Implementation Assignment2

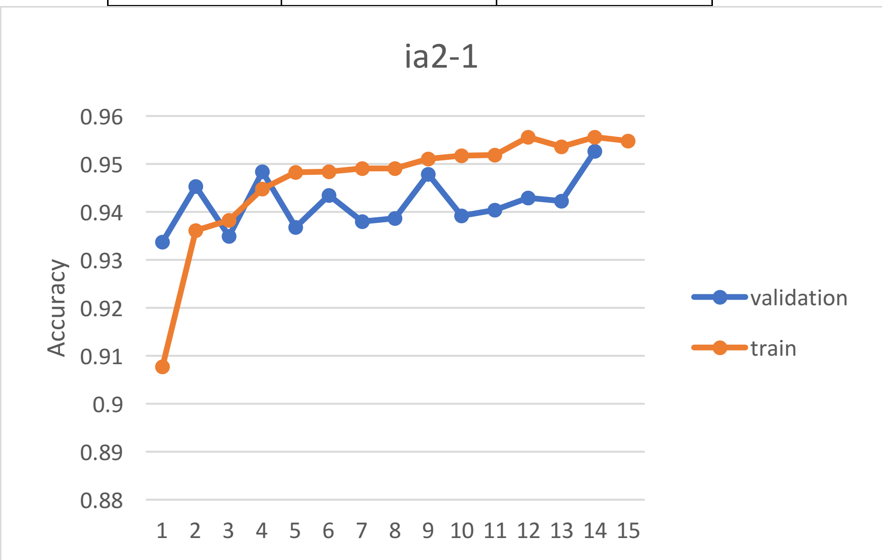
CS534-Fall2018

Chi Wen
Yi-Jung Chiang
Han-Yu Wu

Part 1 (20 pts) : Online Perceptron.

(a) Implement the online perceptron model with algorithm described in Algorithm 1. Set the iters = 15. During the training, at the end of each iteration use the current w to make prediction on the validation samples. Record the accuracies for the train and validation at the end of each iteration. Plot the recorded train and validation accuracies versus the iteration number.

	Accaccuracy	
iteration	Train data	Validation data
1	0.907733224	0.933701657
2	0.936170213	0.945365255
3	0.938216039	0.934929405
4	0.944762684	0.948434622
5	0.948240589	0.936771025
6	0.948445172	0.943523634
7	0.94905892	0.937998772
8	0.94905892	0.938612646
9	0.951104746	0.947820749
10	0.951718494	0.939226519
11	0.951923077	0.940454266
12	0.955605565	0.942909761
13	0.953559738	0.942295887
14	0.955605565	0.952731737
15	0.954787234	0.945979128



(b) Does the train accuracy reach to 100%? Why?

The results we got show that the train accuracy can't reach to 100%. We think the training data sets may not be linear separable. Thus, the accuracy can't be perfect.

(c) Use the validation accuracy to decide the test number for iters. Apply the resulting model to make predictions for the samples in the test set. Generate the

prediction le oplabel.csv. Please note that your le should only contain +1 (for 3) and -1 (for 5) and the number of rows should be the same as pa2 test.csv.

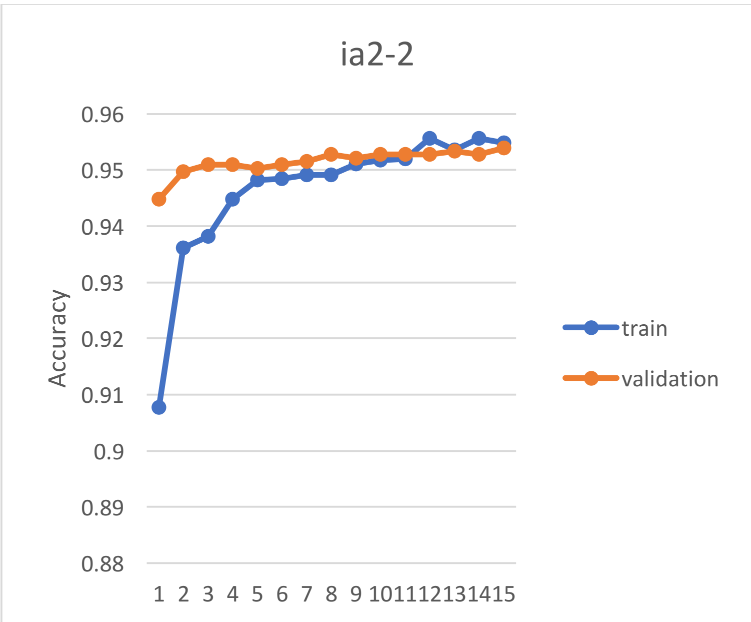
We choose the number 14 iteration results to predict the test data.

