EMEC – 5173: Intelligence Tools for Engineering Application

Project Report

Name: Kuo Yang

Student ID: 0682523

Date: 2019/04/13

# **Introduction:**

5173 project is an application implemented by the fuzzy neuro network. The project is to config fuzzy neuro network and train all the params by the supervised learning algorithm. And the trained model can perform the prediction of the new data input. The whole training process applies hybrid training methods including steepest gradient decent (SGD) and recursive least square estimator(RLSE).

The project has 3 inputs as x, y, z and for each input which has 2 membership functions and this will generate totally 8 rules, mapping through all the possible conditions. The sigmoid function is selected as the membership functions, each of which has 2 parameters to be decided in the training process, so totally 12 premise parameters. The fuzzy inference is done by the TSK-1 model with 4 unknown parameters for each rule. Therefore, there are 32 consequent parameters that needs to be trained by the training data. During the training part, maximum epoch limit is 200 and the maximum accumulated error per epoch is 1e-5. In the testing part, 3000 pairs data will be used to test the trained model.

# **Theory Review:**

ANFIS – adaptive network-based fuzzy inference system is a class of adaptive networks that applies hybrid learning rule for the architectures representing both the Sugeno and Tsukamoto fuzzy model.[1]

Layer 1, every node in this layer is an adaptive node with a node function and it is the membership function.

Layer 2, every node in this layer is a fixed node and its output is to product of all the incoming signals.

Layer 3, every node in this layer is a fixed node labeled N. Every node calculates the ratio of the rule’s norm by the fire strength being divided by the sum of all fire strength.

Layer 4, every node in this layer is an adaptive node with a node function which is TSK-1 model in this project.

Layer 5, there is a single node in this layer which is to compute the overall output as the summation of all incoming signals. [1]

There are two kinds of parameters in the ANFIS model and they are premise parameters and consequent parameters. The front is the parameters in the membership function and the latter is the parameters of TSK-1 model in this project. The learning algorithm of premise parameters is the steepest gradient decent method and the one for consequent parameters is the recursive least square estimator method.

# **Model Construction:**

The training data is from the Mackey glass generator and data has 4 number in one pair and the first three are the inputs and the fourth number is the output.

The fuzzy neuro network, in this project, has 1 input layer, 1 output layer and 3 hidden layers. The input layers has 6 neurons and each two of them accepts one input, so all 3 inputs are assigned to 3 pairs of neurons (each pair contains 2 neurons). All 6 neurons are defined as 6 sigmoid membership functions () with 12 different initial parameters. The 1st layer takes the training data pairs into the membership functions and generates the outputs which will be pushed forward to the 2nd layer.

The 2nd layer is a hidden layer which composed all rules. This layer maps through all the possible combinations among 3 pairs of membership functions, so totally there are 8 rules created in this layer. The membership functions are µA1 & µA2, µB1 & µB2, µC1 & µC2 and each pair can accept one input in the 1st layer. After the mapping in this layer, all rules contain [µA1, µB1, µC1], [µA1, µB1 ,µC2], [µA1, µB2, µC1], [ µA1, µB2, µC2] and [µA2, µB1, µC1], [µA2, µB1 ,µC2], [µA2, µB2, µC1], [ µA2, µB2, µC2]. Each rule is composed of 3 different membership functions outputs and the result of each rule is done by the T norm product operation. Therefore, there are 8 outputs in the layer 2 and they will be pushed forward to the layer 3.

In the layer 3, the norm calculation is done based on the formula: . So there are totally 8 outputs going to the next layer.

The layer 4 defines all the action parts of the fuzzy system. For each rule, the action output is according to the formula: .

For each output of layer4, there are 4 params to perform linear combination with the inputs, so the whole number of these consequent params are 32 and each output has 4 params. All the outputs will be summed up in the layer 5 and the result is the inference action.

In terms of training part, SGD as well RLSE are applied in this application. For RLSE, it’s an iterative way to apply the least estimator square method and result is always based on the last step calculation, so the initial consequent params are acquired by performing LSE on first 10 data pairs and the remaining data pairs are used to train consequent params by RLSE in the training process.

For SGD, the chain rule is applied to obtain the derivative of each membership function params. The definition of error function and its chained function are defined as follows:

Note: this equation has 8 different combination for 8 different corresponding rules, so the derivative of params in membership function from this equation must take the combination into account.

The derivative of params of membership function from the error function is super complicated as the fire strength involves different combination of different membership functions. Here gives an example of derivative of of µ from the error function.

