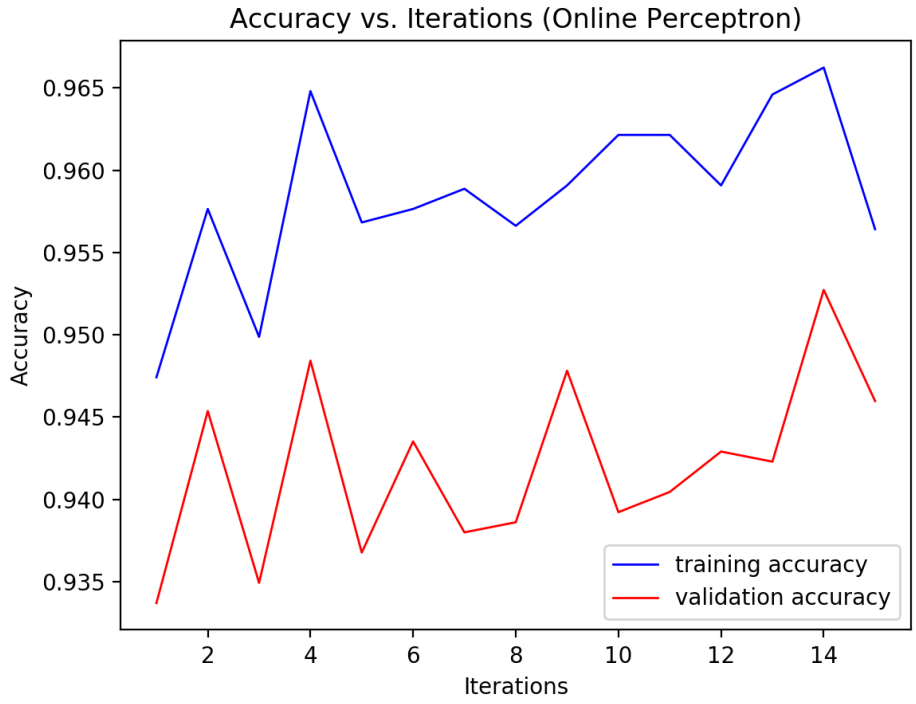
CS534 - Implementation Assignment 2

Deqing Qu (qud@oregonstate.edu)

Part 1: Online Perceptron

(a)

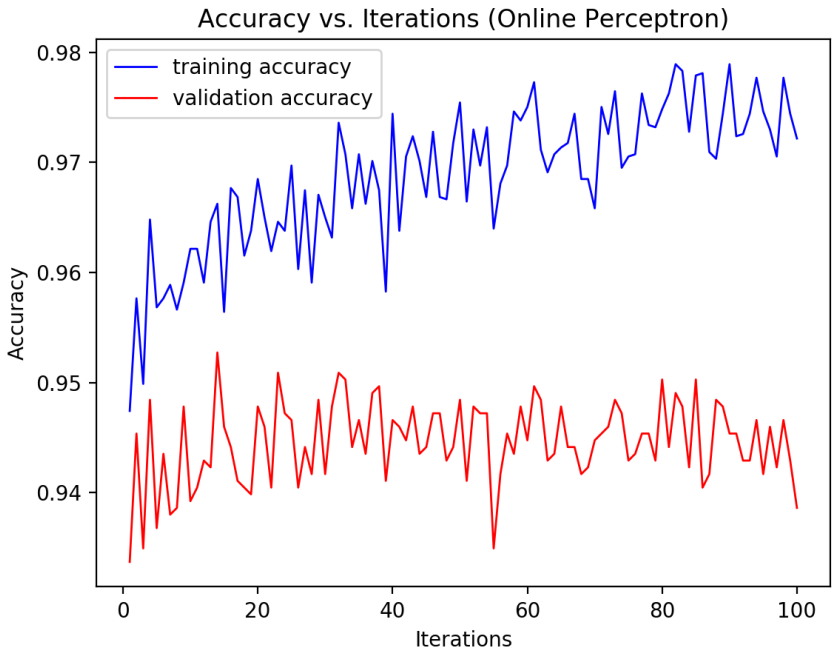
iteration number	accuracy on the training set	accuracy on the validation set
1	0.947422	0.933702
2	0.957651	0.945365
3	0.949877	0.934929
4	0.964812	0.948435
5	0.956833	0.936771
6	0.957651	0.943524
7	0.958879	0.937999
8	0.956628	0.938613
9	0.959083	0.947821
10	0.962152	0.939227
11	0.962152	0.940454
12	0.959083	0.942910
13	0.964607	0.942296
14	0.966244	0.952732
15	0.956424	0.945979