Part 2 (20 pts). Average Perceptron.

(a) Please implement the average perceptron described in Algorithm 2.

(b) Plot the train and validation accuracies versus the iteration number for iters = 1... 15.

	Accuracy	
iteration	Train data	Validation data
1	0.907733224	0.944751381
2	0.936170213	0.94966237
3	0.938216039	0.950890117
4	0.944762684	0.950890117
5	0.948240589	0.950276243
6	0.948445172	0.950890117
7	0.94905892	0.95150399
8	0.94905892	0.952731737
9	0.951104746	0.952117864
10	0.951718494	0.952731737
11	0.951923077	0.952731737
12	0.955605565	0.952731737
13	0.953559738	0.953345611
14	0.955605565	0.952731737
15	0.954787234	0.953959484



(c) **How average model has a affected the validation accuracy comparing to the online perceptron?**

We observed that the average perceptron could achieve higher accuracy level and more stable than online perceptron in part1.

Part3 (40 pts). Polynomial Kernel Perceptron.

(a) **Implement the polynomial kernel function kp in the Algorithm 3. This function takes two vectors x1 and x2 and an integer p for the polynomial degree, and returns a real value.**

We implement in the function Kernel_Perceptron_Train which is in the IA2-part3.py file.

(b) **Implement the polynomial kernel function kp in the Algorithm 3. This function takes two vectors x1 and x2 and an integer p for the polynomial degree, and returns a real value.**

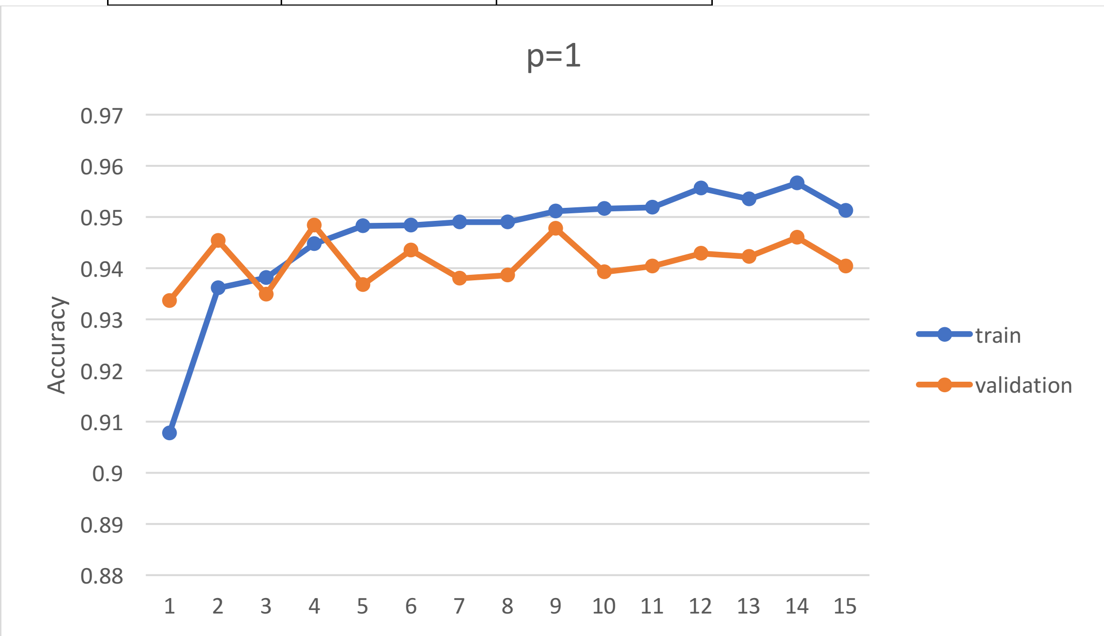
We implement in the function Kernel_Perceptron_Train which is in the IA2-part3.py file

