ARTIFICIAL NEURAL NETWORKS (ANNs)

INTRODUCTION:

An artificial neural network (ANN) is an information-processing model with a structure that is inspired by the dense and parallel connections of neurons in the mammalian brain. Learning in the brain can occur following an adjustment in synaptic connections between neurons. Similarly, learning in ANN's occurs via modification of synaptic weights between connecting neural units following an exposure to a set of training data.

Epilepsy consists of a group of neurological disorders distinguished by unprovoked seizures caused by neuronal hyper-excitability in the brain. This disorder affects close to 1% of the population worldwide and about 300,000 Canadians. Around one third of patients continue to experience seizures despite multiple attempts to control them with medication. For these patients responsive neurostimulation presents a possible therapy capable of aborting seizures before they affect a patient's normal activities.

In order for a responsive neurostimulation device to successfully stop seizures, a seizure must be detected and electrical stimulation applied as early as possible. A seizure that builds and generalizes beyond its area of origin will be very difficult to abort via neurostimulation. Current seizure detection algorithms in commercial responsive neurostimulation devices are tuned to be hypersensitive, and their high false positive rate results in unnecessary stimulation.

In addition, physicians and researchers working in epilepsy must often review large quantities of continuous EEG data to identify seizures, which in some patients may be quite subtle. Automated algorithms to detect seizures in large EEG datasets with low false positive and false negative rates would greatly assist clinical care and basic research.

In this assignment, you will implement a feedforward neural network using MATLAB Neural Network Toolbox to classify seizure and non-seizure brain states using features computed from human patient intracranial EEG data. These data were provided the UPenn - Mayo Clinic Seizure Detection Challenge sponsored by the American Epilepsy Society.

GENERAL OVERVIEW:

The program will be designed to detect brain states (non-seizure vs seizure) using features from EEG recordings.

TRAINING SET: Each input training case consists of 90 values each corresponding to a features extracted from EEG data using a window of 1 second. Features correspond to the frequency power computed from 1-90Hz.

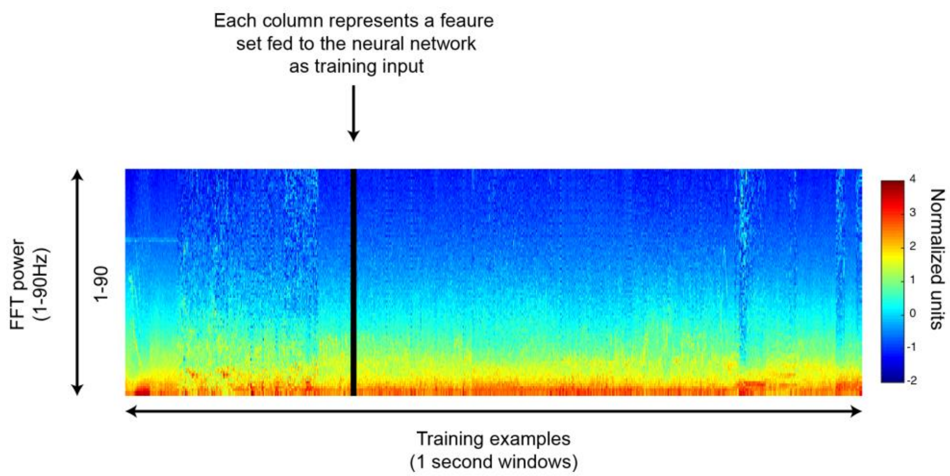
The target of each corresponding training case consists of 2 values, with the 1's in each section corresponding to the correct classification of that state. The training input file contains features computed from over 1000 distinct 1 second windows of EEG traces from seven patients. In other words, your network will be trained from these training cases.

TEST SET: The test set has the same format of features and targets as the training set, however it is computed from EEG data not used to train the network. Once your network is trained by achieving a desired mean-square error (MSE) on the training set, the performance will be determined from the accuracy at which the network can correctly identify these test cases that are not part of the training set.