The accuracy on the training set is fluctuated between 0.945 and 0.965. The accuracy on the validation set is fluctuated between 0.93 and 0.955. The accuracy on the validation set is always lower than the accuracy on the training set.

(b)
The train accuracy doesn't reach 100%. Because the OCR problem is not linear and the online perceptron is a linear classification model, the model can't fit the problem so well and the accuracy can't reach 100%.

(c)
As the iteration number increases, the validation accuracy doesn't improve obviously. When the iteration number is 14, the validation accuracy reaches the maximum value of 0.952732. So I chose 14 as the iteration number to make predictions for the samples on the test set.

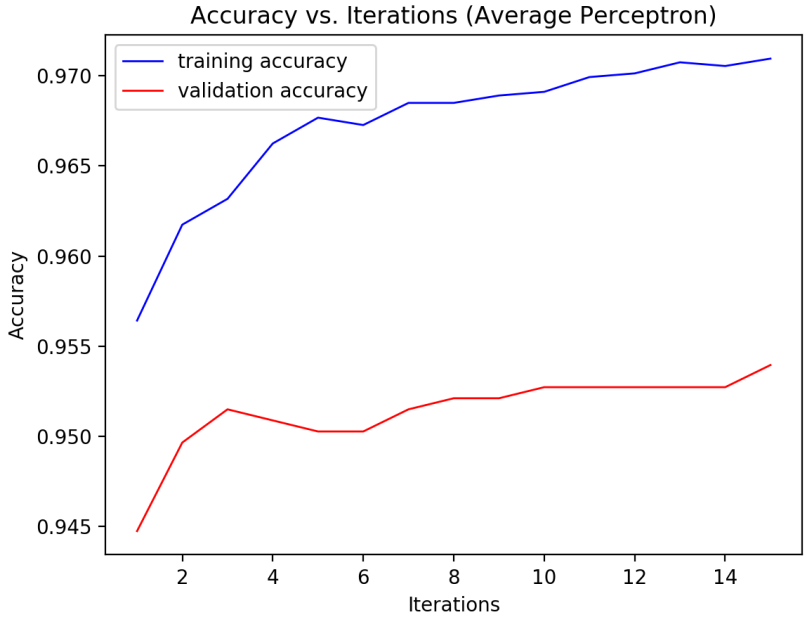


Part 2: Average Perceptron
(a)
Please see the average_perceptron method in main.py

(b)
For the average perceptron, the training accuracy improves as the iteration number increases.

iteration number	accuracy on the training set	accuracy on the validation set
1	0.956424	0. 944751
2	0.961743	0.949662
3	0.963175	0.951504
4	0.966244	0.950890

5	0.967676	0.950276
6	0.967267	0.950276
7	0.968494	0.951504
8	0.968494	0.952118
9	0.968903	0.952118
10	0.969108	0.952732
11	0.969926	0.952732
12	0.970131	0.952732
13	0.970745	0.952732
14	0.970540	0.952732
15	0.970949	0.953959



(c)
Comparing to the online perceptron, the validation accuracy of the average perceptron improves as the iteration number increases. Moreover, the validation accuracy of the average perceptron is smoother than the online perceptron. For the average perceptron, the prediction is relied on the averaged weights, so the process of updating weights is smoother than the online perceptron.