(c) **Implement the rest of the kernel perceptron in Algorithm 3. For each p in [1, 2, 3, 7, 15]:**

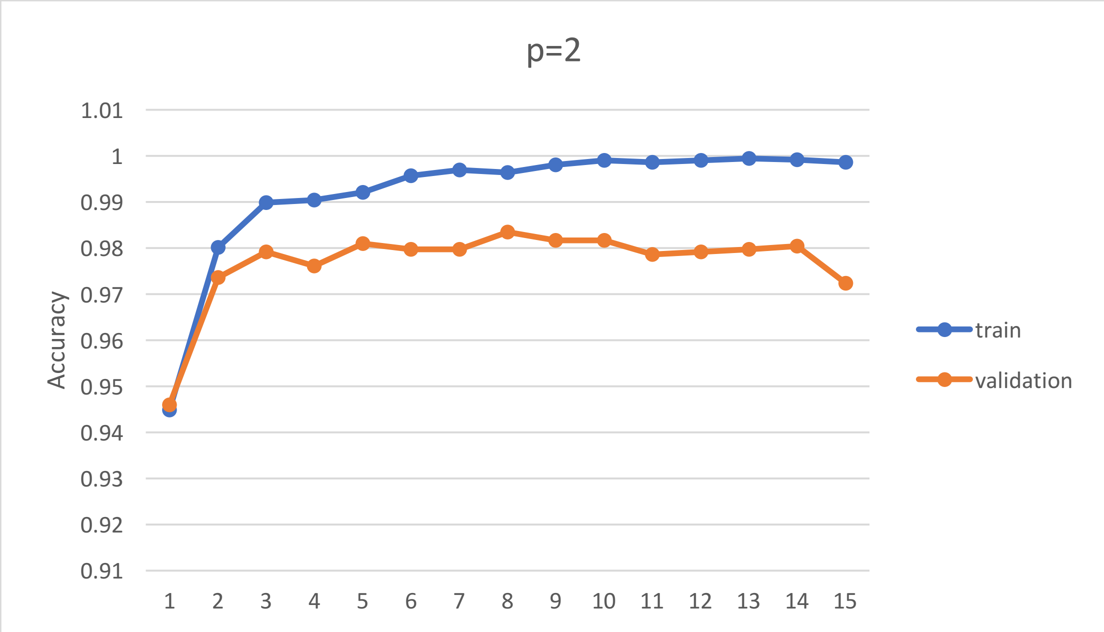
1) **Run the algorithm to compute.**
2) **At the end of each iteration use the current to predict validation set.**
3) **Record the train and validation accuracy for each iteration and plot the train and validation accu- racies versus the iteration number.**

p=1	Accuracy	
iteration	train	validation
1	0.907733224	0.933701657
2	0.936170213	0.945365255
3	0.938216039	0.934929405
4	0.944762684	0.948434622
5	0.948240589	0.936771025
6	0.948445172	0.943523634
7	0.94905892	0.937998772
8	0.94905892	0.938612646
9	0.951104746	0.947820749
10	0.951718494	0.939226519
11	0.951923077	0.940454266
12	0.955605565	0.942909761
13	0.953559738	0.942295887

