Assignment 1) NN matlab simulation for ‘Understanding how to solve pattern recognition problems using backpropagation’ You may find the following two exercises useful as an introduction to the neural network toolbox in matlab. Initial Exercises E1 E2 use matlab help documentation. E1) Read ‘Multilayer Shallow Neural Networks and Backpropagation Training’ for general background. E2) Perform exercise in 'Classify Patterns with a shallow Neural Network' - cancer\_dataset with trainscg and default parameters. Save the ‘advanced script’ at the end of the exercise and use this code as a basis for carrying out the assignment. In particular, you need classification error rather than cross-entropy for Exp 1). (Note that there is a panopto help lecture if needed, see ‘matlab help for using graphical interface to generate code for assignment’) If you are having difficulty with the initial exercises E1,E2 please contact me. For Exp 1) 2) 3) 4) I am happy to clarify what is required but cannot help you to carry out the implementation of assignment. ASSIGNMENT - EXPERIMENTS For help see eeem005\_NNassignment info.pptx Exp 1) Using cancer dataset (as in E2) and ‘trainscg’ or an optimiser of your choice, vary nodes and epochs (that is using early stopping for epochs) over suitable range, to find optimal choice in terms of classification test error rate of node/epochs for 50/50% random train/test split (no validation set). It is suggested that you initially try epochs = [ 1 2 4 8 16 32 64], nodes = [2 8 32], so there would be 21 node/epoch combinations. (Hint 1: from the ‘advanced script’ in E2, nodes and epochs can be changed), Repeat each of the 21 node/epoch combinations at least thirty times, with different 50/50 split and take average and report classification error rate and standard deviation(std). Graph classification train and test error rate and std as node-epoch changes, that is plot error rate vs epochs for different number of nodes. Report the optimal value for test error rate and associated node/epoch values. (Hint 2: as epochs increases you can expect the test error rate to reach a minimum and then start increasing, you may need to set the stopping performance criteria to zero to achieve the desired number of epochs – Hint 3: to find classification error rates for train and test set, you need to check the code from E2. Note: classification error rate = number of mis-classifications divided by total number of patterns, fraction or percentage e.g. predict the class labels from trained network and compute number of differences with target labels. Note that train and test patterns/targets may be obtained using tr.trainInd and tr.testInd (see documentation for train.m) and simulate with y = net(x) and convert to labels as shown in the advanced script from E2. As an alternative, it is also possible to obtain classification error rate from the confusion matrix). Exp 2) For cancer dataset, choose an appropriate value of node and epochs, based on Exp 1) and use ensemble of individual (base) classifiers with random starting weights and Majority Vote to see if performance improves - repeat the majority vote ensemble at least thirty times with different 50/50 split and average and graph (Each classifier in the ensemble sees the same training patterns). Repeat for a different odd number (prevents tied vote) of individual classifiers between 3 and 25, and comment on the result of individual classifier accuracy vs ensemble accuracy as number of base classifiers varies. Consider changing the number of nodes/epochs (both less complex and more complex) to see if you obtain better performance, and comment on the result with respect to why the optimal node/epoch combination may be different for an ensemble compared with the base classifier, as in Exp 1). ( Hint 4: to implement majority vote you need to determine the predicted class labels – probably easier to implement yourself rather than use the ensemble matlab functions. For the graph, use average value over thirty runs for y-axis. The x-axis should be Number of classifiers. Also for complexity, another graph with x-axis as Number of epochs) Exp 3) Repeat Exp 2) for cancer dataset with two different optimisers of your choice e.g. ‘trainlm’ and ‘trainrp’. Comment and discuss the result and decide which is more appropriate training algorithm for the problem. In your discussion, include in your description a detailed account of how the training algorithms (optimisations) work. Exp 4) Please note that you should only attempt this final section if you have time and after you have completed all other sections – only counts ten percent. The requirement is to distinguish between two equi-probable classes of ‘overlapping’ two-dimensional Gaussians’, where class 1 has mean 𝜇1=[0,0] variance 𝜎12=1 and class 2 has 𝜇2=[2,0] variance 𝜎22=4. It can be proved that the Bayes (optimum) boundary for this problem is a circle whose centre is located at [−23⁄,0] and radius 2.34. Find the optimal choice of node/epochs in terms of test error rate, using 300/3000 pattern train/test split, possibly using an ensemble, and plot the decision boundary on the same plot as the Bayes boundary, and discuss your result. (Hint 5:You may find the matlab function mvnrnd.m useful for generating random numbers) It - Explain in detail how to do this assignment

