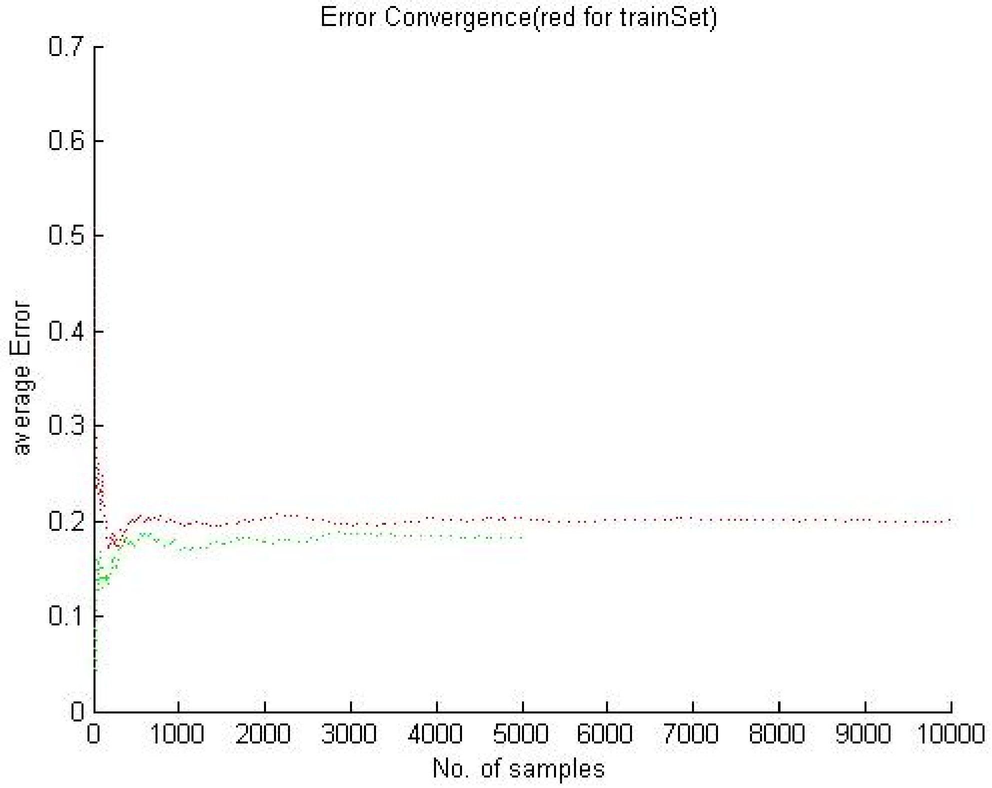
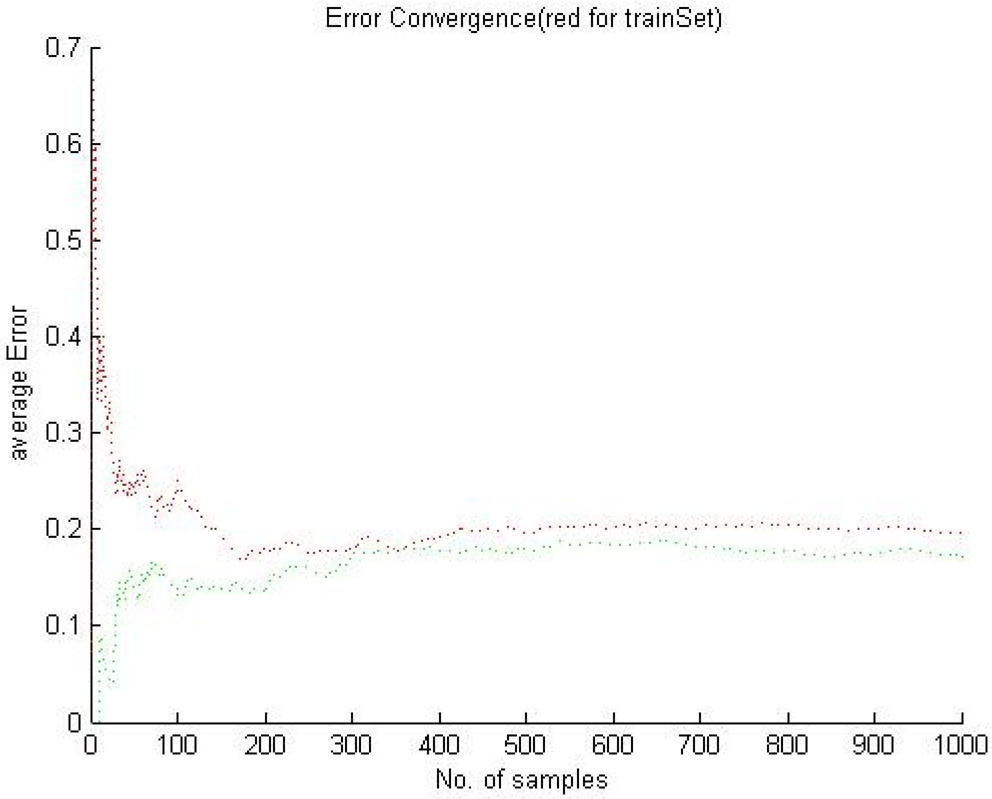
Perceptron Project Document

