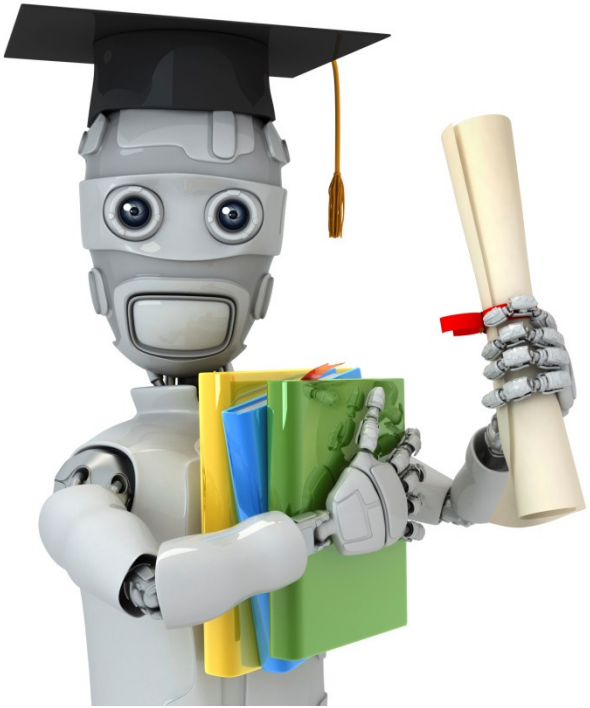
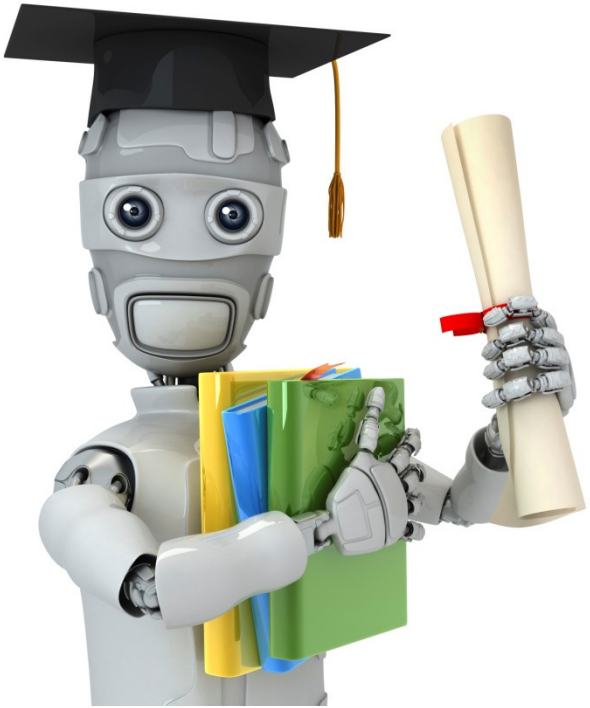