Sure! Here is a detailed step-by-step guide on how to complete the assignment on understanding how to solve pattern recognition problems using backpropagation in MATLAB:

Step 1: Initial Exercises (E1, E2) E1) Read the documentation titled 'Multilayer Shallow Neural Networks and Backpropagation Training' to gain a general background understanding of the topic. E2) Perform the exercise in 'Classify Patterns with a Shallow Neural Network' using the cancer\_dataset and trainscg optimizer with default parameters. Save the 'advanced script' provided at the end of the exercise, as it will serve as a basis for the assignment. Note that in this assignment, we need to use classification error rate instead of cross-entropy for Experiment 1.

Step 2: Understanding the Assignment Carefully read and understand the requirements of the assignment as outlined in the instructions provided. The assignment consists of four experiments (Exp 1, Exp 2, Exp 3, Exp 4), each with specific tasks and instructions.

Step 3: Experiment 1 For this experiment, we will use the cancer\_dataset and 'trainscg' optimizer (or any other optimizer of your choice). We will vary the number of nodes and epochs over a suitable range to find the optimal combination in terms of classification test error rate for a 50/50 random train/test split (without using a validation set).

1. Set the range of values for nodes and epochs. For example, nodes = [2 8 32] and epochs = [1 2 4 8 16 32 64].
2. Modify the 'advanced script' obtained from E2 to change the number of nodes and epochs according to the range you set.
3. Repeat each of the 21 node/epoch combinations at least thirty times with different 50/50 train/test splits.
4. For each run, record the classification error rate for both train and test sets.
5. Calculate the average and standard deviation of the classification error rates for both train and test sets for each node/epoch combination.
6. Plot the classification error rate vs. epochs for different numbers of nodes.
7. Report the optimal value of the test error rate and its associated node/epoch values. You can also comment on any observations or trends observed from the plots.

Step 4: Experiment 2 For this experiment, we will again use the cancer\_dataset, but this time we will implement an ensemble of individual classifiers using the majority vote technique to see if performance improves.

1. Choose an appropriate value of nodes and epochs based on the optimal values obtained from Experiment 1.
2. Choose an odd number of individual classifiers between 3 and 25.
3. Modify the 'advanced script' obtained from E2 to train multiple individual classifiers with random starting weights.
4. Implement the majority vote technique to determine the predicted class labels for the test set.
5. Repeat the ensemble of individual classifiers at least thirty times with different 50/50 train/test splits.
6. Calculate the average classification accuracy for the ensemble and plot it against the number of base classifiers.
7. Repeat the experiment with different values of nodes and epochs (both less complex and more complex) to observe any changes in performance.
8. Comment on the results, particularly on the comparison between individual classifier accuracy and ensemble accuracy as the number of base classifiers varies.

Step 5: Experiment 3 For this experiment, we will repeat Experiment 2, but this time we will use two different optimizers of your choice, such as 'trainlm' and 'trainrp', to train the individual classifiers in the ensemble.

1. Choose two different optimizers of your choice.
2. Modify the 'advanced script' obtained from E2 to train the individual classifiers

Now give an Introduction for this assignment related to its title and the experiments of this assignment

Introduction:

The assignment titled "Understanding how to solve pattern recognition problems using backpropagation" focuses on using neural networks for pattern recognition tasks, specifically using the backpropagation algorithm for training. The assignment involves four experiments (Exp 1, Exp 2, Exp 3, and Exp 4) that explore different aspects of neural network training and ensemble methods using MATLAB's neural network toolbox.