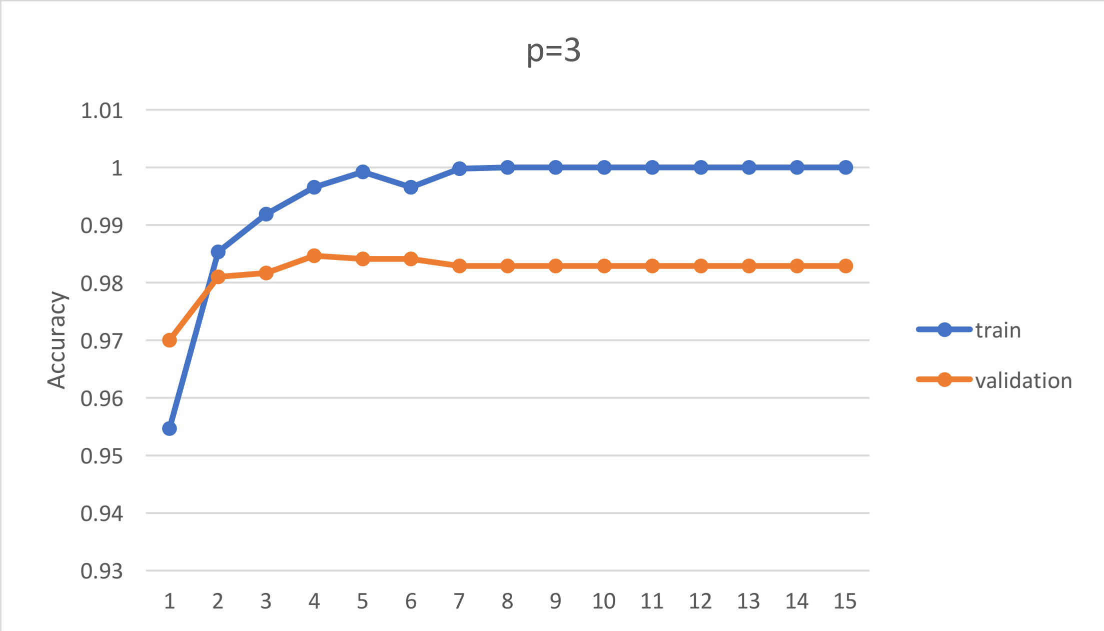
14	0.956628478	0.945979128
15	0.951309329	0.940454266



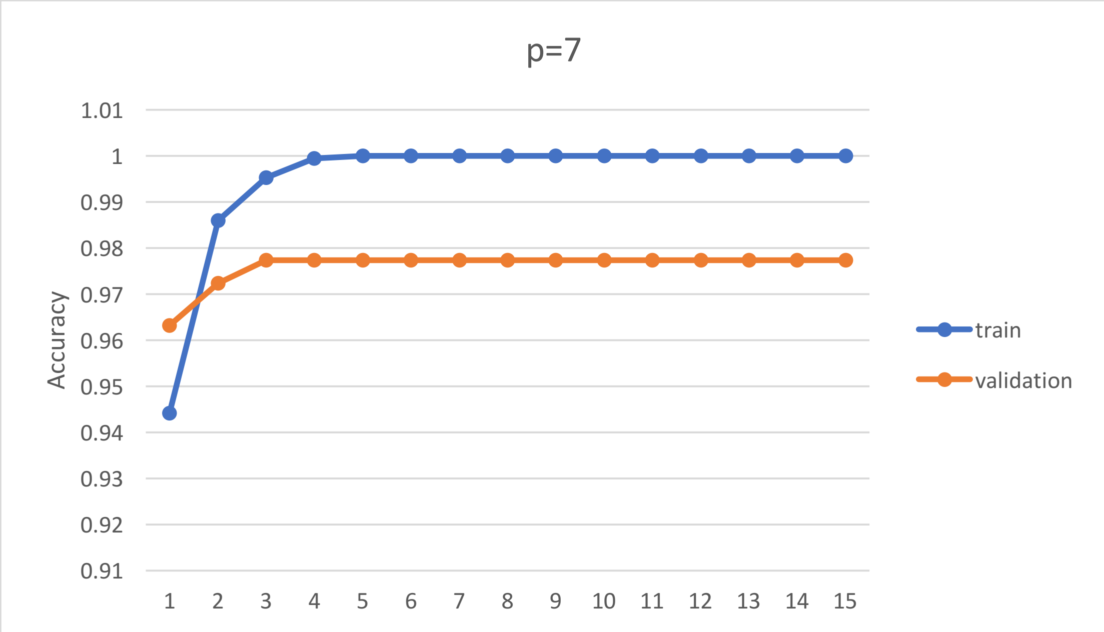
p=2	Accuracy	
iteration	train	validation
1	0.944762684	0.945979128
2	0.980155483	0.973603438
3	0.989770867	0.9791283
4	0.990384615	0.976058932
5	0.992021277	0.98096992
6	0.995703764	0.979742173
7	0.99693126	0.979742173
8	0.996317512	0.983425414
9	0.997954173	0.981583794
10	0.998977087	0.981583794
11	0.998567921	0.978514426
12	0.998977087	0.9791283
13	0.999386252	0.979742173
14	0.999181669	0.980356047
15	0.998567921	0.972375691



