The task: In this project, you will implement a multi-layer perceptron and a variation of the backpropagation algorithm. Below a roadmap is given for the analysis of the algorithm on a two simple tasks. Provide a detailed description of the network model and the learning algorithm, with the equations. Make sure that your discussion is clear and complete such that your work can be understood and replicated by someone who knows little about neural nets.

Implement the standard backpropagation algorithm, and test it on the XOR problem.
Two hidden neurons (that is the network is 2input X 2hidden X 1output).
tanh() is the activation function of both the hidden and the output neurons.
The patterns are bipolar, that is -1 or +1.
Use incremental training scheme with randomized presentation of the patterns.

Plot the error (E) with respect to number of epochs.
Draw the decision surfaces learned by the two hidden neurons.
Now redefine the error term as \( E = E_{\text{standard}} + \alpha \sum_{i,j} \text{square}(w_{ij}) \). That is add the squared sum of all the weights (including the bias) as multiplied by a parameter \( \alpha \) to the standard error function of backpropagation. Choose \( \alpha \) such that the extra term, it does not prevent the learning of the patterns. If you wish you can make it adaptive.

Derive the new weight update equations for the new learning algorithm.
Your task is to analyze and demonstrate the effect of this new term is on the learning and generalization.

Comment on the effect of this new term.
Analyze the effect by running the standard and the variation learning on the following problem:
Use a multi-layer perceptron of size 5-5-1, with the inputs are named as A,B,C,D, and E.

The task is to learn to compute the Boolean function \((A \text{ and } B) \text{ or } (C \text{ and } E)\). Write down the truth table which consist of 32 possible combinations. Create a training sets by choosing 70%, 90% and 100% of the combinations. Create a training sets by choosing 70%, 90% and 100% of the combinations. The remaining percentage (only for the first two cases) are used as the test set. Create at least 5 different training and test sets for the first two percentages.

Run both types of algorithms on these different training sets
Use enough number of epochs for training, and plot the error (both total and the different components).
Analyze the test set performances for the 70% and 90% cases.
Analyze the weights for the 100% case.. Cross-check with your initial comments.
Feel free to add extra analysis.[Also, if the number of hidden units turn out to be not sufficient and you can have more.]