```python
import numpy as np

X = np.array(([2, 9], [1, 5], [3, 6]), dtype=float)
y = np.array(([92], [86], [89]), dtype=float)

X = X / np.amax(X, axis=0)
y = y / 100

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_grad(x):
    return x * (1 - x)

epoch = 1000
eta = 0.2
input_neurons = 2
hidden_neurons = 3
output_neurons = 1

wh = np.random.uniform(size=(input_neurons, hidden_neurons))
bh = np.random.uniform(size=(1, hidden_neurons))
wout = np.random.uniform(size=(hidden_neurons, output_neurons))
bout = np.random.uniform(size=(1, output_neurons))

for i in range(epoch):
    # Forward Propagation
    h_ip = np.dot(X, wh) + bh
    h_act = sigmoid(h_ip)
    o_ip = np.dot(h_act, wout) + bout
    output = sigmoid(o_ip)

    # Backpropagation
    Eo = y - output
    outgrad = sigmoid_grad(output)
    d_output = Eo * outgrad

    Eh = d_output.dot(wout.T)
    hiddengrad = sigmoid_grad(h_act)
    d_hidden = Eh * hiddengrad

    wout += h_act.T.dot(d_output) * eta
    wh += X.T.dot(d_hidden) * eta

print("Normalized Input: \n" + str(X))
print("\nActual Output: \n" + str(y))
print("\nPredicted Output: \n", output)
```