Machine Learning

# Octave Tutorial

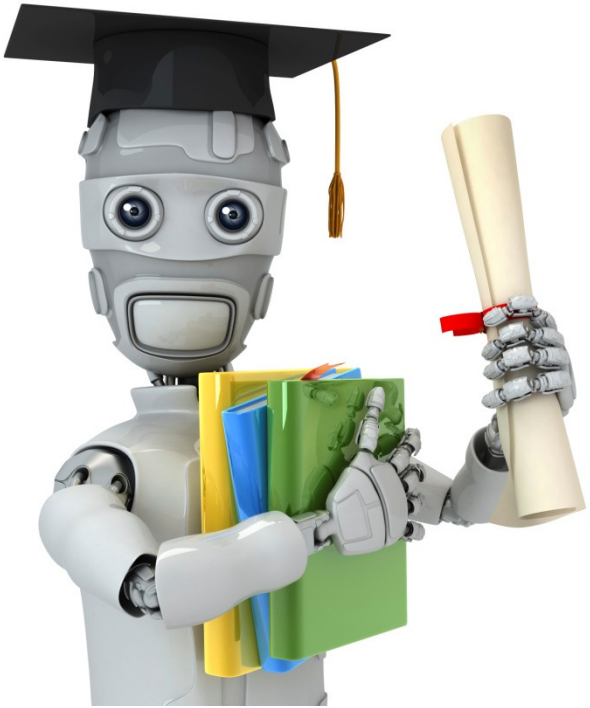
## Basic operations



Machine Learning

# Octave Tutorial

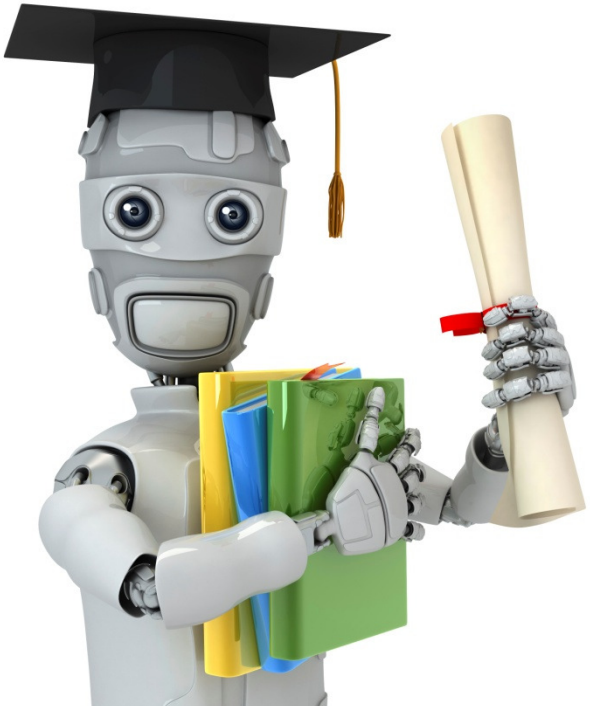
## Moving data around



Machine Learning

# Octave Tutorial

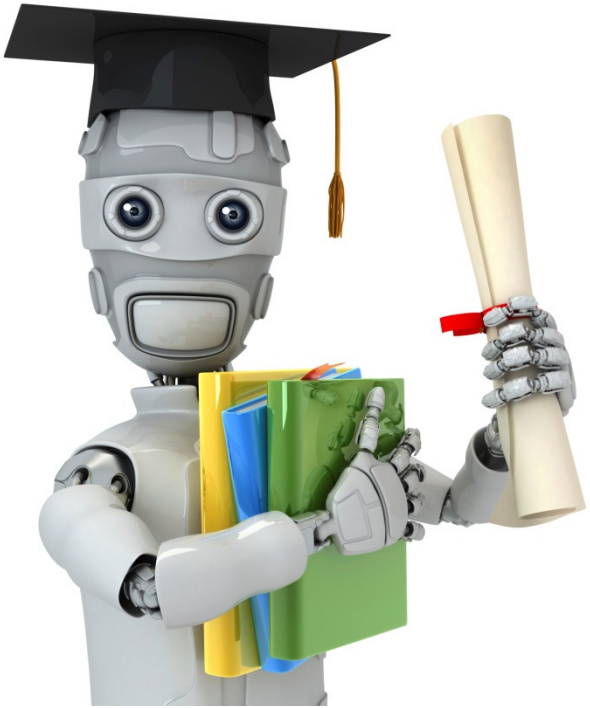
## Computing on data



Machine Learning

# Octave Tutorial

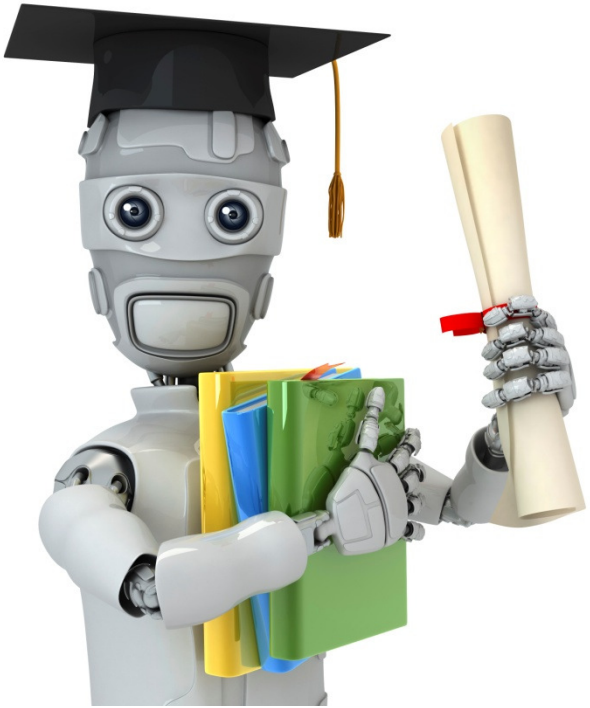
# Plotting data



Machine Learning

# Octave Tutorial

Control statements: for, while, if statements



Machine Learning

# Octave Tutorial

## Vectorial implementation

# Vectorization example.

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

$$= \theta^T x$$

## Unvectorized implementation

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

## Vectorized implementation

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

## Vectorization example.

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

$$= \theta^T x$$

### 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

*what is h*

*xij should be a part of SUM operation?*

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

Error

(for all $j$)

---

*the different of xj*

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

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

*total n-loop*

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

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

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

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

$(n = 2)$

$$u(j) = 2v(j) + 5w(j) \quad \text{(for all } j\text{)}$$

$$u = 2v + 5w$$