- Submission Files
    - source code pack(data.mat included)
    - graphs that is used for parameter choosing
- In the experiment, we trained a perceptron over **trainSet** and test its performance with **testSet**, throught which we obtained average errors *v.s.* iteration. Plotting the above two arrays helps in demostrating **the convergence of the algorithm**.Results are presented as followings:

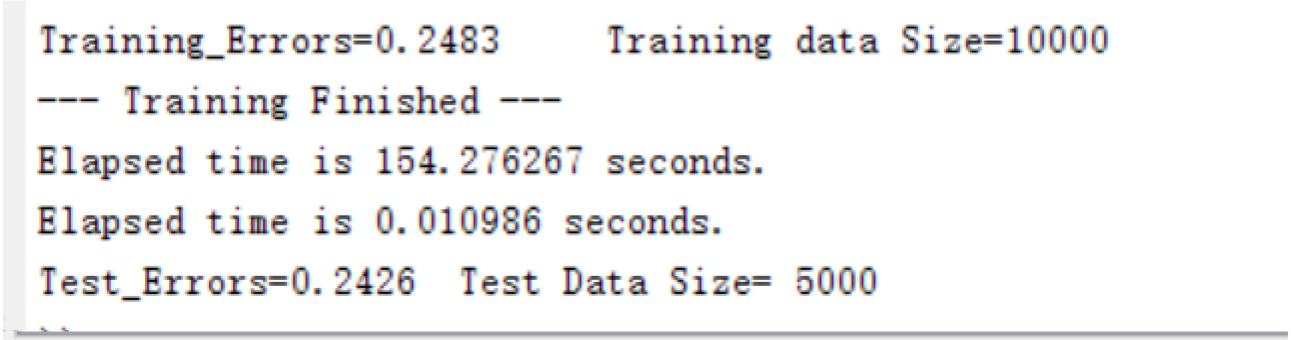
The 'convergence' graph



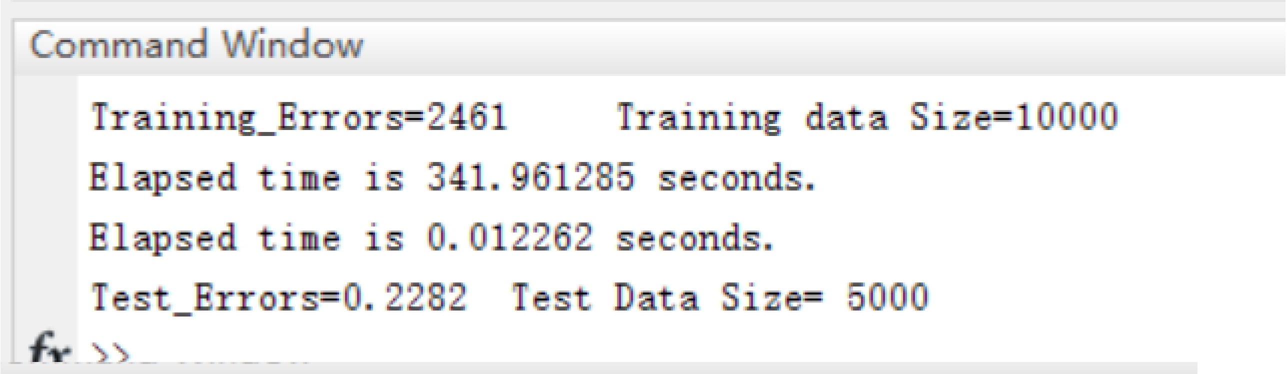
As is shown above, the average error, given by either **trainSet** or **testSet**, is converged to proximately 0.2. Based on the fact that the number of input samples is **finite** (N), it's safe to assert the number of errors is also finite (0.2N).



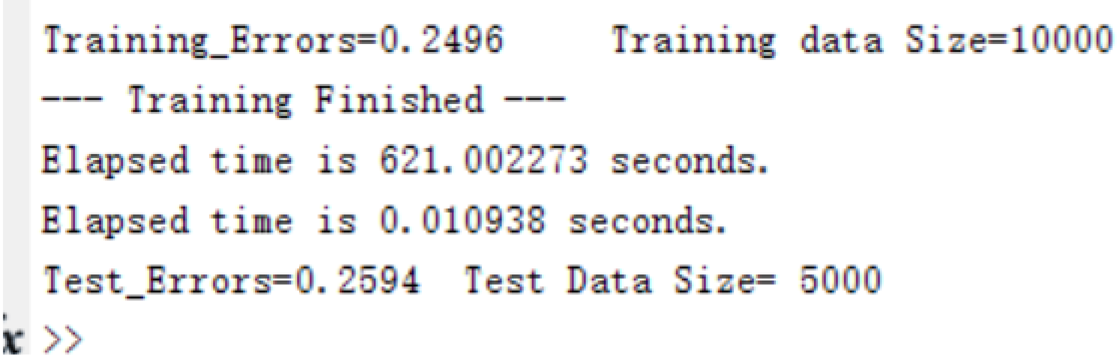
In detail, after K=1000, the curves fluctuate very little. While the two curves aren't decreasing, indicating the trained classifier misclassifies even after we pass all training samples redundantly. Then we choose to pass **the whole** trainSet n times(n=20000, 100000, 5000), figuring out the average errors almost stays the same.



```
Training_Errors=0.2483     Training data Size=10000
--- Training Finished ---
Elapsed time is 154.276267 seconds.
Elapsed time is 0.010986 seconds.
Test_Errors=0.2426  Test Data Size= 5000
```



```
Training_Errors=2461     Training data Size=10000
Elapsed time is 341.961285 seconds.
Elapsed time is 0.012262 seconds.
Test_Errors=0.2282  Test Data Size= 5000
fx >>
```



```
Training_Errors=0.2496     Training data Size=10000
--- Training Finished ---
Elapsed time is 621.002273 seconds.
Elapsed time is 0.010938 seconds.
Test_Errors=0.2594  Test Data Size= 5000
x >>
```

Based on the above obervation and numerous trails, we set *wholeSet pass times* as 20, which brings us acceptable accuracy with tolerable time consumption(the convergence graph is generated as *wholeSet pass times =20*.