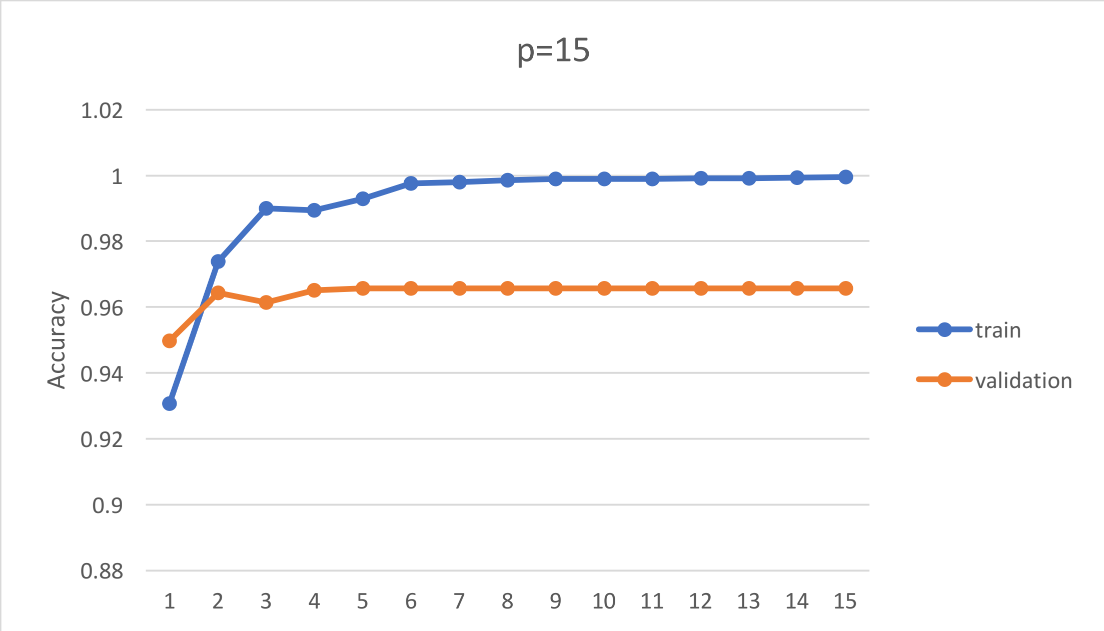
p=3	Accuracy	
iteration	train	validation
1	0.954582651	0.969920196
2	0.985270049	0.98096992
3	0.991816694	0.981583794
4	0.996522095	0.984653161
5	0.999181669	0.984039288
6	0.996522095	0.984039288
7	0.999795417	0.982811541
8	1	0.982811541
9	1	0.982811541
10	1	0.982811541
11	1	0.982811541
12	1	0.982811541
13	1	0.982811541
14	1	0.982811541
15	1	0.982811541



p=7	Accuracy	
iteration	train	validation
1	0.944148936	0.963167587
2	0.985883797	0.972375691
3	0.995294599	0.977286679
4	0.999386252	0.977286679
5	1	0.977286679
6	1	0.977286679
7	1	0.977286679
8	1	0.977286679
9	1	0.977286679
10	1	0.977286679
11	1	0.977286679
12	1	0.977286679
13	1	0.977286679
14	1	0.977286679
15	1	0.977286679