Note: this partial derivative is a vector involving f1 to f4 and this is because that the first 4 norms all related to µ as the each norm is calculated based on every rule. Each rule is a combination of 3 different membership function as [µA1, µB1, µC1], [µA1, µB1 ,µC2], [µA1, µB2, µC1], [ µA1, µB2, µC2] and [µA2, µB1, µC1], [µA2, µB1 ,µC2], [µA2, µB2, µC1], [ µA2, µB2, µC2]. Therefore, the first 4 combination must be taken in to account of this derivative calculation.

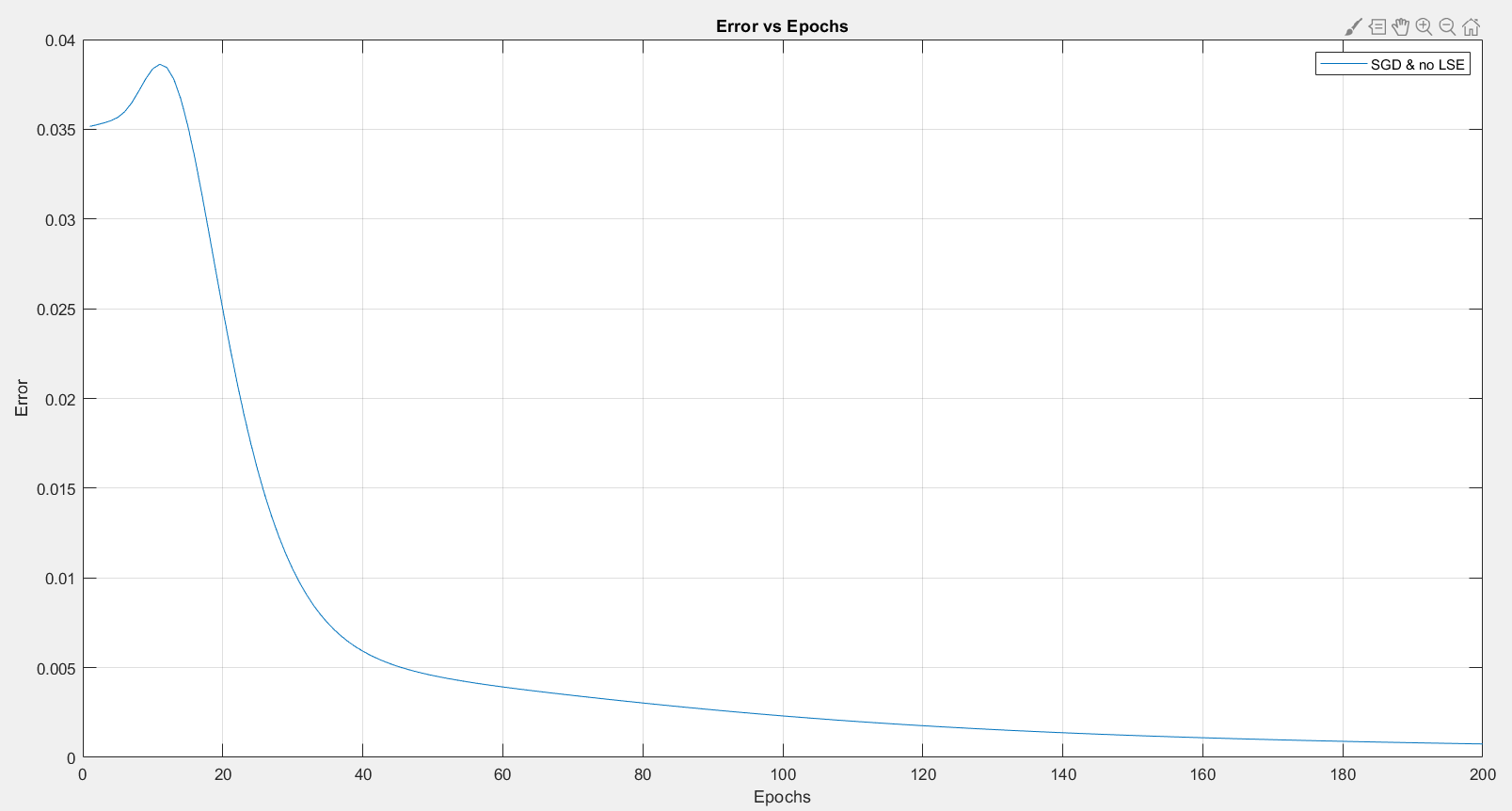
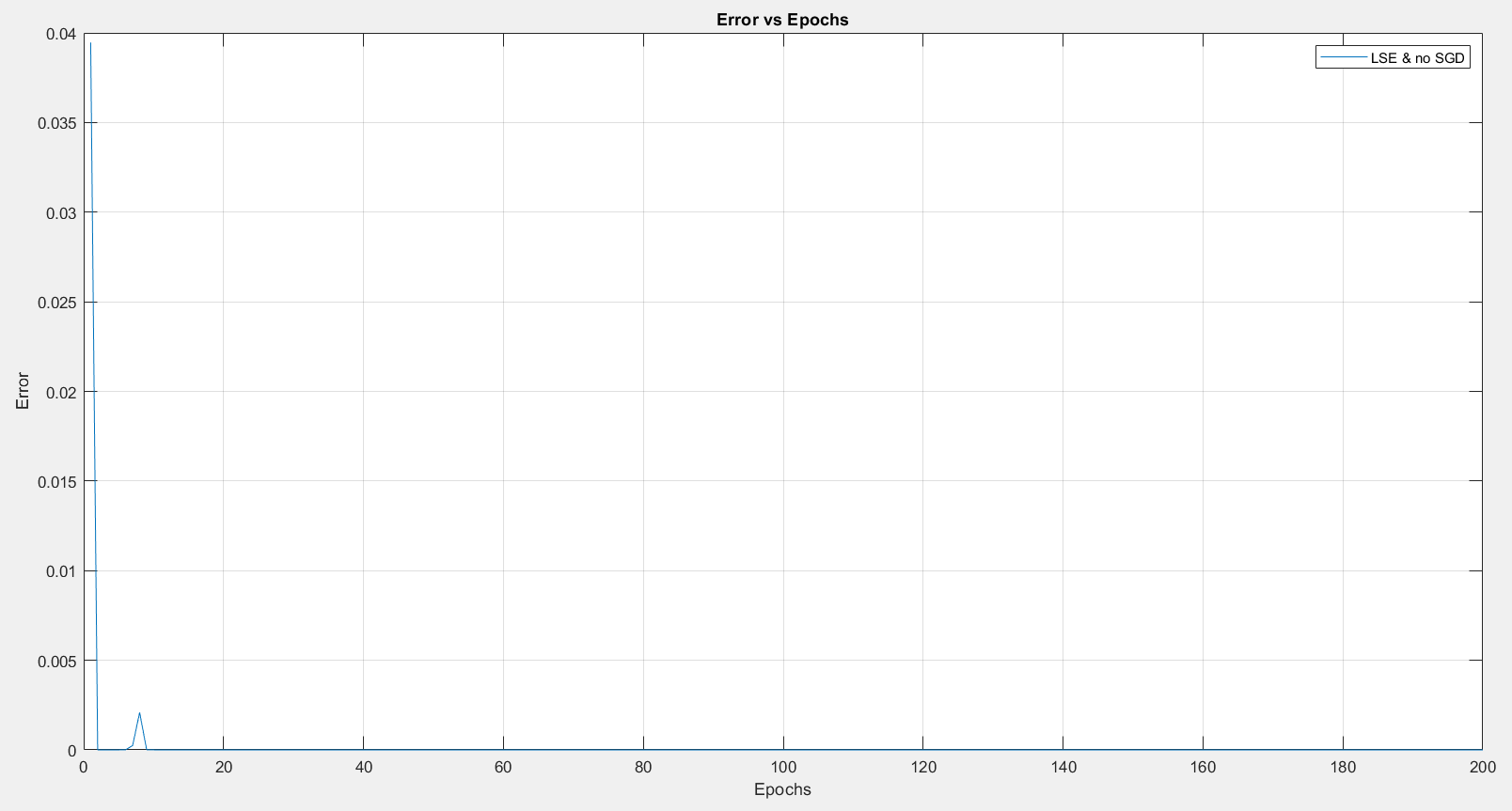
Note: as the output for each rule performs the dot operation so this derivative is also a vector.

