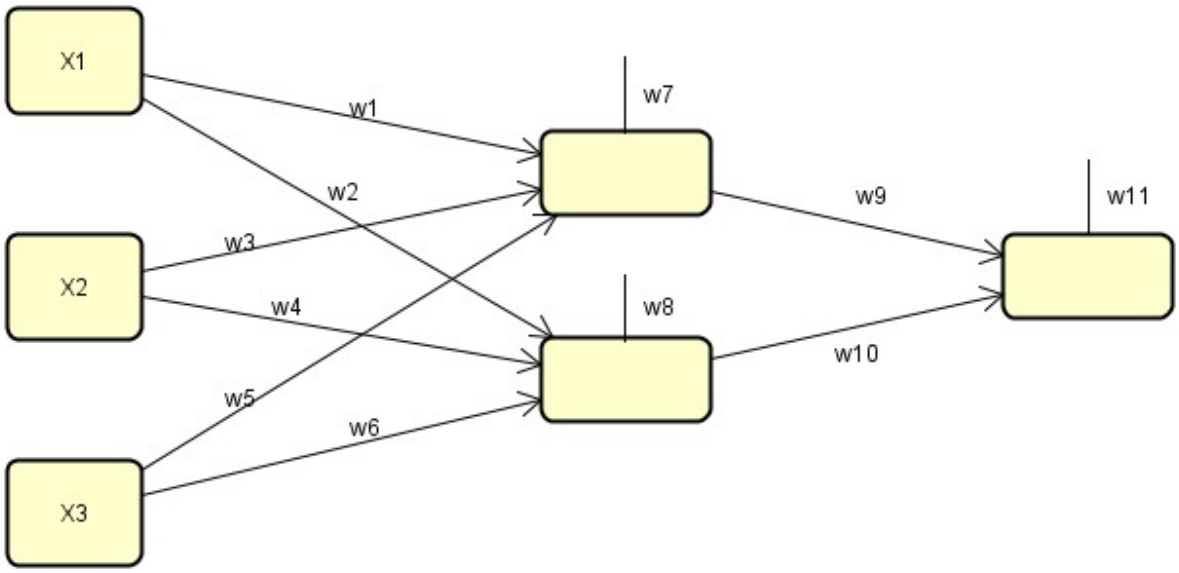
1. Give the structure of your ANN.



2. Give all the weights of the trained ANN.

Layer #0 (hidden layer):

Neuron1: [0.42281477800527256, 0.045289831110553715, 1.107607415874252, 0.8414475358461728]
Neuron2: [0.8032734979222897, 0.9822581306521022, -0.16364366871214714, 0.8525145890092779]

w1=0.42281477800527256
w2=0.8032734979222897
w3=0.045289831110553715
w4=0.9822581306521022
w5=1.107607415874252
w6=-0.16364366871214714
w7=0.8414475358461728
w8=0.8525145890092779

Layer #1 (output layer):

Neuron1: [-0.9802053318792855, 0.09262331874006624, 0.20078070465457504]

w9=-0.9802053318792855
w10=0.09262331874006624
w11=0.20078070465457504

3. Give the equations that you used to update the weights explaining all the parameters on them.

Artificial neural networks (ANNs) are built out of interconnected artificial units, each of which taking a number of real-valued inputs and producing a real valued output. Those units are connected through weights between layers (input, hidden, output). The main purpose of backpropagation algorithm is finding a weight vector to best approximate the training examples, i.e. have network outputs as close to target values as possible. The idea is to modify the weights according to the negative of gradient of the error function to get a fast reduction of error on this example.

Formula that is used for the output layer units:

$$e_j = (t_j - o_j)o_j(1 - o_j)$$

The error of each unit in the output layer is represented as the difference between expected output and the actual output multiplied by activation function derivative for that unit. Activation function derivative is calculated as the product of output and difference between 1 and output.

Formula that is used for the hidden layer units:

$$e_j = -(1 - o_j)o_j * \sum_{k \in Downstream(j)} e_k * w_{kj}$$

The error of each unit in the hidden layer is represented as the product of negative activation function derivative and downstream. Downstream is calculated as the sum of the product of the error of the next layer unit and the weight that is connecting unit from the current layer and the unit from the next layer.

Formula for the weight update:

$$\Delta w_{kj} = \eta * e_k * x_{kj}$$

Every weight in a layer is calculated as the product of the error, given input and learning rate (η). Learning rate specifies the step size in the gradient search.

4. Give the percentage of correctness of the total validation data set (the last 700 cases) and give the percentage of correctness of the cases in the validation set that survived according to the expected value.

Percentage of total correct: 76.42857142857142
Percentage of positive correct: 38.57142857142858

5. Give the percentage of survive that every individual had according to the ANN output and compare it with the expected value.

Survival rate according to ANN: 20.308950477055884
Survival rate according to expected: 32.30349840981372