Part 3: Polynomial Kernel Perceptron
(a, b)
The vector version of the kernel function shows as follows:

```
def kernel_function(x, y, p):
    return (1 + np.dot(x, y)) ** p
```

However, it will be very slow as there are 4,888 vectors in the training set. The matrix multiplication is applied to speed up the kernel function. The matrix version of the kernel function shows as follows:

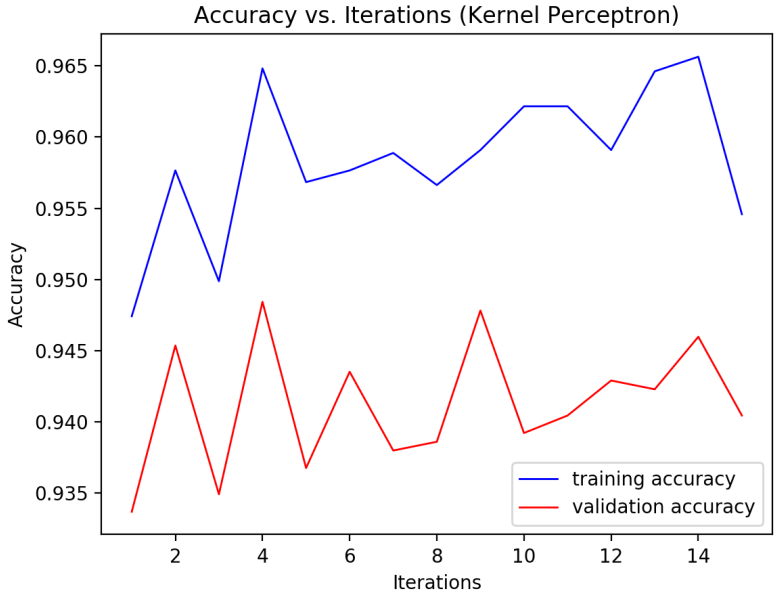
```
def kernel_function(x, y, p):
    return np.power((np.matmul(x, y.T) + 1), p)
```

The parameter x, y are matrices and the return value of the function is a matrix of real values (the gram matrix).

(c)

P = 1

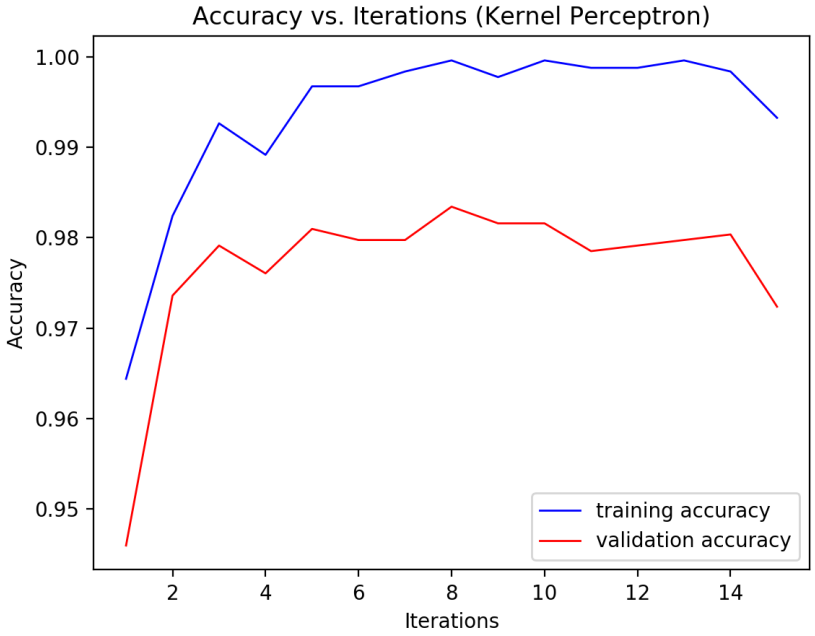
iteration number	accuracy on the training set	accuracy on the validation set
1	0.947422	0.933702
2	0.957651	0.945365
3	0.949877	0.934929
4	0.964812	0.948435
5	0.956833	0.936771
6	0.957651	0.943524
7	0.958879	0.937999
8	0.956628	0.938613
9	0.959083	0.947821
10	0.962152	0.939227
11	0.962152	0.940454
12	0.959083	0.94291
13	0.964607	0.942296
14	0.96563	0.945979
15	0.954583	0.940454