The derivative of other membership function params are same as the process of this example.

# **Testing:**

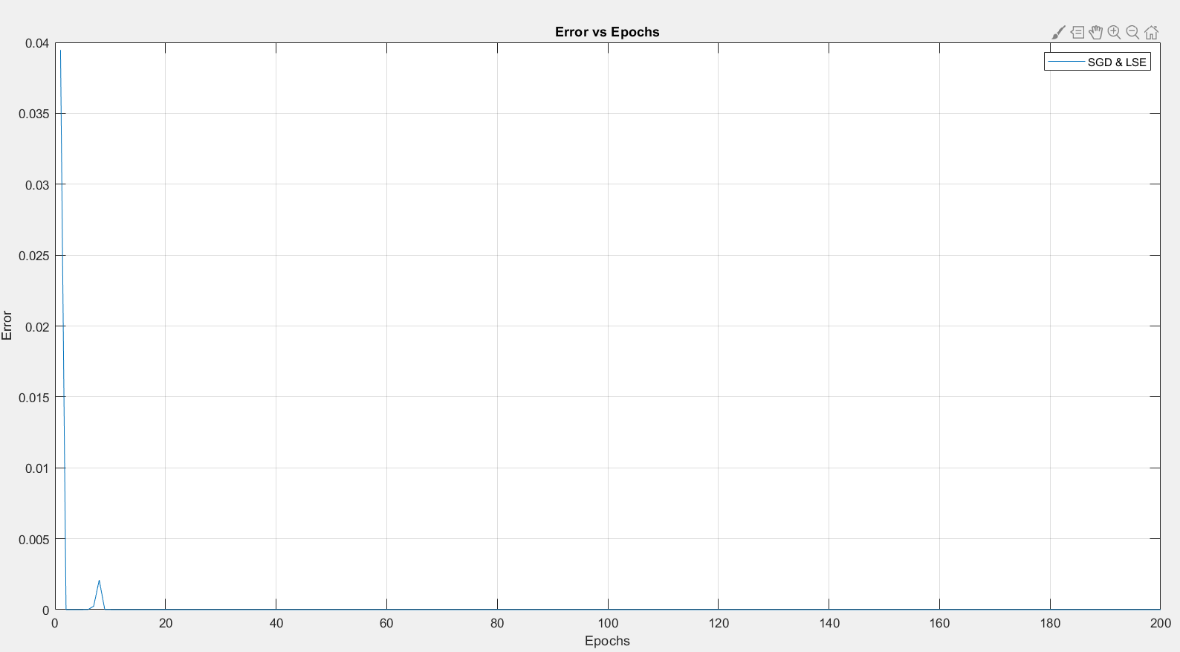
The fuzzy neuro network performs the hybrid training process for both premise params and consequent params. In this project, both SGD and RLSE are tested individually. The training data for both premise and consequent parameters are 100 pairs. To see the differences of result, SGD & no LSE is performed at first, in which only SGD is used to train the premise parameters and keep all initial consequent parameters. In the next step, RLSE & no SGD is performed and it keeps all default premise parameters and training all consequent parameters with Recursive Least Square Estimator. Finally, both SGD & LSE are used to training whole network. And then, 3000 data pairs are used to predicts the result with this model.