The data file "training.mat" is available for download on the course website under: laboratory notes/lab #5. It consists of four variables, they are:

TrainingFeatures	(Input ACSII file for training, 90 x 1058 double)
TrainingTargets	(Target ACSII file for training, 2 x 1058 double)
TestingFeatures	(Input ACSII file for testing, 90 x 348 double)
TestingTargets	(Target ACSII file for testing, 2 x 348 double)

Input



Target

Target vector:	[0 1]	[1 0]
State:	seizure	non-seizure

Figure 1: Computation of features for all training examples their target vectors

According to the aforementioned data format, your network should have 90 input units and 2 output units.

The goal of this experiment is to design different feedforward neural networks by changing the following parameters and evaluating their performance. Performance evaluation should include

1) *training rate*: The number of iterations required to achieve the design objective of MSE,

2) *accuracy* on test cases that the network did not see during the learning phase.

EXPERIMENTS:

Today you will run four experiments **IA**, **IB**, **IC** experiment, and experiment **II**.

IA) EFFECT OF HIDDEN UNITS:

Using a neural network with one hidden layer, and gradient descent method with step size of 0.5, alter the number of units in the hidden layer (5, 10, 40) and explore whether increased number of hidden units will have a positive effect on the network performance.

IB) EFFECT OF HIDDEN LAYER NUMBERS:

Alter the number of hidden layer (no hidden layers, 1 hidden layer or 2 hidden layers) in the feedforward neural network, using 20 units per hidden layer and gradient descent method with step size of 0.5. Determine whether increased number of hidden layers will have a positive effect on the network performance.

IC) STEP SIZE/LEARNING RATE CONSIDERATIONS

Try different step sizes (lr = 0.1, 0.5, 1) using gradient descent training function on a neural network with one hidden layer network with 40 hidden units. Determine whether large step size will always expedite learning.

II) EFFECT OF MOMENTUM

Investigate the effect of momentum using network with 1 hidden layer (20 units) and gradient descent with momentum (mc = 0.1, 0.5, 0.9). Determine whether a strong momentum term will always expedite learning.

In order to start,

Linux: launch MATLAB. Download *training.mat* from the course website and double click on it to load it into MATLAB's workspace.

Windows: launch MATLAB. Open '*S:/BME445/Source/Lab5'* by selecting "File" icon and double click on "training.mat" to load it into MATLAB's workspace.

Instruction:

1. Type command **nnstart** and then select **pattern recognition** to load the Neural Network Toolbox.

2. Click "**Import...**" button to load the *TrainingFeatures* into "**Input Data**" and to load the *TrainingTargets* variables into **"Target Data**".

3. Click "**Train"** to train your new neural network. Figure 1 shows the architecture.

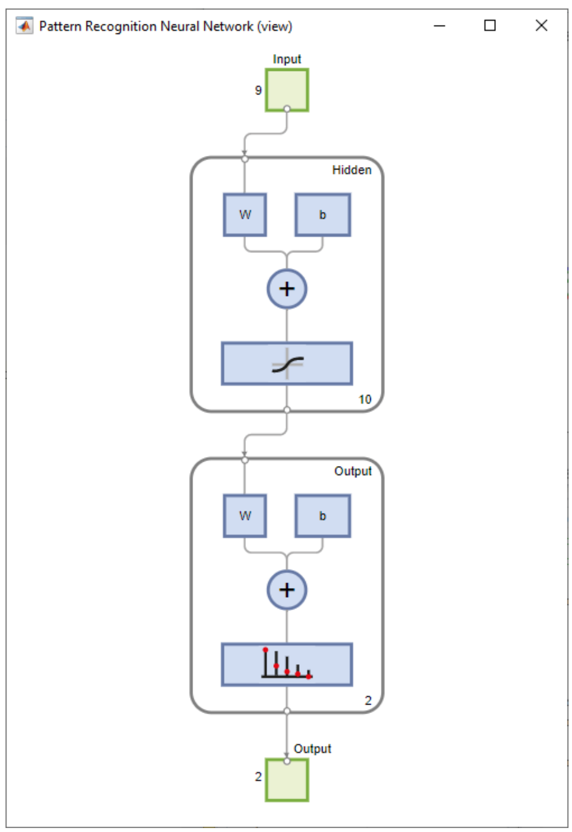


Figure 1: Pattern Recognition Neural Network (view)

4. Select **Generate Code** > **Generate Simple Training Script** to create MATLAB code to reproduce the previous steps from the command line.