The 'Stopping' condtion

The stopping condition can be described as, 'stop' in either two following condtions:

```
+ interation times reaches the threshold(manual);
+ no error detected(model is trained perfectly)
```

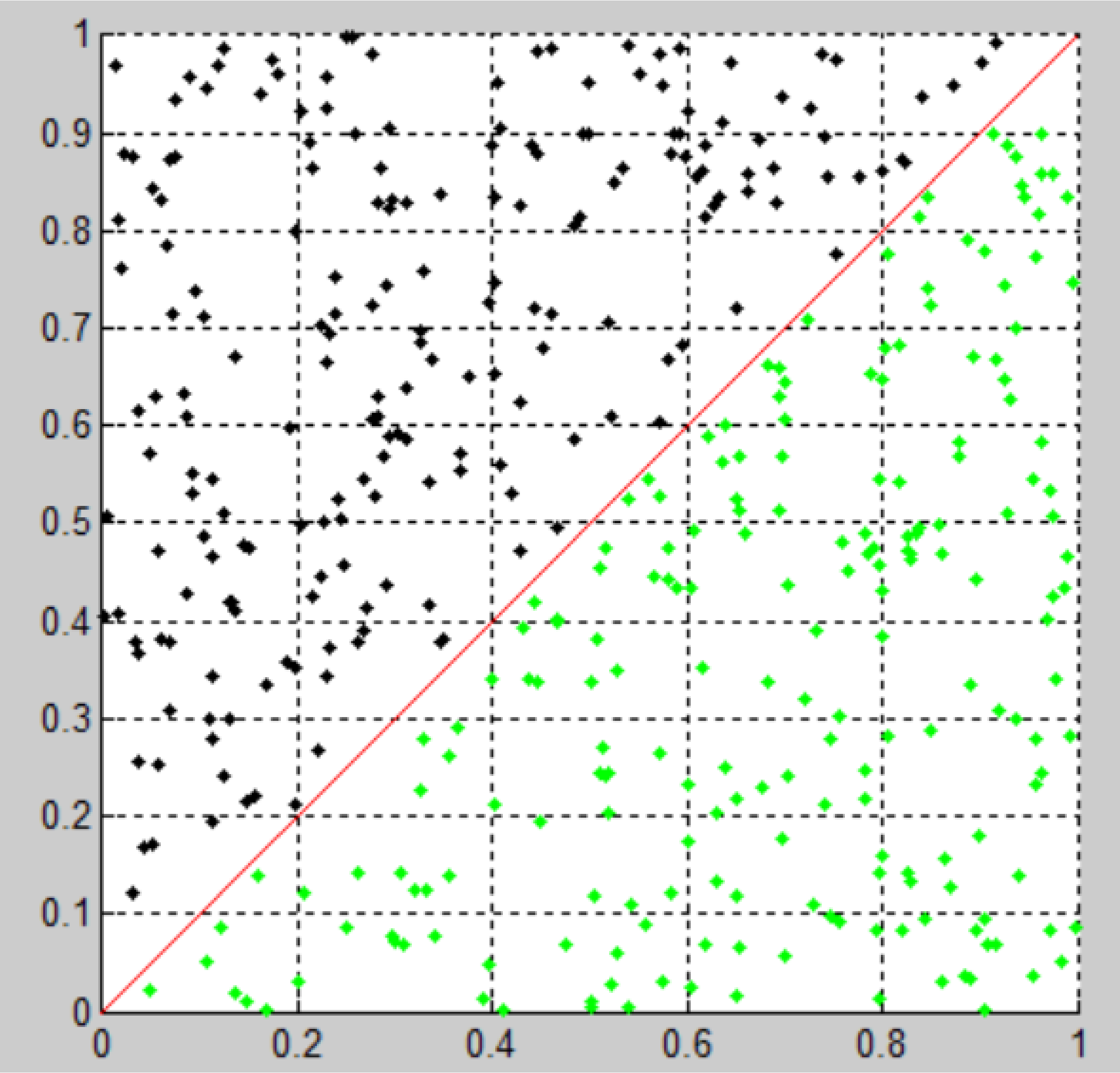
The latter one is given by the definition. The former one is imposed should the sample set itself is **inseparable**.

Evaluation

According the first figure(convergence), it's obivious that the error curve of **testSet** is always lower than that of **trainSet**, in other words, the **trained** classifier makes less mistakes, even faced with samples it has never met. To some extent, it's a success. But our ultimate aim is **almost zero error**, in this case, we presumably assert the dimension of features is **not sufficient for distinguishing phonemes**. Intuitively, it's sometimes hard to distinguish /el/ and /e/.