# **Result Analysis:**

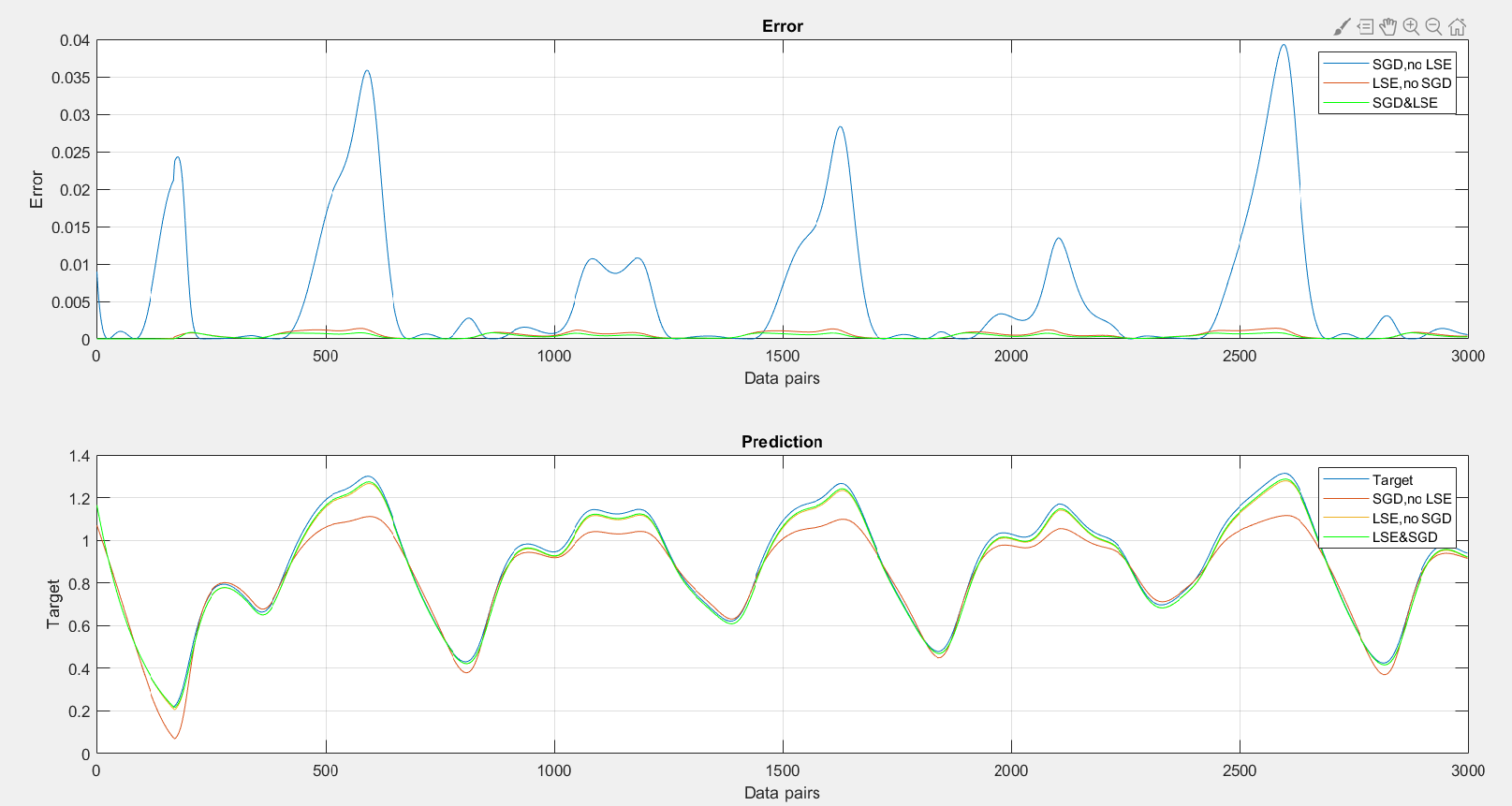
1. SGD & no LSE, the error decreases to under 0.0007 after 200 epochs and it converges relatively slow and the accumulated error in each epoch is still quite large.
2. LSE & no SGD, the model converge super fast and the accumulated error is less than 1e-6.

In the 2nd condition, recursive LSE is used to update the consequent parameters by 100 data pairs, but all the premise parameters of membership functions all keep as default.

1. LSE & SGD, both of techniques are used to train the model and the plotting is pretty mush similar as the LSE & no SGD. The model converges super fast but with the even smaller accumulated error. The major difference between condition 2 and 3 will be shown in the prediction plotting.



1. The prediction is done by 3000 data pairs. In the prediction plotting, the blue line is the target curve generated by the data pairs. The red curve is the model prediction with SGD training only. The yellow curve is the model with LSE & no SGD, and the green curve is the mode prediction with SGD & LSE. In the error plotting, the model with SGD & no LSE has the largest error which is the blue line. Both of the models trained by LSE has the maximum error around 0.0005 and the model trained by both LSE & SGD can be even more accurate.



# **Conclusion:**

According to the analysis result, RLSE makes a real difference in the training process in the fuzzy neuro network and SGD even decreases the error but it cannot optimize the model only. From SGD & no LSE plotting, the model converges after 200 epochs but the max error is up to 0.04 in the prediction plotting. The reason is that the range of membership function is between 0 and 1, but part of data pairs have the over 1 and SGD cannot minimize the error between output and target if the output function itself has the limitation. If the data set can be modified a bit and make it into [0,1], the training model will be even better.

Advantages of NFNN are hybrid training process involving both SGD and RLSE. Even SGD converges slowly but RLSE compensates this drawbacks, so it converges pretty fast. Also SGD can also contributes its optimizing on top of the RLSE method and it decreases the error even smaller. This can be seen by the prediction plot in which the green line is more close to the target and error is even smaller.

Limitation of NFNN are on the RLSE. This project has only 32 consequent params which means the model is not a big one at all. Before implementing RLSE algorithm, the initial consequent params must be provided and both LSE and RLSE all involves the inverse of the matrix. If the model involves over 100 or even 1000 consequent params, the inverse calculation can be really slow and also decreases the accuracy and boosts the calculation cost. Furthermore, this project is using the TSK-1 model which is a 1 order polynomial equation but if a higher order of equation is used to solve regression or classification problem. The LSE has to be performed by SGD as well and then RLSE cannot be implemented at all.

# **Reference:**

[1]  Jang, Jyh-Shing Roger, Chuen-Tsai Sun, Eiji Mizutani Neuro-Fuzzy and Soft Computing 1997.