Python Machine Learning: Solutions

Machine Learning Reading Club

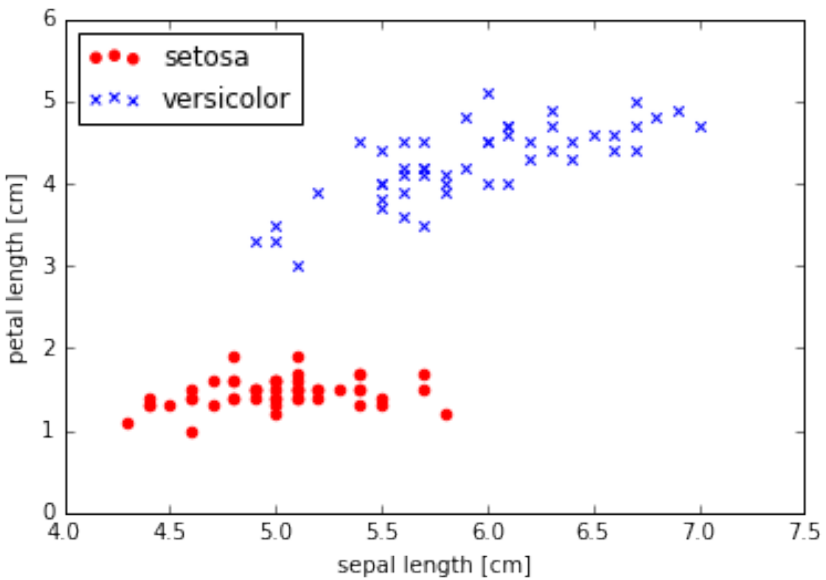
Mathematical Informatics Laboratory

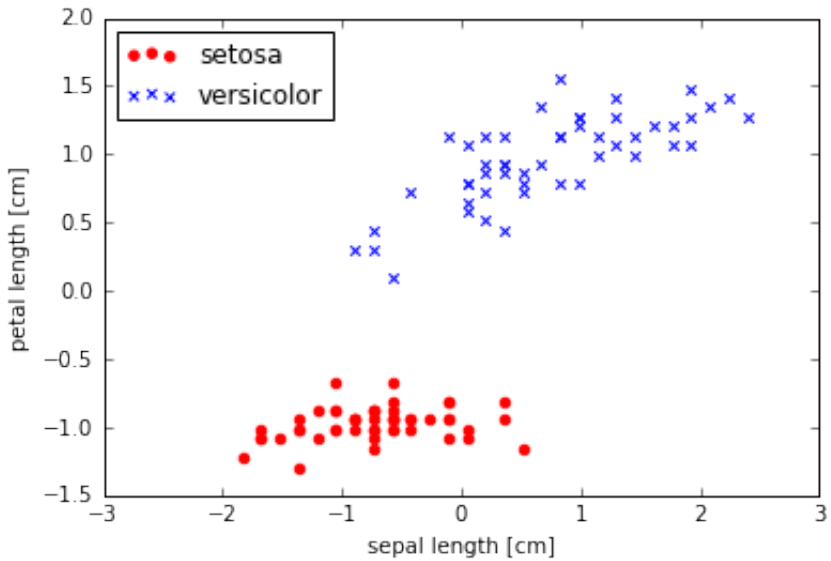
2018/07/09

Exercise 1: Standard Normalization

Equation for standardization: $X_{std} = \frac{(X - \mu_X)}{\sigma_X}$

```
X_std = (X - X.mean(axis=0))/X.std(axis=0)
```