We can see that the degree P = 1 doesn't work so well. The accuracy on the training set and the validation set are both fluctuated. Because the problem is not linear, the linear model can't fit it well.

P = 2

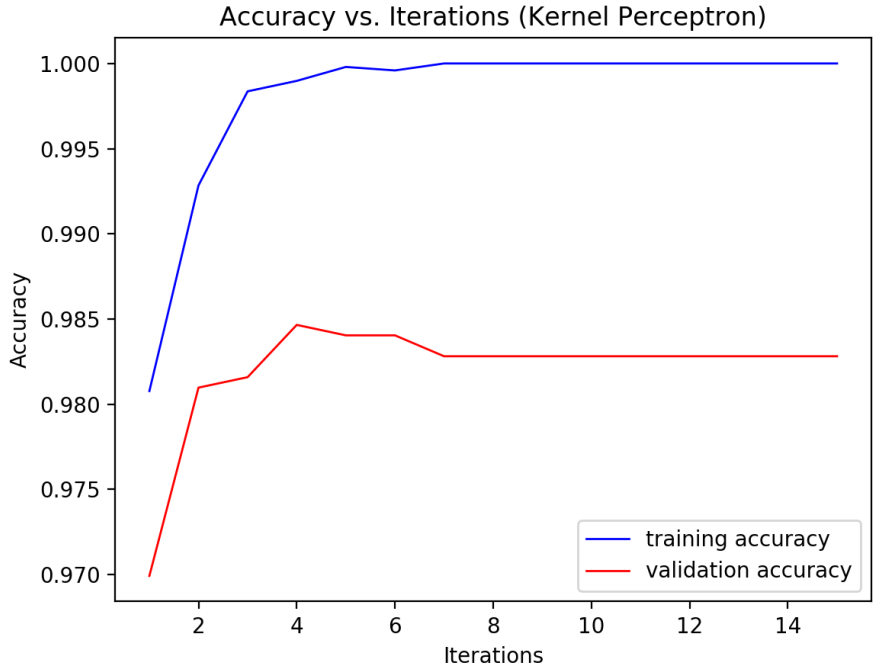
iteration number	accuracy on the training set	accuracy on the validation set
1	0.964403	0.945979
2	0.982406	0.973603
3	0.992635	0.979128
4	0.989157	0.976059
5	0.996727	0.980970
6	0.996727	0.979742
7	0.998363	0.979742
8	0.999591	0.983425
9	0.997750	0.981584
10	0.999591	0.981584
11	0.998773	0.978514
12	0.998773	0.979128
13	0.999591	0.979742
14	0.998363	0.980356
15	0.993249	0.972376



We can see that the degree P = 2 works better than P = 1. The quadratic model can fit the non-linear problem better than the linear model. The accuracy on the training set almost reaches 1.0 and the best accuracy on the validation set is more than 0.98.

