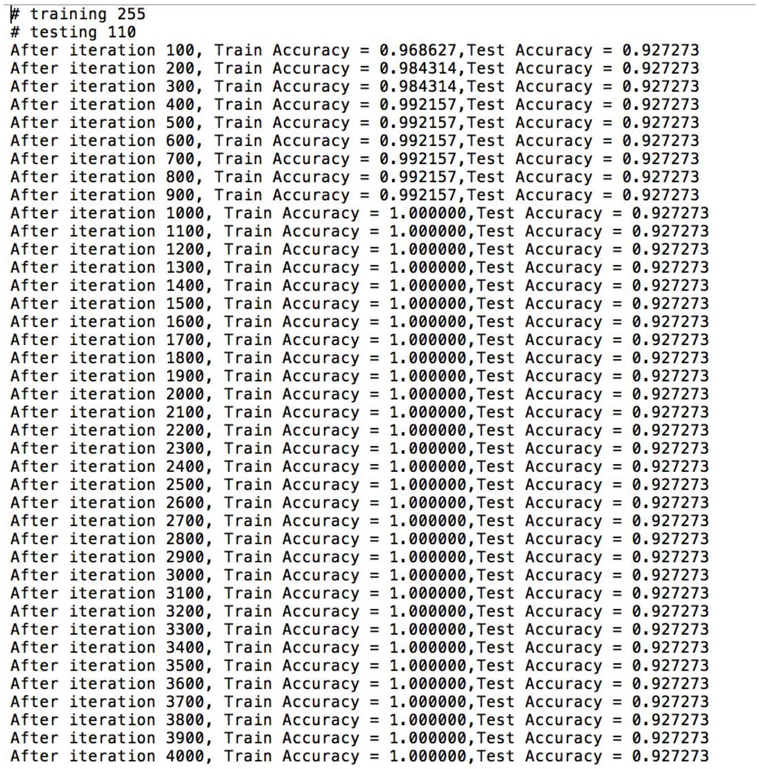
Comaprison between Neural Networks and AdaBoost /SVM Results

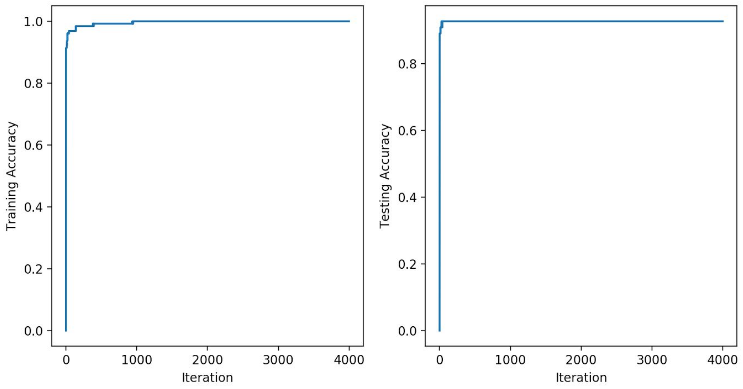
Data used: digits.csv (Training: 70%; Testing: 30%)
Programming Language: Python

1. **AdaBoost Results** (Training and Testing Accuracy):

