Ata Berk Çakır
21703127
Tutorial 3

```
J = theta * x

dtheta = x

theta_plus = theta + epsilon
theta_minus = theta - epsilon
cost_plus = forward_propagation(x, theta_plus)
cost_mines = forward_propagation(x, theta_minus)
gradapprox = (cost_plus - cost_mines)/(2*epsilon)

grad = backward_propagation(x, theta)

numerator = np.linalg.norm(grad - gradapprox)
denominator = np.linalg.norm(grad) + np.linalg.norm(gradapprox)
difference = numerator/denominator

   theta_plus = np.copy(parameters_values)
   theta_plus[i] = theta_plus[i] + epsilon
   J_plus[i],_ =  forward_propagation_n(X, Y, vector_to_dictionary(theta_plus))

   theta_minus = np.copy(parameters_values)
   theta_minus[i] = theta_minus[i] - epsilon
   J_minus[i],_ =  forward_propagation_n(X, Y, vector_to_dictionary(theta_minus))

   gradapprox[i] = (J_plus[i] - J_minus[i])/(2*epsilon)

difference = np.linalg.norm(grad - gradapprox)/(np.linalg.norm(grad) +
np.linalg.norm(gradapprox))
```