Exercise 2: Perceptron Algorithm

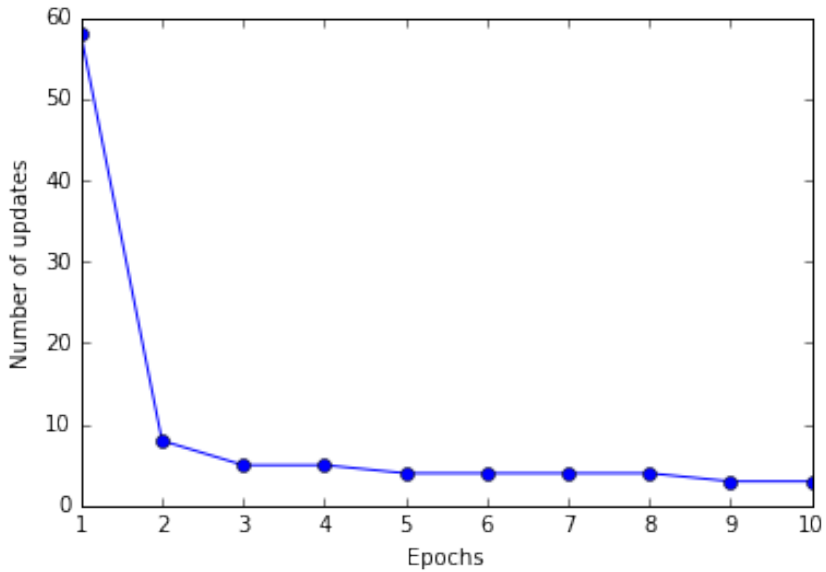
$$w_j := w_j + \Delta w_j$$

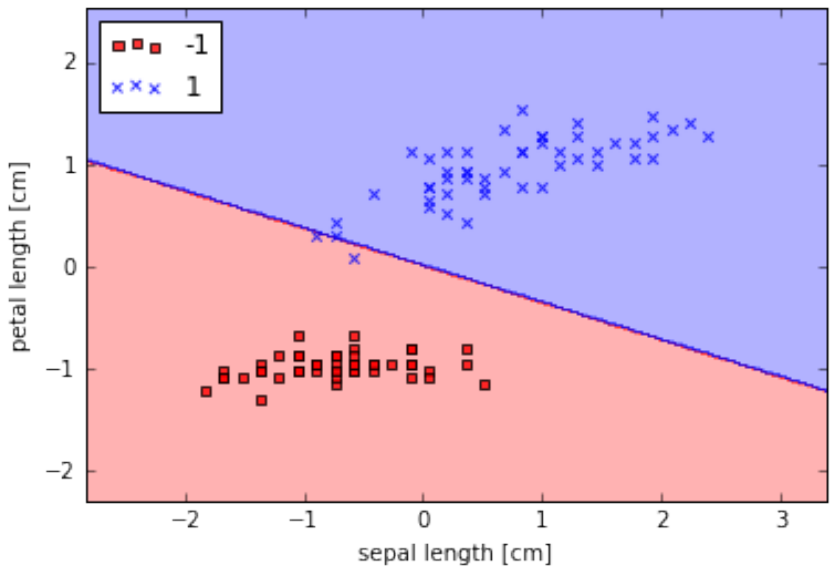
$$\Delta w_j = \eta \sum_i (y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$$

$$\Delta \mathbf{w} = \eta \mathbf{X}^T (\mathbf{y} - \hat{\mathbf{y}})$$

```
y_hat = self.predict(X)
X_ext = np.append([[1]] * X.shape[0], X, 1)
w_delta = np.dot((y - y_hat), X_ext)
w_delta *= self.eta
self.w_ += w_delta
errors = sum(y != y_hat)
```