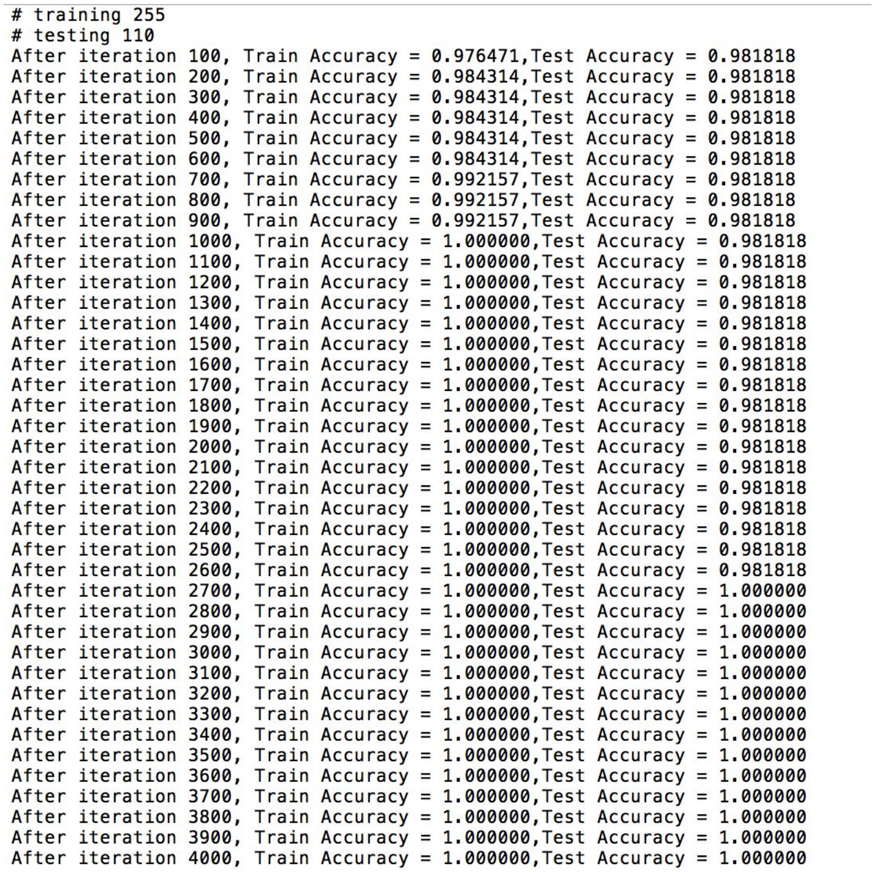


```
# training 255
# testing 110
After iteration 100, Train Accuracy = 0.968627,Test Accuracy = 0.927273
After iteration 200, Train Accuracy = 0.984314,Test Accuracy = 0.927273
After iteration 300, Train Accuracy = 0.984314,Test Accuracy = 0.927273
After iteration 400, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 500, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 600, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 700, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 800, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 900, Train Accuracy = 0.992157,Test Accuracy = 0.927273
After iteration 1000, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1100, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1200, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1300, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1400, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1500, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1600, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1700, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1800, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 1900, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2000, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2100, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2200, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2300, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2400, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2500, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2600, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2700, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2800, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 2900, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3000, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3100, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3200, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3300, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3400, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3500, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3600, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3700, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3800, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 3900, Train Accuracy = 1.000000,Test Accuracy = 0.927273
After iteration 4000, Train Accuracy = 1.000000,Test Accuracy = 0.927273
```

2. **AdaBoost Plots** (Training Accuracy and Testing Accuracy V/s. Iteration):