P = 3

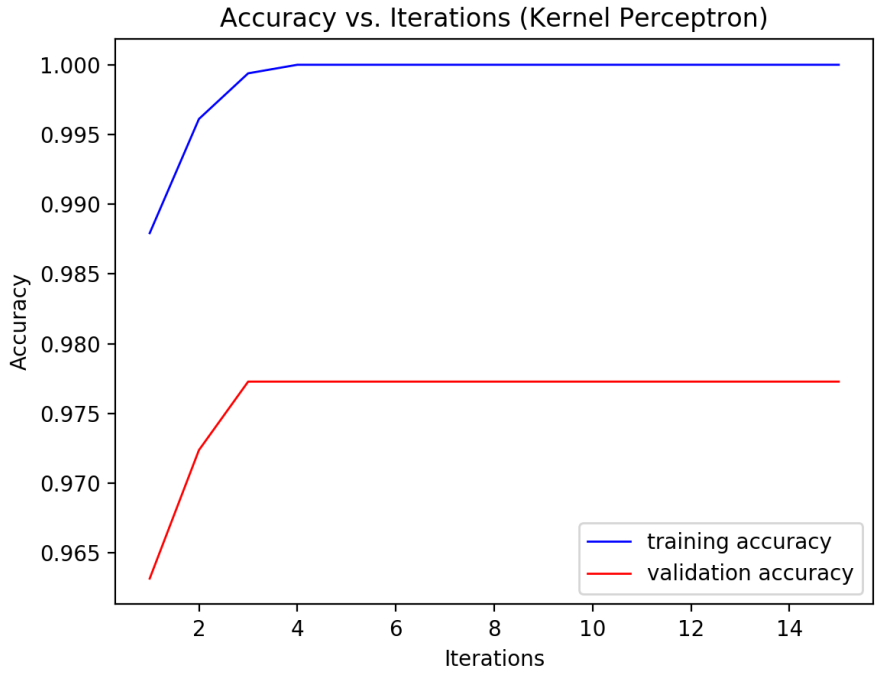
iteration number	accuracy on the training set	accuracy on the validation set
1	0.980769	0.969920
2	0.992840	0.980970
3	0.998363	0.981584
4	0.998977	0.984653
5	0.999795	0.984039
6	0.999591	0.984039
7	1.000000	0.982812
8	1.000000	0.982812
9	1.000000	0.982812
10	1.000000	0.982812
11	1.000000	0.982812
12	1.000000	0.982812
13	1.000000	0.982812
14	1.000000	0.982812
15	1.000000	0.982812



When the degree is set to 3, the accuracy on the training set can reach 1.0 and the best accuracy on the validation set is more than 0.985. In all degrees we tried, P = 3 reaches the highest accuracy on the validation set. As the accuracy on the training set reaches 100%, the model can't be optimized anymore, so the accuracy on validation set can't be improved.

P = 7

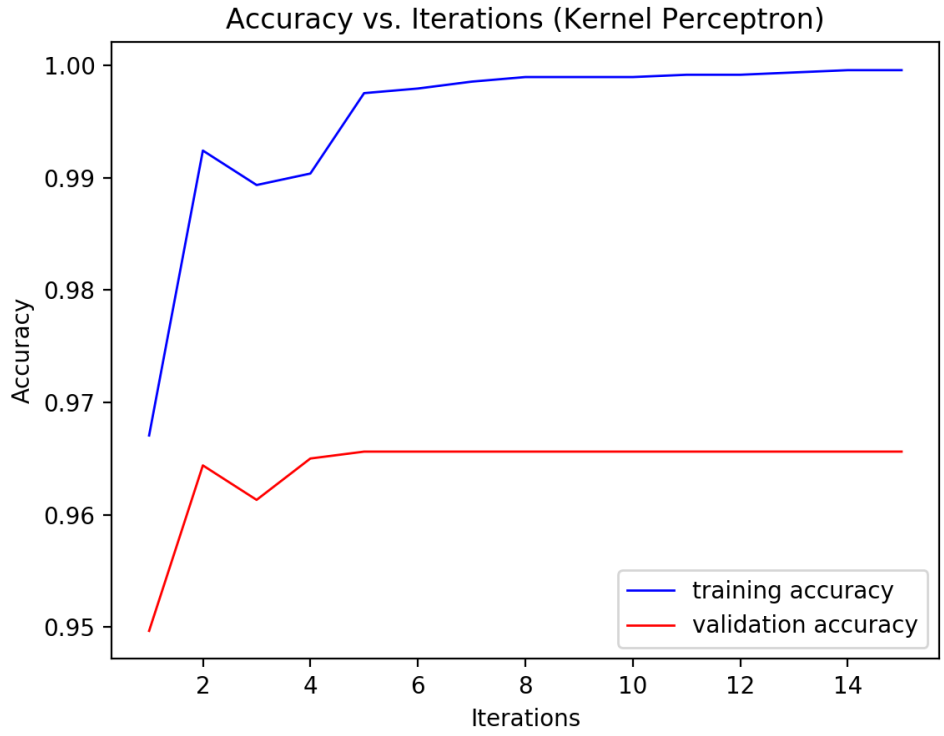
iteration number	accuracy on the training set	accuracy on the validation set
1	0.987930	0.963168
2	0.996113	0.972376
3	0.999386	0.977287
4	1.000000	0.977287
5	1.000000	0.977287
6	1.000000	0.977287
7	1.000000	0.977287
8	1.000000	0.977287
9	1.000000	0.977287
10	1.000000	0.977287
11	1.000000	0.977287
12	1.000000	0.977287
13	1.000000	0.977287
14	1.000000	0.977287
15	1.000000	0.977287