Exercise 2: Perceptron Algorithm





Exercise 3: Adaline Algorithm

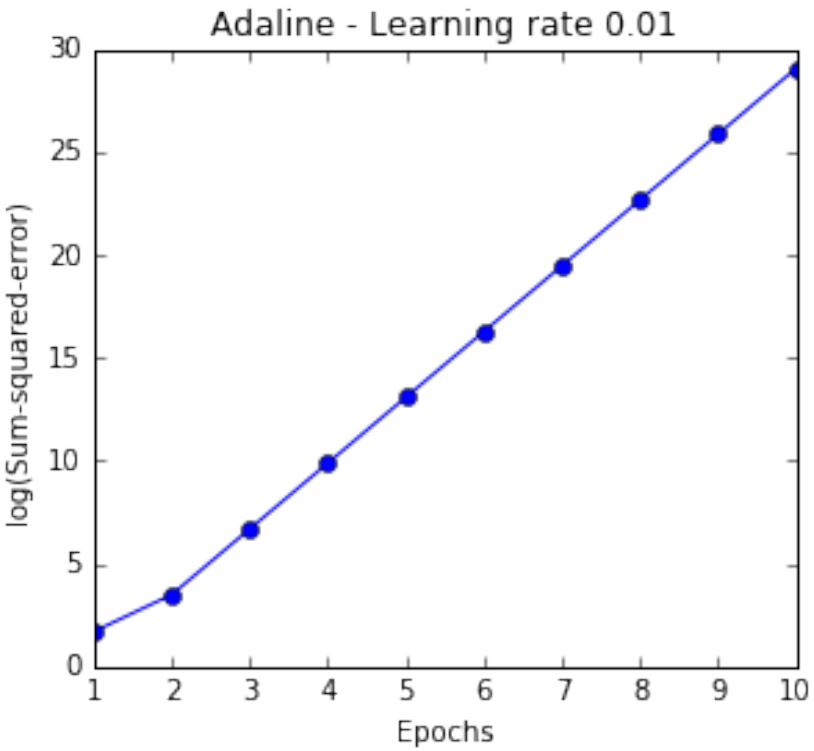
$$w_j := w_j + \Delta w_j$$

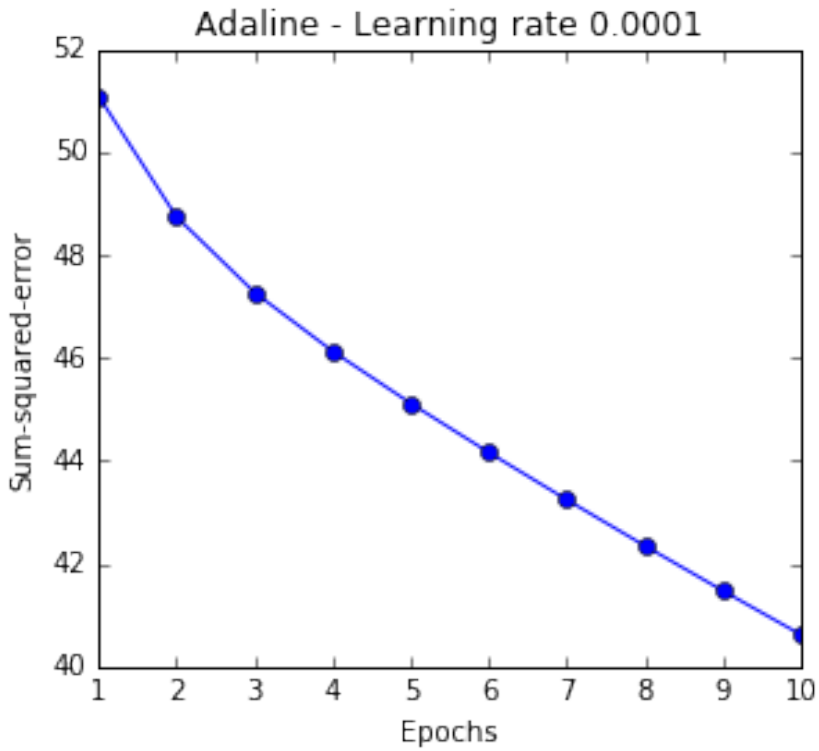
$$\Delta w_j = \eta \sum_i (y^{(i)} - \phi(\mathbf{w}^T \mathbf{x}^{(i)})) x_j^{(i)}$$

$$\Delta \mathbf{w} = \eta \mathbf{X}^T (\mathbf{y} - \phi(\mathbf{w}^T \mathbf{X}))$$