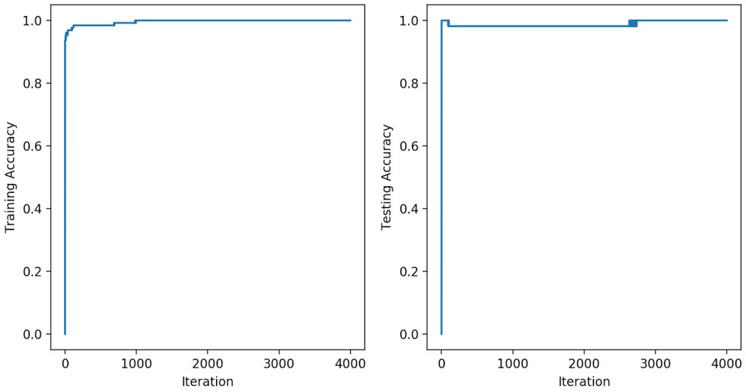


3. **SVM results**(Training and Testing Accuracy)**:**

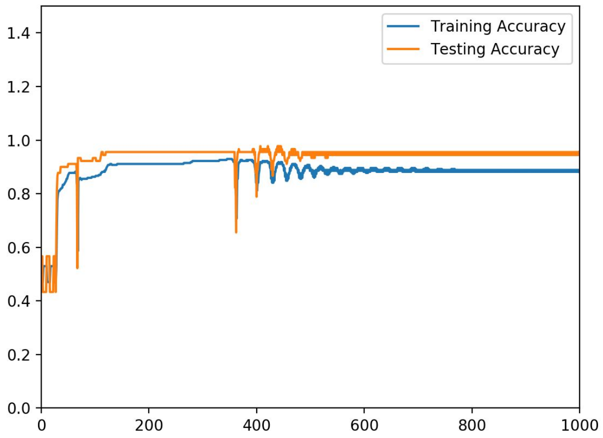


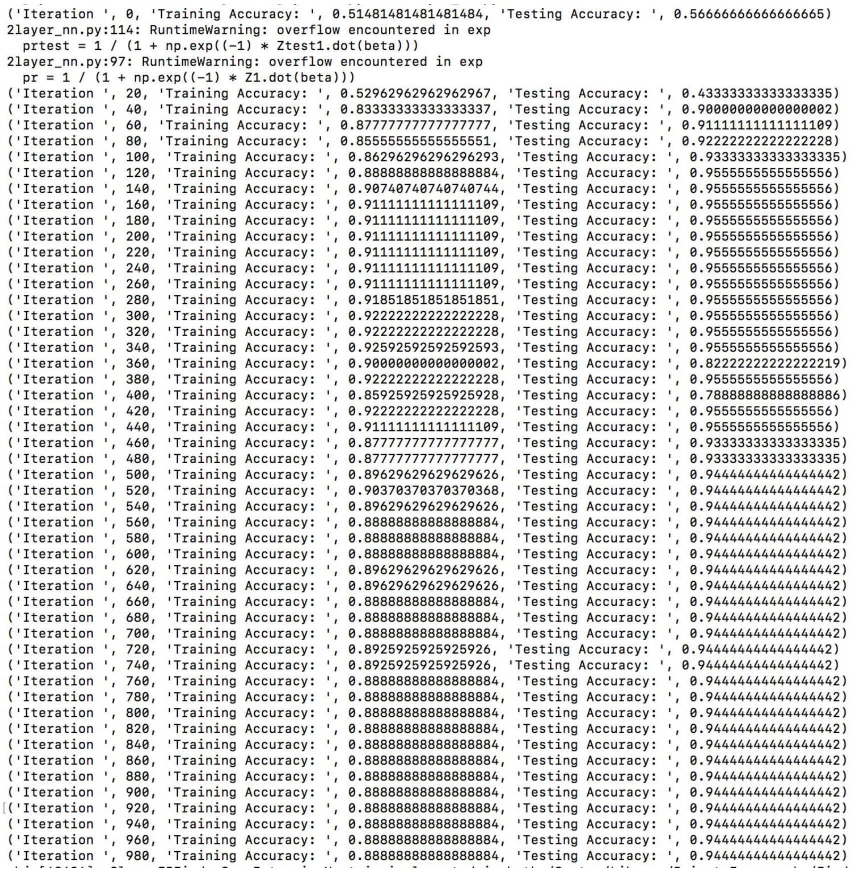
```
# training 255
# testing 110
After iteration 100, Train Accuracy = 0.976471,Test Accuracy = 0.981818
After iteration 200, Train Accuracy = 0.984314,Test Accuracy = 0.981818
After iteration 300, Train Accuracy = 0.984314,Test Accuracy = 0.981818
After iteration 400, Train Accuracy = 0.984314,Test Accuracy = 0.981818
After iteration 500, Train Accuracy = 0.984314,Test Accuracy = 0.981818
After iteration 600, Train Accuracy = 0.984314,Test Accuracy = 0.981818
After iteration 700, Train Accuracy = 0.992157,Test Accuracy = 0.981818
After iteration 800, Train Accuracy = 0.992157,Test Accuracy = 0.981818
After iteration 900, Train Accuracy = 0.992157,Test Accuracy = 0.981818
After iteration 1000, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1100, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1200, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1300, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1400, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1500, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1600, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1700, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1800, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 1900, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2000, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2100, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2200, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2300, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2400, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2500, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2600, Train Accuracy = 1.000000,Test Accuracy = 0.981818
After iteration 2700, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 2800, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 2900, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3000, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3100, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3200, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3300, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3400, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3500, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3600, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3700, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3800, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 3900, Train Accuracy = 1.000000,Test Accuracy = 1.000000
After iteration 4000, Train Accuracy = 1.000000,Test Accuracy = 1.000000
```