When the degree is set to 7, the accuracy on the training set reaches 1.0 at the 4th iteration and the accuracy on the validation set can't improve anymore. It means that the prediction model with P = 7 tends to overfit on the problem.

P = 15

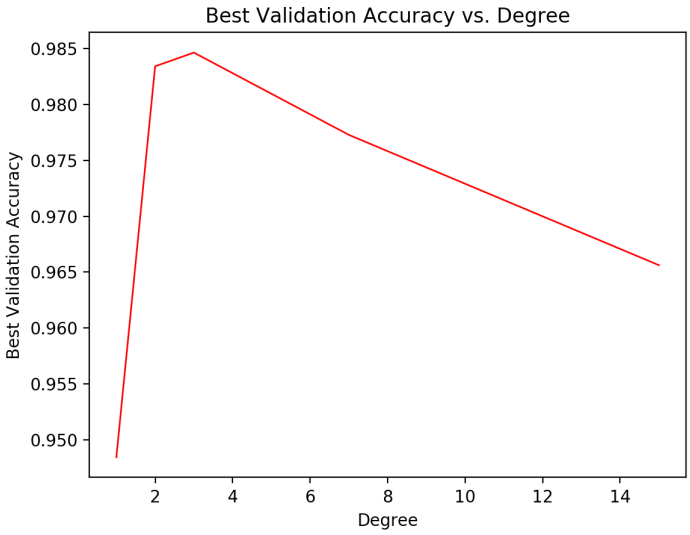
iteration number	accuracy on the training set	accuracy on the validation set
1	0.967062	0.949662
2	0.992430	0.964395
3	0.989362	0.961326
4	0.990385	0.965009
5	0.997545	0.965623
6	0.997954	0.965623
7	0.998568	0.965623
8	0.998977	0.965623
9	0.998977	0.965623
10	0.998977	0.965623
11	0.999182	0.965623
12	0.999182	0.965623
13	0.999386	0.965623
14	0.999591	0.965623
15	0.999591	0.965623



When the degree is set to 15, the accuracy on the training set almost reach 1.0 and the accuracy on the validation set can reach more than 0.965. It means that the degree P = 15 is too high and the prediction model tends to overfit on the problem.

(d)

Degree	the best validation accuracy
1	0.948435
2	0.983425
3	0.984653
7	0.977287
15	0.965623



From the recorded data, we can achieve the best validation accuracy if we choose the degree as 3. We can find that if we choose the degree as 1, the model is linear and can't fit the problem well. The validation accuracy has a great improvement when we increase the degree to 2 and 3. It means the degree P = 3 fits the problem best. When we continue to increase the degree, the validation accuracy starts to drop, since the larger degrees tend to overfit the problem.

(e) The degree P = 3 and the iterations equals to 4 is chose to predict the test data set.