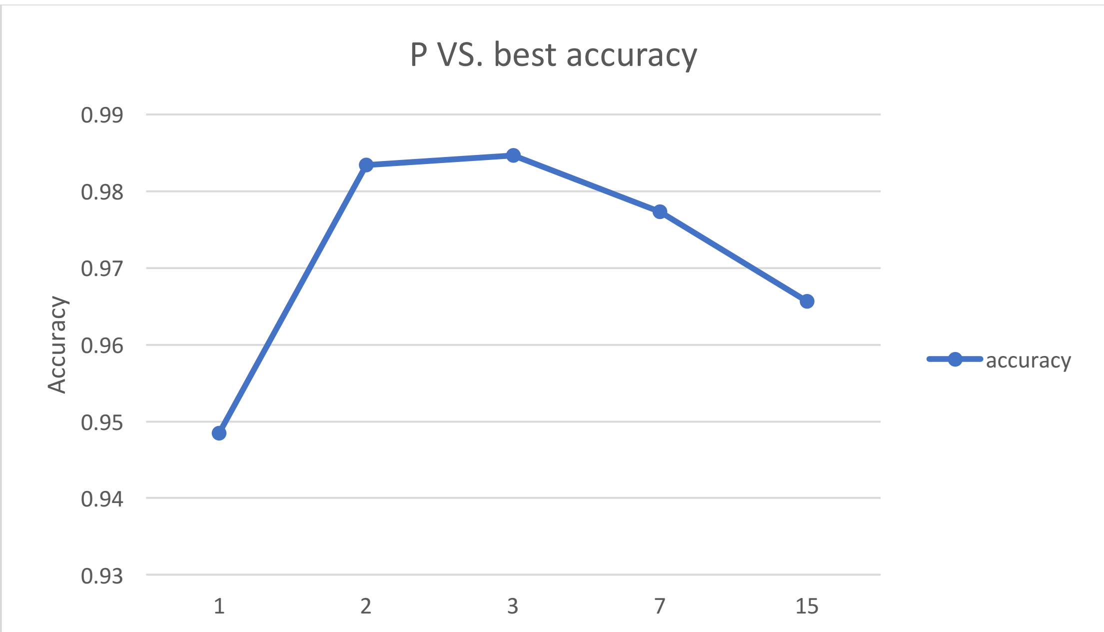
p=15	Accuracy	
iteration	train	validation
1	0.930646481	0.94966237
2	0.973813421	0.964395335
3	0.98997545	0.961325967
4	0.989361702	0.965009208
5	0.992839607	0.965623082
6	0.997545008	0.965623082
7	0.997954173	0.965623082
8	0.998567921	0.965623082
9	0.998977087	0.965623082
10	0.998977087	0.965623082
11	0.998977087	0.965623082
12	0.999181669	0.965623082
13	0.999181669	0.965623082
14	0.999386252	0.965623082
15	0.999590835	0.965623082



4) Record the best validation accuracy achieved for each p over all iterations.

p	Best accuracy
1	0.948434622
2	0.983425414
3	0.984653161
7	0.977286679
15	0.965623082

(d) Plot the recorded best validation accuracies versus degrees. Please explain how p is a ecting the train and validation performance.



The value p affects the training speed and predict accuracy. If p value is going bigger the convergence speed will be more quickly and reach accuracy=1. However, the p value large enough the speed and accuracy stead going down.

(e) Use your best (the best you found over all d and iterations above) to predict the test data-set. Please name the predicted le as kplabel.csv.

The file attached in the zip file