4. **SVM Plots** (Training and Testing Accuracy V/s. Iteration):



5. Neural Networks:





```
('Iteration ', 0, 'Training Accuracy: ', 0.51481481481481484, 'Testing Accuracy: ', 0.56666666666666665)
2layer_nn.py:114: RuntimeWarning: overflow encountered in exp
  prtest = 1 / (1 + np.exp((-1) * Ztest1.dot(beta)))
2layer_nn.py:97: RuntimeWarning: overflow encountered in exp
  pr = 1 / (1 + np.exp((-1) * Z1.dot(beta)))
('Iteration ', 20, 'Training Accuracy: ', 0.52962962962962967, 'Testing Accuracy: ', 0.43333333333333335)
('Iteration ', 40, 'Training Accuracy: ', 0.83333333333333337, 'Testing Accuracy: ', 0.90000000000000002)
('Iteration ', 60, 'Training Accuracy: ', 0.87777777777777777, 'Testing Accuracy: ', 0.91111111111111109)
('Iteration ', 80, 'Training Accuracy: ', 0.85555555555555551, 'Testing Accuracy: ', 0.92222222222222228)
('Iteration ', 100, 'Training Accuracy: ', 0.86296296296296293, 'Testing Accuracy: ', 0.93333333333333335)
('Iteration ', 120, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 140, 'Training Accuracy: ', 0.90740740740740744, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 160, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 180, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 200, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 220, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 240, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 260, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 280, 'Training Accuracy: ', 0.91851851851851851, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 300, 'Training Accuracy: ', 0.92222222222222228, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 320, 'Training Accuracy: ', 0.92222222222222228, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 340, 'Training Accuracy: ', 0.92592592592592593, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 360, 'Training Accuracy: ', 0.90000000000000002, 'Testing Accuracy: ', 0.82222222222222219)
('Iteration ', 380, 'Training Accuracy: ', 0.92222222222222228, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 400, 'Training Accuracy: ', 0.85925925925925928, 'Testing Accuracy: ', 0.78888888888888886)
('Iteration ', 420, 'Training Accuracy: ', 0.92222222222222228, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 440, 'Training Accuracy: ', 0.91111111111111109, 'Testing Accuracy: ', 0.9555555555555556)
('Iteration ', 460, 'Training Accuracy: ', 0.87777777777777777, 'Testing Accuracy: ', 0.93333333333333335)
('Iteration ', 480, 'Training Accuracy: ', 0.87777777777777777, 'Testing Accuracy: ', 0.93333333333333335)
('Iteration ', 500, 'Training Accuracy: ', 0.89629629629629626, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 520, 'Training Accuracy: ', 0.90370370370370368, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 540, 'Training Accuracy: ', 0.89629629629629626, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 560, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 580, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 600, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 620, 'Training Accuracy: ', 0.89629629629629626, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 640, 'Training Accuracy: ', 0.89629629629629626, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 660, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 680, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 700, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 720, 'Training Accuracy: ', 0.8925925925925926, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 740, 'Training Accuracy: ', 0.8925925925925926, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 760, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 780, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 800, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 820, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 840, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 860, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 880, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 900, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 920, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 940, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 960, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
('Iteration ', 980, 'Training Accuracy: ', 0.88888888888888884, 'Testing Accuracy: ', 0.94444444444444442)
```

SVM has the best accuracy while neural networks is the slowest.