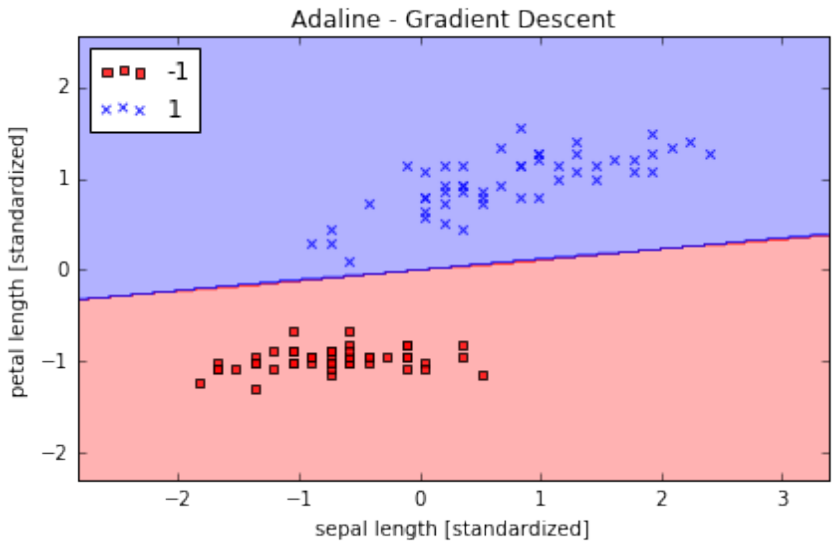
```
p_in = self.net_input(X)
p_out = self.activation(p_in)
X_ext = np.append([[1]] * X.shape[0], X, 1)
errors = y - p_out
w_delta = self.eta*np.dot(errors, X_ext)
self.w_ += w_delta
cost = np.sum(errors**2) / 2
```

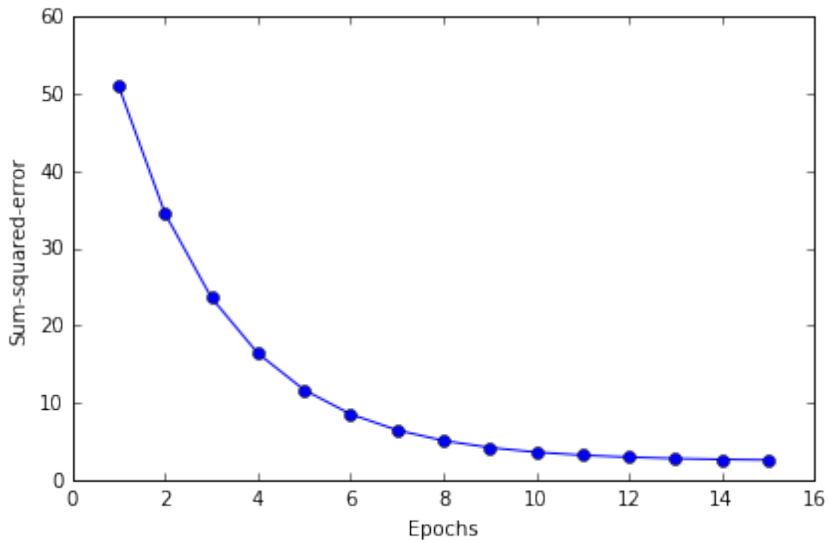
Exercise 3: Adaline Algorithm





Exercise 3: Adaline Algorithm





Exercise 4: Stochastic Gradient Descent

$$w_j := w_j + \Delta w_j$$

$$\Delta w_j = \eta \sum_i (y^{(i)} - \phi(\mathbf{w}^T \mathbf{x}^{(i)})) x_j^{(i)}$$

```
p_in = self.net_input(xi)
p_out = self.activation(p_in)
error = target - p_in
X_ext = np.append([1], xi)
w_delta = self.eta*np.dot(error, X_ext)
self.w_ += w_delta
cost = error**2 / 2
```

Exercise 4: Stochastic Gradient Descent