Exp 1: In this experiment, the cancer dataset (used in Exercise 2) is used to train a neural network using the "trainscg" optimization algorithm, or an optimizer of choice. The number of nodes and epochs are varied over a suitable range to find the optimal choice in terms of classification test error rate. Multiple runs are performed with different random train/test splits to obtain average classification error rates and standard deviations. The results are graphed to analyze the relationship between the number of nodes, epochs, and the classification error rate. The goal is to determine the optimal node/epoch combination that minimizes the classification error rate.

Exp 2: This experiment builds on Exp 1 by using an ensemble of individual (base) classifiers with random starting weights and applying the Majority Vote method for classification. The goal is to investigate if performance improves by combining multiple classifiers in an ensemble. The number of base classifiers is varied between 3 and 25, and the classification accuracy of the ensemble is compared with that of individual classifiers. Additionally, the effect of changing the number of nodes/epochs on the ensemble's performance is analyzed to understand how the optimal node/epoch combination may differ for an ensemble compared to the base classifier.

Exp 3: Similar to Exp 2, this experiment involves building an ensemble of classifiers with different optimizers (e.g., 'trainlm' and 'trainrp') to train the neural network. The performance of the ensemble is compared to that of individual classifiers to determine which optimizer is more appropriate for the given problem. A detailed account of how the training algorithms (optimizations) work is provided in the discussion of the results.

Exp 4: This experiment is an optional section that involves a more complex pattern recognition problem where two classes of overlapping 2-dimensional Gaussians need to be distinguished. The optimal choice of node/epochs in terms of test error rate is determined using a 300/3000 pattern train/test split, possibly using an ensemble. The decision boundary obtained from the neural network is compared to the Bayes boundary, which is the optimum boundary for the given problem. The results are discussed in detail.

Overall, these experiments provide a comprehensive understanding of how to effectively use neural networks for pattern recognition problems, including the impact of various parameters such as the number of nodes, epochs, optimizers, and ensemble methods. The results obtained from these experiments will help in making informed decisions on the optimal choices for solving pattern recognition problems using backpropagation.

Modify the above passage to include brief introduction about what is pattren recognition, MLP backpropogation

Introduction:

Pattern recognition is a fundamental task in machine learning and artificial intelligence that involves identifying patterns or regularities in data. One popular approach for solving pattern recognition problems is through artificial neural networks, which are biologically-inspired computational models that can learn from data and make predictions. Multi-layer Perceptron (MLP) is a type of neural network architecture that consists of multiple layers of interconnected nodes (neurons) and is trained using the backpropagation algorithm.

The assignment titled "Understanding how to solve pattern recognition problems using backpropagation" focuses on using MLP neural networks for pattern recognition tasks, specifically using the backpropagation algorithm for training. Backpropagation is a supervised learning algorithm that adjusts the weights of the network in a way that minimizes the error between the predicted outputs and the actual outputs. This assignment involves four experiments (Exp 1, Exp 2, Exp 3, and Exp 4) that explore different aspects of neural network training and ensemble methods using MATLAB's neural network toolbox.

Exp 1: In this experiment, the cancer dataset (used in Exercise 2) is used to train an MLP neural network using the "trainscg" optimization algorithm, or an optimizer of choice. The number of nodes and epochs are varied over a suitable range to find the optimal choice in terms of classification test error rate. Multiple runs are performed with different random train/test splits to obtain average classification error rates and standard deviations. The results are graphed to analyze the relationship between the number of nodes, epochs, and the classification error rate. The goal is to determine the optimal node/epoch combination that minimizes the classification error rate.

Exp 2: This experiment builds on Exp 1 by using an ensemble of individual (base) classifiers with random starting weights and applying the Majority Vote method for classification. The goal is to investigate if performance improves by combining multiple classifiers in an ensemble. The number of base classifiers is varied between 3 and 25, and the classification accuracy of the ensemble is compared with that of individual classifiers. Additionally, the effect of changing the number of nodes/epochs on the ensemble's performance is analyzed to understand how the optimal node/epoch combination may differ for an ensemble compared to the base classifier.

Exp 3: Similar to Exp 2, this experiment involves building an ensemble of classifiers with different optimizers (e.g., 'trainlm' and 'trainrp') to train the MLP neural network. The performance of the ensemble is compared to that of individual classifiers to determine which optimizer is more appropriate for the given problem. A detailed account of how the training algorithms (optimizations) work is provided in the discussion of the results.