Under such a condition, we applied the algorithm to another *strictly separable* trainSet, and comes to the following results:

- strictly separable trainSet/testSet generation
  - Graph:

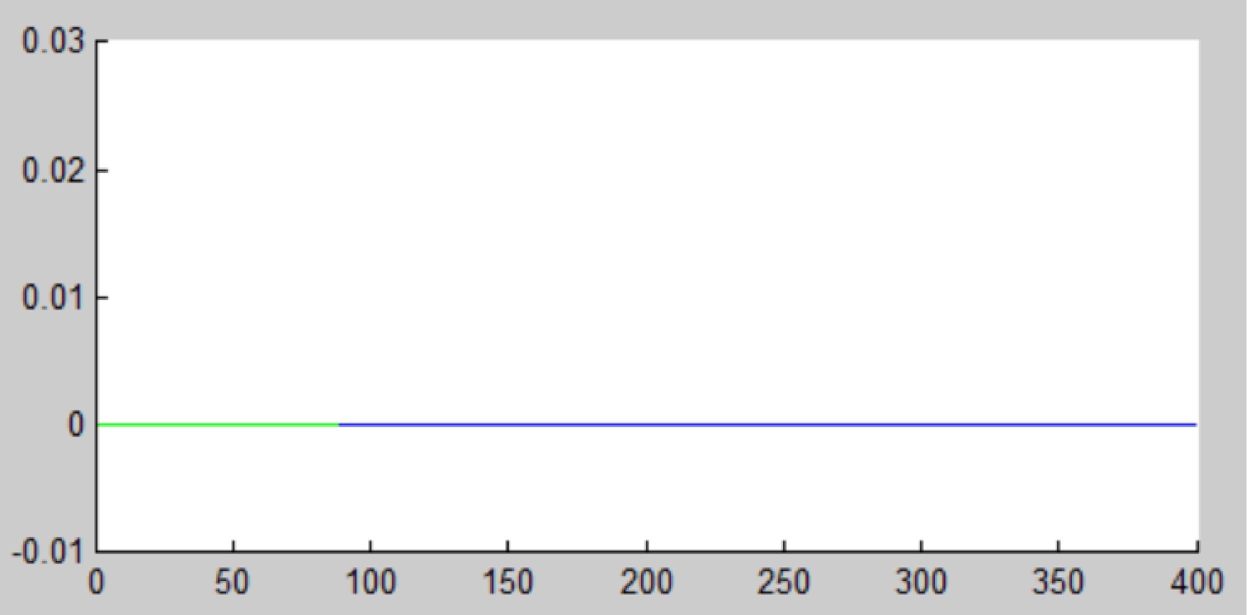


  - Code:

```
mydata = rand(500,2);
acceptindex = abs(mydata(:,1)-mydata(:,2))>0.012;
acceptindex = abs(mydata(:,1)-mydata(:,2))>0.012;
mydata = mydata(acceptindex,:); % data
myclasses = mydata(:,1)>mydata(:,2); % labels
[m n]=size(mydata);
%training data
    x=mydata(1:400,:);   y=myclasses(1:400);
% test data
    xt=mydata(401:m,:); yt=myclasses(401:m);
```

- the convegence graph

  Plotting average error *v.s.* samples, of the **Strictly Separable** trainSet, we obtain the below one, which suggest that **error that cannot converge to zero may be attributed to inseparabilty.**



3. Details & Others
   1. source code package:
      - **the matlab scripts is based on [Ibraheem Al-Dhamari](#) 's version([access](#))**
        - All right reserved as the author requires/required
        - Core ideal remains same;
        - Modifications:
          - bias modification added; (the original $y = w^T x$, ours $y = w^T x + b$)
          - deleting redundant variables and commands;
          - re-organize all functions and re-assign its output
      - pcptron.m <--PerecptronTrn.m / PerecptronTst.m
        - Trn--> train | Tst --> test
      - *.fig/jpg is figures that may be refered to
      - pcptronPrj.md: original mardown doc of the report.