5. Depending on which experiments you are running, change the Training function name, Adaption learning function, and the Number of layers via script. NOTE: The number of hidden layers in Neural

Network script corresponds to the number of all layers (hidden layers) except the last layer (output layer). Output layer has a fixed number of neurons. Think about why that is. 2) The Transfer Function for the output layer must be TANSIG, all other layers – LOGSIG.

Tables below outline the appropriate network parameters for different experiments:

EXPERIMENT IA			
	Network 1	Network 2	Network 3
Training function	traingd	traingd	traingd
Performance function	MSE	MSE	MSE
Number of layers	2	2	2
Number of neurons for layer 1	5	10	40
Number of neurons for layer 2	2*	2*	2*

EXPERIMENT IB			
	Network 1	Network 2	Network 3
Training function	traingd	traingd	traingd
Performance function	MSE	MSE	MSE
Number of layers	1	2	3
Number of neurons for layer 1	2*	20	20
Number of neurons for layer 2	---	2*	20
Number of neurons for layer 3	---	---	2*

* Output layer

EXPERIMENT IC			
	Network 1	Network 2	Network 3
Training function	traingd	traingd	traingd
Performance function	MSE	MSE	MSE
Number of layers	2	2	2
Number of neurons for layer 1	20	20	20
Number of neurons for layer 2	2*	2*	2*

* Output layer

EXPERIMENT II			
	Network 1	Network 2	Network 3
Training function	traingdm	traingdm	traingdm
Performance function	MSE	MSE	MSE
Number of layers	2	2	2
Number of neurons for layer 1	40	40	40
Number of neurons for layer 2	2*	2*	2*

* Output layer

Hyperparameter Adjustment:

- *net = patternnet(hiddenSizes,trainFcn,performFcn)*

returns a pattern recognition neural network with a hidden layer size of hiddenSizes, a training function, specified by trainFcn, and a performance function, specified by performFcn.

Make sure to put ‘mse’ as performFcn

Change the *epochs*, *goal*, and *lr* parameters in the script using following commands:

- *net.trainParam.epochs — Maximum number of epochs to train. The default value is 1000.*
- *net.trainParam.goal — Performance goal. The default value is 0.*
- *net.trainParam.lr — Learning rate. The default value is 0.01.*
- *net.trainParam.mc — Momentum constant. The default value is 0.9.*
- *net.trainParam.max_fail — Maximum validation failures. The default value is 6.*

- *Epochs* is the maximum number of iterations. You can set the epochs to 30000 or more if you want.
- *Goal* is the stopping criteria. A recommended goal for this project should be set to *0.01*, that is, training will terminate when MSE reaches below 0.01. However, you may find that this goal will take too long to obtain. For this lab, it is acceptable if you train your network with MSE goal of 0.1. These parameters determine the stopping criteria for network training. The network will stop learning either when its performance has reached the desired MSE or when the number of training step has reached the maximum number of iterations even though the desired learning objective is not achieved. In the latter case, you might want to reinitialize the network weights to re-train the network from a different initial position.
- *lr* is the learning rate, or step size of the network; the increment by which the network weights will change. There is a trade-off to find here. An overly high learning rate can cause the training to skip over desirable error minima; an overly low learning rate can get cause the training to get stuck in local minima and consume valuable time.

6. Depending on your experiment, you will need to enter the appropriate values for the learning parameters:

Learning rate (lr)			
	Network 1	Network 2	Network 3
Experiment IA	0.1	0.1	0.1
Experiment IB	0.1	0.1	0.1
Experiment IC	0.1	0.5	1

Experiment II			
	Network 1	Network 2	Network 3
Learning rate (lr)	0.1	0.1	0.1
Momentum (me)	0.1	0.5	0.9

7. **Run the script** to train the network. Sometimes your training might stop due to reaching validation check limit. This represents that the network reached a threshold number of successive iterations that network validation performance failed to decrease, and the network has not

necessarily reached its full learning potential. While the default value (6) should be fine, if you notice that you reach validation check limits, you can increase it by changing the *max_fail* parameter. NOTE: During and after training you can check on the performance (among other things) of your neural network through *performance = perform(net ,t, y)* . However, if done during the ANN training, it will significantly slow down the training process.