Exp 4: This experiment is an optional section that involves a more complex pattern recognition problem where two classes of overlapping 2-dimensional Gaussians need to be distinguished. The optimal choice of node/epochs in terms of test error rate is determined using a 300/3000 pattern train/test split, possibly using an ensemble. The decision boundary obtained from the neural network is compared to the Bayes boundary, which is the optimum boundary for the given problem. The results are discussed in detail.

Overall, these experiments provide a comprehensive understanding of how to effectively use MLP neural networks with backpropagation for solving pattern recognition problems, including the impact of various parameters such as the number of nodes, epochs, optimizers, and ensemble methods. The results obtained from these experiments will help in making informed decisions on the optimal choices for solving pattern recognition problems using backpropagation.

Sure! Here's a step-by-step guide on how to conduct Experiment 2, which involves implementing an ensemble of individual classifiers using the majority vote technique on the cancer\_dataset:

Step 1: Choose an appropriate value of nodes and epochs based on the optimal values obtained from Experiment 1.

* Refer to the results obtained from Experiment 1, where you trained a single classifier using a neural network on the cancer\_dataset.
* Identify the optimal values of nodes (number of neurons in the hidden layers) and epochs (number of times the entire dataset is passed through the network during training) that resulted in the highest classification accuracy.
* Use these optimal values as a starting point for Experiment 2, where you will train multiple individual classifiers.

Step 2: Choose an odd number of individual classifiers between 3 and 25.

* Decide on the number of individual classifiers (base classifiers) you want to include in your ensemble.
* Choose an odd number of classifiers between 3 and 25, as this will ensure that there is always a majority class in case of ties during the majority vote technique.

Step 3: Modify the 'advanced script' obtained from Experiment 1 to train multiple individual classifiers with random starting weights.

* Start with the 'advanced script' that you used in Experiment 1 to train a single neural network classifier on the cancer\_dataset.
* Modify the script to create multiple copies of the neural network, each with random initial weights.
* Train each individual classifier using the same training dataset but with a different set of random initial weights.
* Store the trained models for each individual classifier for later use during the majority vote.

Step 4: Implement the majority vote technique to determine the predicted class labels for the test set.

* After training the individual classifiers, use them to predict the class labels for the test dataset.
* For each instance in the test dataset, collect the predicted class labels from each individual classifier.
* Implement the majority vote technique, where the predicted class label that occurs most frequently among the individual classifiers is chosen as the final predicted class label for that instance.

Step 5: Repeat the ensemble of individual classifiers at least thirty times with different 50/50 train/test splits.

* To ensure the robustness of the ensemble, repeat the entire process of training and predicting with individual classifiers multiple times.
* Randomly split the cancer\_dataset into training and test sets using a 50/50 split multiple times (at least thirty times).
* Train the ensemble of individual classifiers on the training set and evaluate their performance on the test set using the majority vote technique.
* Keep track of the classification accuracy of the ensemble for each run.

Step 6: Calculate the average classification accuracy for the ensemble and plot it against the number of base classifiers.

* After repeating the ensemble process multiple times, calculate the average classification accuracy of the ensemble across all runs.
* Plot the average classification accuracy of the ensemble against the number of base classifiers used in the ensemble.
* Observe any trends or patterns in the plot to understand how the performance of the ensemble changes with the number of base classifiers.

Step 7: Repeat the experiment with different values of nodes and epochs (both less complex and more complex) to observe any changes in performance.

* To understand how the performance of the ensemble is affected by the complexity of the individual classifiers, repeat the experiment with different values of nodes and epochs.
* Choose less complex (lower number of nodes, fewer epochs) and more complex (higher number of nodes, more epochs) configurations for training the individual classifiers.
* Follow the same process of training, predicting, and evaluating the ensemble with these different configurations.
* Compare the performance of the ensemble with different configurations to identify any changes in performance.

Step 8: Comment on the results, particularly on the comparison between individual classifier accuracy and ensemble