Parisa Suchdev
Generalized Regression Neural Network - GRNN

I loved working on this NN because it's one pass and weights are not scary! I closely followed one of the papers provided called "ASSESSING LINKAGES IN STREAM HABITAT, GEOMORPHIC CONDITION, AND BIOLOGICAL INTEGRITY USING A GENERALIZED REGRESSION NEURAL NETWORK1" and it's amazingly written. Easy to read and implemented.

My understanding of algorithm: Unsupervised, nonparametric, relatively fast as its only feedforward, trains quickly on sparse data. The first layer is input layer that is fully connected to the next layer. The job of layer 1 is to distribute the scaled patterns to the second layer (Pattern Unit Layer). Second layer contain n nodes and n is defined as the number of samples in the training set. The distance is calculated between all sets of training samples(weights) and input pattern. Those distances are then fed into non-linear kernel that is exponential activation function that has a parameter called sigma. The only parameter that requires tuning to best fit the model. The third layer is called summation layer that has two nodes. First node (A) is the summation of outputs of the hidden layer and second node (B) is the summation of activation outputs of hidden layer. The last layer is the division of A/B is the output and estimation of y.

For the coding part: I didn't do scaling although it's a non-parametric algorithm but the reason I didn't do it is because there was only one feature in the training set to feed and y is dependent on X (making this model parametric?). I coded up all the layers and then calculated the error with RMSE for several sigma values. The goal is gradually decrease sigma until the RMSE low on testing set.
Training dataset:
My first test was with 4 sigma values in figure 1 and we can see the overfitting when as the sigma values approaches close to zero and underfitting as it approaches close to 1.

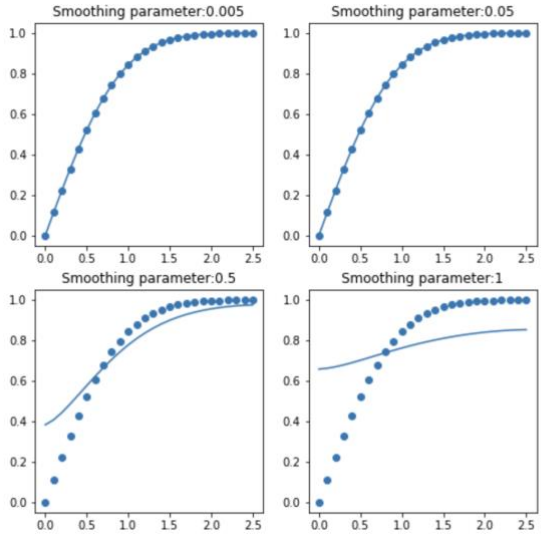

Figure 1 Predicted values plotted over input

Testing dataset (figure2): It worked well for interpolation! 0.5 value of sigma generalized well but it would be good idea to iterate model over bunch of values between 0.1 to 1.0 and choose the one that has lowest RSME on testing dataset. GRNN is speedy, but is there a tradeoff with the accuracy? Because it is based on conditional approximation y given X. Maybe will have to test a bigger dataset.

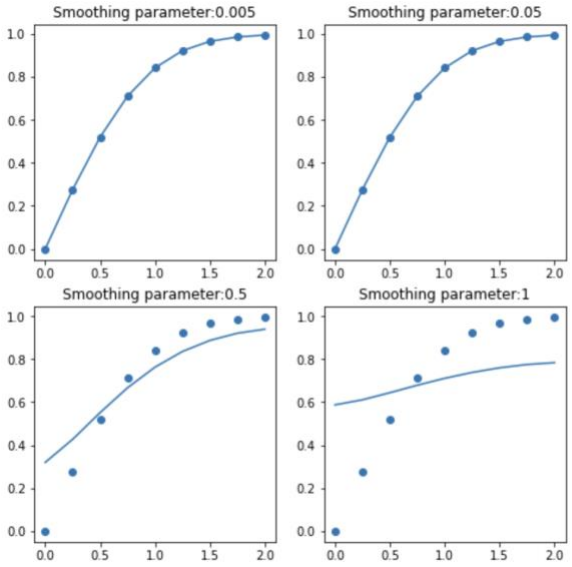

Figure 2

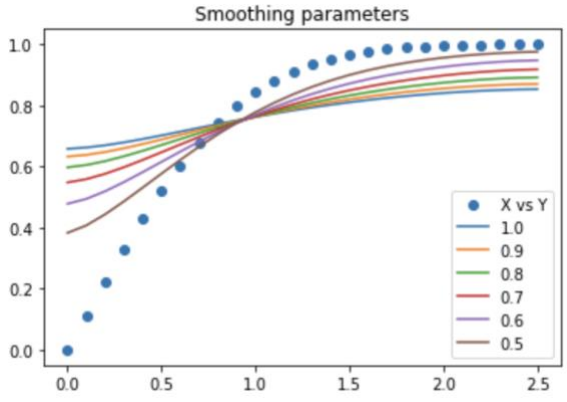

Figure 3 sigma: 0.5 to 1

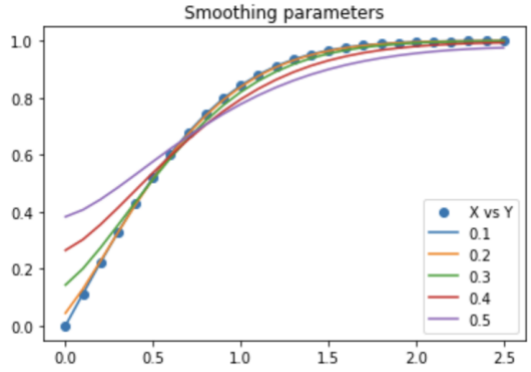

Figure 4 0.1 to 0.5

Different smoothing parameters were testing just to test the statement written in paper:
"Typically, the values range between 1.0 and 0.5. In the worst-case scenario, the algorithm would need to iterate over six sigma values (r = 1.0, 0.9, 0.8, 0.7, 0.6, and 0.5)"
Will all the values from 0.1 – 1, 0.5 fits best and generalizes well.