8. After you have exported your trained network, you can simulate or test it using the test set by typing the following in the MATLAB workspace:

testoutput = sim(net, feat_test);

where *testoutput* will be the network output for the test data, *net* is the name of your newly trained network, *TestingFeatures* is the test set input that was given to you.

You now need to compare the *testoutput* with the test set targets *(TestingTargets)* provided, to determine the percentage of accuracy on the test set. Notice that both matrices are 2x348 in size, so you will have to compare the corresponding columns of each matrix to determine whether the trained network was able to correctly identify each test case. For example, the 1st column of the variable *testoutput* represents the likelihood that the nth test case is a non-seizure state, and the second column represents the likelihood that the nth test case is a seizure state. For example, if the nth column of *testoutput* is:

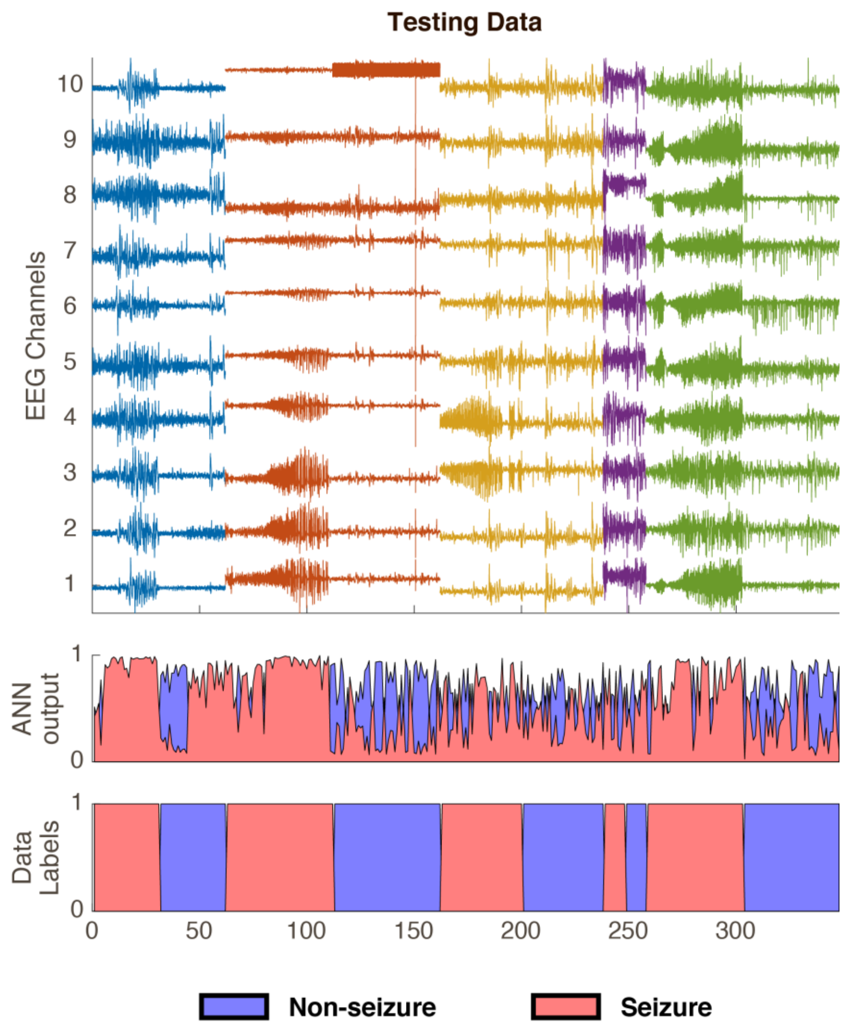
$$[\mathbf{0.7239} \quad 0.2726]^T$$

Then according to the trained network, it would classify the nth test case as non-seizure state. However, when compared with the nth column of *TestingTargets,*

$$[0 \quad \mathbf{1}]^T$$

The nth test case should be classified as seizure state**.** We see that this case has been identified incorrectly, hence constituting a failure. The percentage of accuracy can then be computed by counting the total number of success over the total number of test cases. HINT: One way to streamline this comparison is by finding the indices of maximum values within the 2x348 matrices – both *testoutput* and *TestingTargets*.

