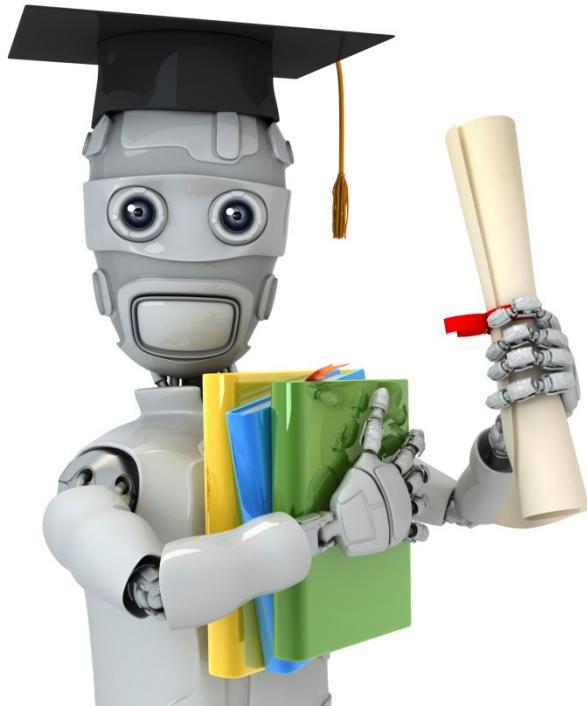


Machine Learning

Octave Tutorial

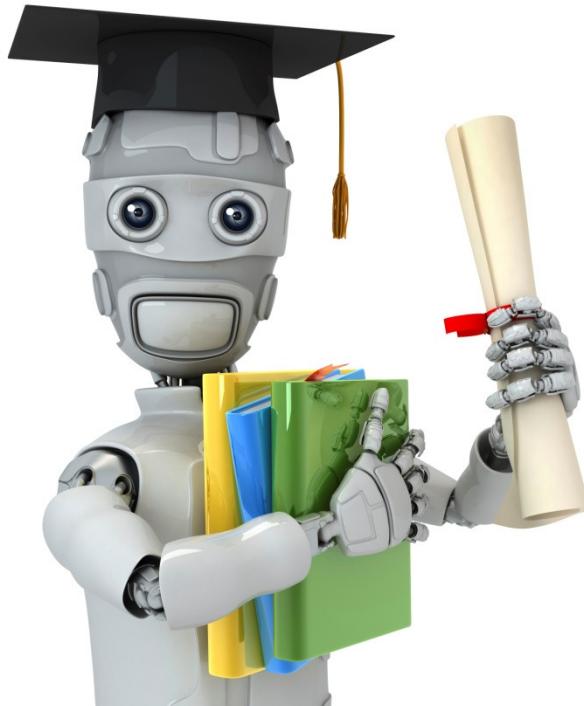
Basic operations



Machine Learning

Octave Tutorial

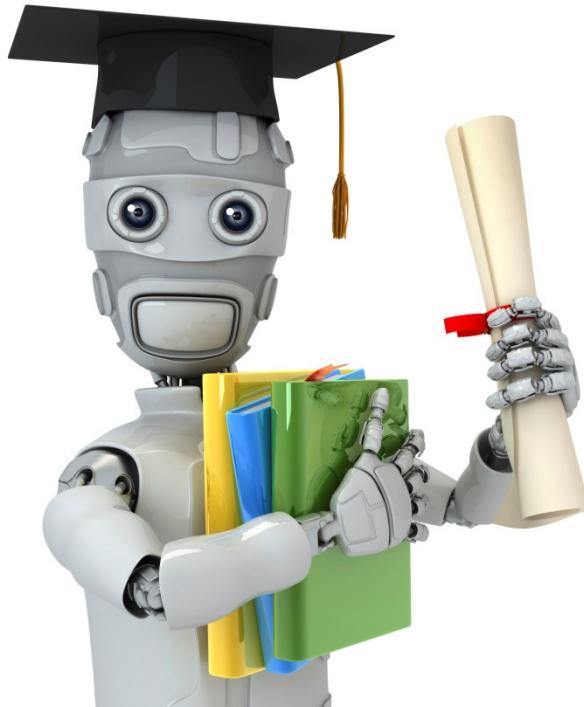
Moving data around



Machine Learning

Octave Tutorial

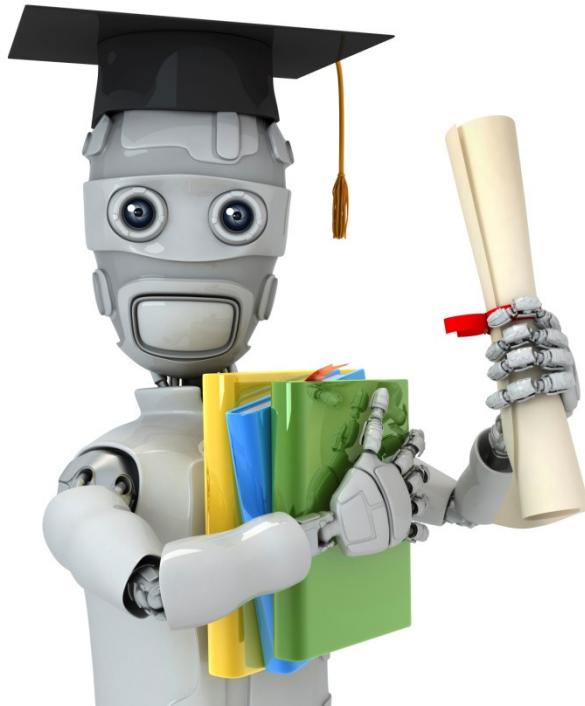
Computing on data



Machine Learning

Octave Tutorial

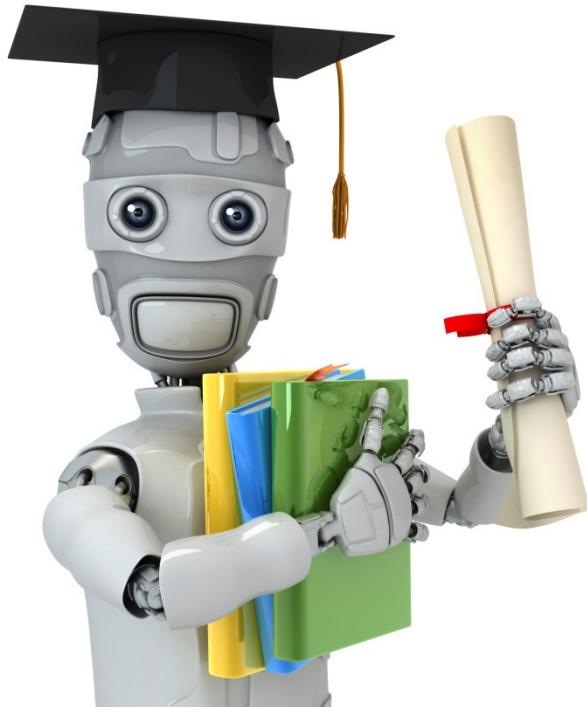
Plotting data



Machine Learning

Octave Tutorial

Control statements: for,
while, if statements



Machine Learning

Octave Tutorial

Vectorial implementation

Vectorization example.

$$h_{\theta}(x) = \sum_{j=0}^n \theta_j x_j \\ = \theta^T x$$

Unvectorized implementation

```
prediction = 0.0;  
for j = 1:n+1,  
    prediction = prediction +  
        theta(j) * x(y)  
end;
```

Vectorized implementation

```
prediction = theta' * x;
```

Vectorization example.

$$h_{\theta}(x) = \sum_{j=0}^n \theta_j x_j \\ = \theta^T x$$

Unvectorized implementation

```
double prediction = 0.0;  
for (int j = 0; j < n; j++)  
    prediction += theta[j] * x[y];
```

Vectorized implementation

```
double prediction  
= theta.transpose() * x;
```

Gradient descent

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (\text{for all } j)$$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

$$\begin{aligned}\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} \\ \theta_2 &:= \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)} \\ (n = 2) &\end{aligned}$$

$$\left| \begin{array}{l} u(j) = 2v(j) + 5w(j) \quad (\text{for all } j) \\ u = 2v + 5w \end{array} \right.$$