Backpropagation algorithm
Studying the problem of curve fitting

Backpropagation:

Given cost function: $E = \frac{1}{n}\sum_{i=1}^{n}(d_i - f(x_i, w))^2$

Applying backpropagation to calculate the derivatives across 1 x N x 1 network:

Consider a_1 and a_2 as the local fields of the hidden the two-layer network and, φ_1 and φ_2 as the activation functions. W1, b1 are the weights and biases for the hidden layer and W2 and b2 are the weights and bias for the output layer. Thus, the derivates are calculated as follows.

$$y' = d_i - f(x_i,\ w)$$
$$\frac{\partial E}{\partial W2} = -\ \varphi_1(a_1)\ * y' *\ \varphi_2'(a_2)$$
$$\frac{\partial E}{\partial b2} = -\ y' *\ \varphi_2'(a_2)$$
$$\frac{\partial E}{\partial W1} = -X_i\ * y' * \varphi_2'(a_2)\ *\ W2\ * \varphi_1'(a_1)$$
$$\frac{\partial E}{\partial b2} = -y' * \varphi_2'(a_2)\ *\ W2\ * \varphi_1'(a_1)$$

Pseudocode:

1. Given N = 300
2. Initialize X $\in$ $R^{300 \times 1}$ at random on [0, 1], v $\in$ $R^{300 \times 1}$ at random on [-1/10, 1/10].
3. Initialize d $\in$ $R^{300 \times 1}$ as $d_i = \sin(20x_i) + 3x_i + v_i$.
4. Initialize W1 $\in$ R^{24x2} by drawing from a normal distribution with mean = 0 and variance = 1. Similarly, initialize W2 $\in$ R^{25x1} from a normal distribution with mean = 0 and variance = 1 / N.
5. Set eta1 = 2 and eta2 = 1/√N * 0.01
6. Initialize epoch = 0
7. Initialize mse_values(epoch) = 0 for epoch = 0,1,2,...
8. Do:
    a. For i = 1 to N:
        i. Perform the forward pass as follows:
            1. Calculate a1_induced = W1 * X_i where X_i is the current training example.
            2. Calculate a1_activated = tanh(a1_induced)
            3. Initialize finalA = [1 a1_activated].T where the first row is all 1's and the second row is the elements of a1_activated.
            4. Calculate a2_induced = W2 * finalA
            5. Calculate a2_activated = a2_induced * 1
            6. Set y = a2_activated
        ii. Perform the backward pass as follows:
            1. Calculate a2_backward = a2_induced * 1
            2. Calculate a1_backward = 1 - $\tanh(\text{a1\_induced})^2$
        iii. Perform the weights updated as follows:

1. Calculate y_prime = d_i - y where d_i is the desired output and y is the calculated output for training data X_i.
2. Calculate
dW2 = - a1_activated * y_prime * a2_backward
3. Calculate db2 = - y_prime * a2_backward
4. Calculate
dW1 = - X_i * y_prime * a2_backward * W2' * a1_backward
where W2' is given by removing the first element of W2.
5. Calculate
db1 = - y_prime * a2_backward * W2' * a1_backward
6. Calculate newW1 = Second row of W1 - (eta1 * dW1)
Calculate newb1 = First row of W1 - (eta1 * db1)
Update W1 ← [newb1, newW1] concatenated row wise.
7. Calculate newW2 = W2' - eta2 * dW2
Calculate newb2 = First element of W2 - eta2 * db2
Updated W2 ← [newb2, newW2] appended element wise.

b. Epoch ← epoch + 1
c. Generate y_i for each X_i by performing the forward pass as described in 8.a.i.
d. Calculate MSE = $\frac{1}{n}\sum_{i=1}^{n}(d_i - f(x_i, w))^2$
e. If the absolute difference subsequent MSE values < threshold value (~ $1e^{-7}$), break out of the loop.
f. If the current MSE value > previous MSE value, update eta1 = eta1 - 0.1 * eta1

9. Calculate y_i for each X_i by performing the forward pass as described in 8.a.i.
10. Plot the generated y_i and desired outputs d_i against X_i for i = 1 to N on the same plot. The two curves should overlap each other for a best fit.
11. If the curves do not overlap as required, tweak the learning rates (eta1 and eta2) until a good fit is achieved.