0001, 0.001, 0.01, 0.1;
set default number of epoch 50 and default size of training set 10k.



3.5 Observation

When the size of the training set is taken as the hyper-parameter, the f score converged after the size is over 8k.

When epoch is taken as the hyper-parameter, the f score converged after the epoch is over 80.

When the learning rate is taken as the hyper-parameter, the f score increases as the learning rate increases between learning rate=0.0001 and 0.01. But it overfits when the learning rate is 0.1.

4 Winnow

4.1 Winnow Algorithm

Algorithm 5: Winnow Training Algorithm($D, MaxIter, \alpha$)

```
W ←[1 for d in 1..D] ;
for iter = 1..MaxIter do
    for (x, y) ∈ D do
        activation = W · x;
        if y × activation ≤ 0 then
            for i = 1..D do
                if x_i == 1 then
                    W_i ← W_i × y × α ;
        else
            continue ;
return (W) ;
```

Algorithm 6: Winnow Classify Algorithm(W, x)

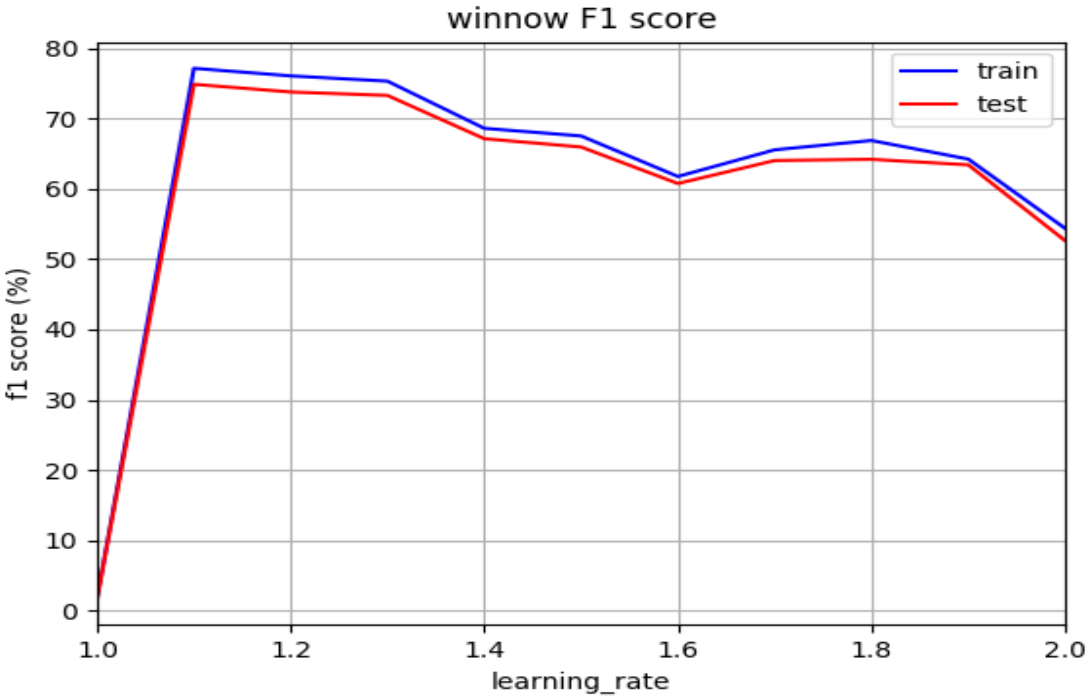
```
activation = W · x ;
return SIGN(activation);
```

4.2 Hyper-parameters

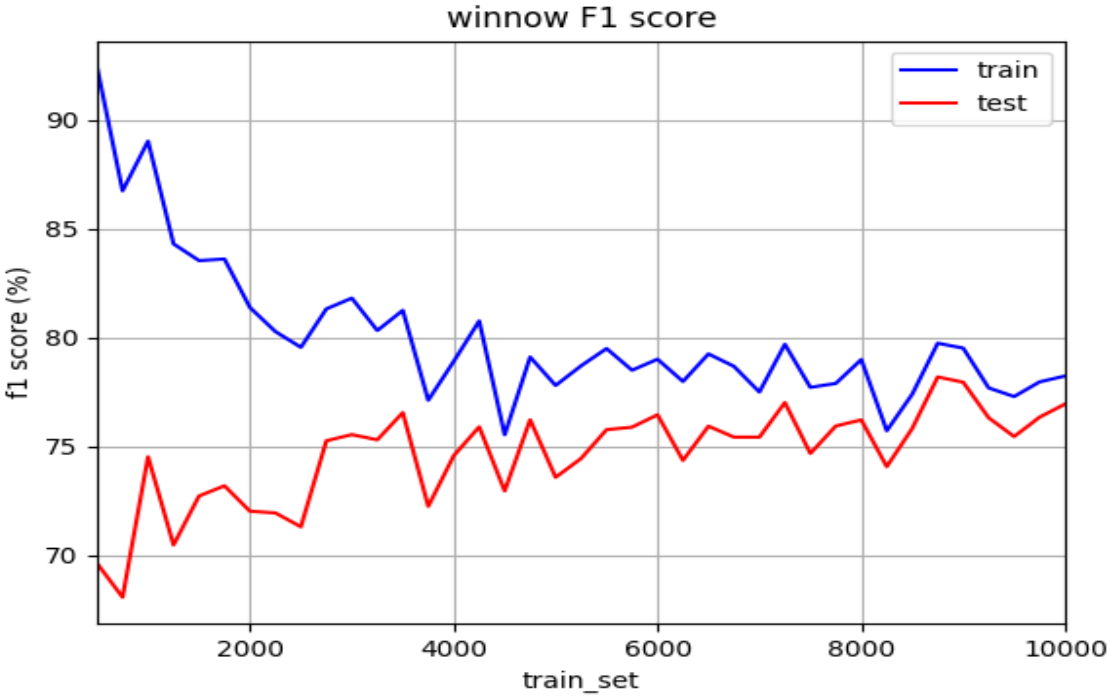
First I pick the multiplication factor $\boldsymbol{\alpha}$ for promotion and denoted as **learning rate**, the hyper-parameter to tune:

Vary the learning rate for the values from 1.0 to 2.0 with step of 0.1;

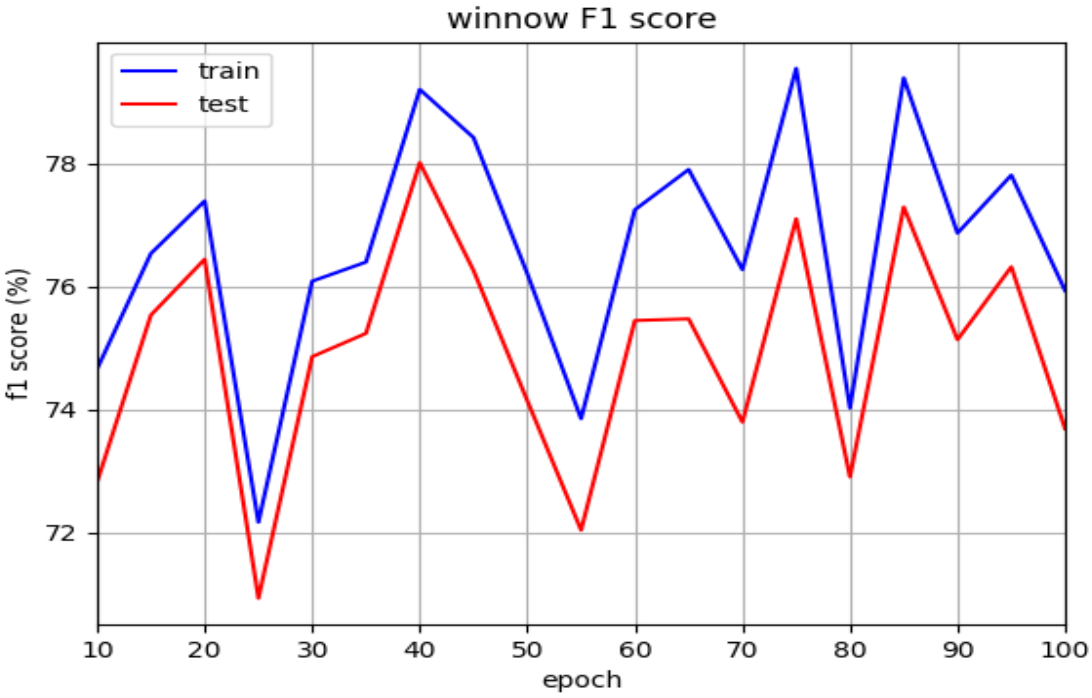
set default number of epoch 50 and default size of training set 10k.



Then pick the size of training set from 500 to 10000 with a step of 250;
set default number of epoch 50 and default learning rate 1.1.



Finally, pick the number of epoch from 10 to 100 with step of 5;
set default size of training set 10k and default learning rate 1.1.



4.3 Observation

When the learning rate is taken as the hyper-parameter, the f score goes down when the learning rate exceeds 1.1, which the weights were updated overly.

When the size of the training set is taken as the hyper-parameter, the f score converged after the size is over 6k.

When epoch is taken as the hyper-parameter, the f score rises as the epoch increases, although the graph didn't display the trend clearly.

5 Open Ended Questions

5.1 Compare Vanilla Perceptron, Average Perceptron and Winnow algorithm based on you experiment above and state your observation.

In vanilla perceptron, each training error is having huge impact, which is more easily to overfit or even underfit.

Average perceptron is averaging all the weights we use during training, which means the training error won't have huge impact and cause overfitting or underfitting. The average perceptron has the best training and testing f1 score among the three classifiers in this assignment.

Winnow algorithm is updating the weights by multiplication and division, which updates the weights faster than perceptron algorithms. Thus the multiplication factor is needed to be modified carefully to get graphs that are easy to analyze.

5.2 Define in one sentence: Mistake Bound (Should include all the components described in class).

Mistake bound limits the number of training errors to a polynomial function of n where n is the dimension of input.

5.3 Suggest a mistake bound algorithm for learning Boolean conjunctions (hint: recall the elimination algorithm for monotone conjunctions). Show that your algorithm is a mistake bound algorithm for Boolean conjunctions.

The elimination algorithm can be adjusted and use as the mistake bound algorithm for learning Boolean conjunctions.

First, initialize a most specific conjunctions including all positive and negative literals, and set the default output to be negative.

During the learning process, we only care about the positive examples, because the default output is set to be negative, which means only positive examples will have the chance to get learning errors.

For each error, we will remove at least one literal from the conjunctions that don't satisfy. After all positive examples are learned trough, we will get the correct Boolean conjunctions that is found by a mistake bound algorithm.

5.4 Given a linearly separable dataset consisting of 1000 positive examples and 1000 negative examples, we train two linear classifier using the perceptron algorithm. We provide the first classifier with a sorted dataset in which all the positive examples appear first, and then the negative examples appear. The second classifier is trained by randomly selecting examples at each training iteration.

5.4.1 Will both classifiers converge?

Both classifiers will converge since the dataset is linearly separable.

5.4.2 What will be the training error of each one of the classifiers?

Both classifiers will have the same mistake bound, but the training time will be different. For the first classifier with a sorted dataset, the value of epoch should be greater than the second classifier. Because the number of errors per epoch will be fewer since all positive examples are sorted together, thus more iteration is needed to reach the correct classifier.