9. Run the function *displayTestData(testoutput, TestingTargets, time, testData)* to visualize the EEG data from 5 patients and the corresponding ANN output and data labels (*TestingTargets*). Note the test cases where the neural network fails to classify the right brain state.



Repeat the above procedures for ALL your network designs.

Laboratory 5 Investigation and Write-up

Learning in Artificial Neural Networks

In laboratory 5, you will explore feed forward neural networks using the MATLAB Neural Network Toolbox. During your investigation, please explore MATLAB and consider the following:

1) Attach the MSE plot for each network trained. There should be **<u>12 figures</u>** in total (3 figures per each case). **To obtain the MSE plot: Once a network has been trained, in the Neural Network Training window, choose Performance under the PLOTS section to plot the MSE plot.**

2) Summarize the accuracy of **<u>each</u>** network on the test set. After the training MSE has reached the desired level, simulate the trained network on the test set and evaluate the percentage of correct network response for each brain state, as well as the overall accuracy. **Also, indicate the MSE goal used to train the network (i.e. 0.01 or 0.05).**

	IA			IB		
	Network 1	Network 2	Network 3	Network 1	Network 2	Network 3
Non-seizure state	________%	________%	________%	________%	________%	________%
Seizure state	________%	________%	________%	________%	________%	________%
Overall Accuracy	________%	________%	________%	________%	________%	________%

	IC			II		
	Network 1	Network 2	Network 3	Network 1	Network 2	Network 3
Non-seizure state	________%	________%	________%	________%	________%	________%
Seizure state	________%	________%	________%	________%	________%	________%
Overall Accuracy	________%	________%	________%	________%	________%	________%

3) Answer **ALL** questions in the four sections below.

EFFECT OF HIDDEN UNITS:

I) Does increased number of hidden unit always imply better performance on testing? Why or why not?

II) If a classification problem can be solved using an ANN with 10 hidden units, would an ANN with 40 units also solve the same problem? How about vice versa?

III) What are the pros and cons of having more hidden units?

EFFECT OF NUMBER OF HIDDEN LAYERS:

I) Does increased number of hidden layers always imply better performance on testing? Why or why not?

II) If a classification problem can be solved using an ANN without any hidden layers, would an ANN having one hidden layer also solve the same problem? How about vice versa?

III) Can an artificial neural network without any hidden layers be trained to perform the "AND" logic? How about the "OR" logic?

STEP SIZE/ LEARNING RATE CONSIDERATIONS:

I) Is larger step size always better? If you are using only a simple gradient descent search algorithm, what potential problems would you run into if the step size is too large? Too small?

EFFECT OF MOMENTUM:

I) What is the motivation in using momentum with simple gradient descent algorithm?

II) What potential problems are there if you use a momentum term that is too large?

4) For you particular ANNs, are there any specific test cases that are more difficult to classify? Can you speculate why?

W

Lab Write-up Format

title/date	Laboratory title and date the experiment was performed
name	Names of **everyone** in the laboratory group [one **(1)** lab report is to be submitted by each lab group]. Include full names and all student numbers.
purpose	One or two sentences identifying the objective or purpose of the investigation
results/discussion	All questions from the lab investigation and write-up section should be answered here. Any additional observations, analysis performed during the laboratory may be included. In addition, a simple one paragraph explanation of what you did and what you found out should be included at the end of the observations/discussion section. Write this paragraph as if you were explaining your results to someone who is not familiar with the laboratory topic (i.e. someone who hasn't taken this course before). **(maximum of 200 words for this paragraph).**
figures	Print out and include all figures and/or screen shots (All figures/screen shots requested in the lab investigation and write-up section must be included, along with any additional figures that may complement your observations/results).