1. Consider the neural network with 3 layers shown in Figure 1, which is used for regression.

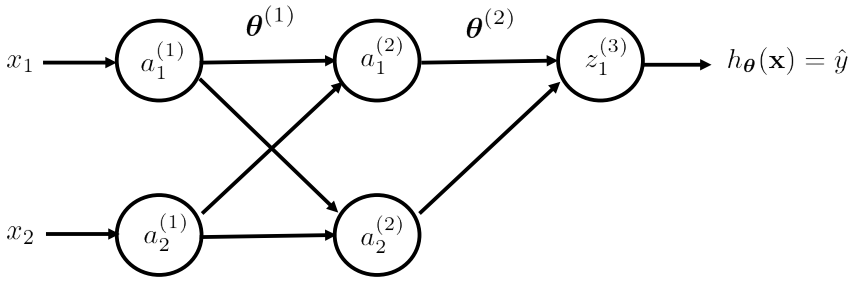

Figure 1: Neural network with 3 layers.

Define

$$\mathbf{a}^{(1)} = \mathbf{x} = (x_1, x_2), \mathbf{a}^{(2)} = g(\mathbf{z}^{(1)}), \text{ and } \mathbf{z}^{(l+1)} = \boldsymbol{\theta}^{(l)}\mathbf{a}^{(l)}, \ l = 1, 2,$$

and the predicted value $\hat{y}$ is given by $h_{\boldsymbol{\theta}}(\mathbf{x}) = z_1^{(3)}$. Suppose that the squared error loss is used.

(a) Write a code that uses backpropagation for training the given neural network.
Ans: The Matlab cost is provided on the next page.

(b) Consider a single sample with $\mathbf{x} = [2\ 1]^T$ and $y = 3$. Initialize all weights of the edges using independent and identically distributed (i.i.d.) uniform(0,1) random variables (i.e., the probability that it belongs to the interval (0, p) is equal to p, $p \in (0, 1)$). Train the neural network using the sample for 50 iterations with a learning rate $\gamma = 0.05$. Plot the squared error loss as a function of iteration.
Ans: The plot of the cost is shown in Figure 2.

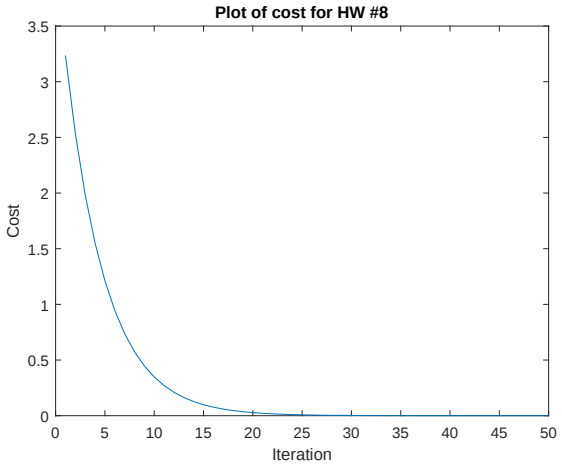

Figure 2: Plot of the cost.

```
% Generate random parameters
Theta = rand(2, 3, 2) ;
dTheta = zeros(size(Theta)) ; % derivative of cost functioin w.r.t. Theta

input = [2 1]' ;
y = 3 ;
gamma = 0.05 ; % learning rate

a = zeros(2, 2) ; % output of activation function in layers 1 and 2
a(:,1) = input ; % a^(1) = input
da = zeros(2, 2) ; % derivative of cost function w.r.t. a
z = zeros(2, 3) ; % output of input functions at each layer
dz = zeros(2, 3) ; % derivative of cost function w.r.t. z

Tmax = 50 ; % max # of iterations
cost = zeros(1, Tmax) ; % contains the cost for each iteration

for t = 1:Tmax,  % # of iterations
        % Forward propagation
        z(:,2) = Theta(:,:,1) * [1 ; a(:,1)] ;
        a(:,2) = exp(z(:,2)) ./ sum(exp(z(:,2))) ;
        z(1,3) = Theta(1,:,2) * [1 ; a(:,2)] ;
        cost(t) = (y - z(1,3))^2 ;

        % Backward propagation
        dz(1, 3) = - 2 * (y - z(1,3)) ; % derivative for regression problem
        dTheta(1, 2:3, 2) = dz(1, 3) * a(:,2)' ;
        dTheta(1, 1, 2) = dz(1, 3) ;
        da(:, 2) = Theta(1,2:3,2)' * dz(1, 3) ;
        dz(:, 2) = da(:, 2) .* exp(-z(:, 2)) ./ (1 + exp(-z(:, 2))).^2 ;
        dTheta(:, 2:3, 2) = dz(:, 2) * a(:,1)' ;
        dTheta(:, 1, 1) = dz(:, 2) ;

        % update parameters using gradient-based algorithm
        Theta = Theta - gamma * dTheta ;

end

% Plot the cost as a function of iteration
plot([1:Tmax], cost) ;
```