RECIPE FOR AN EXQUISITE NEURAL NETWORK:

PREDICTION OF CARS FUEL EFFICIENCY

WHAT WE NEED:

- We will be using the “Auto MPG Data Set” of UCI Repository.

You can find the original files in this webpage:

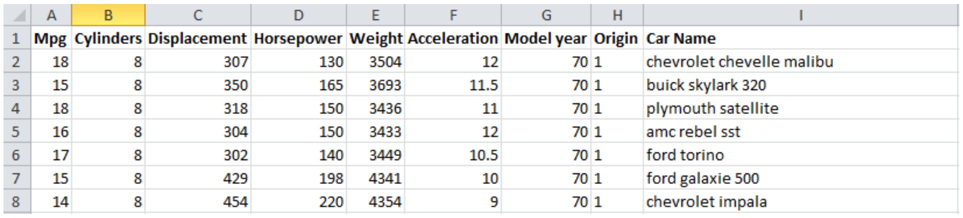
https://archive.ics.uci.edu/ml/datasets/auto+mpg

- Microsoft Excel

HOW TO MAKE A NEURAL NETWORK?

Open the *.DATA file with excel. Yes, I know, the format [*.data] is confusing. Change it to [*.txt] and import it as text with defined width.

Add a Header with the labels. (Remember, only one row of header)



	A	B	C	D	E	F	G	H	I
1	Mpg	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model year	Origin	Car Name
2	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu
3	15	8	350	165	3693	11.5	70	1	buick skylark 320
4	18	8	318	150	3436	11	70	1	plymouth satellite
5	16	8	304	150	3433	12	70	1	amc rebel sst
6	17	8	302	140	3449	10.5	70	1	ford torino
7	15	8	429	198	4341	10	70	1	ford galaxie 500
8	14	8	454	220	4354	9	70	1	chevrolet impala

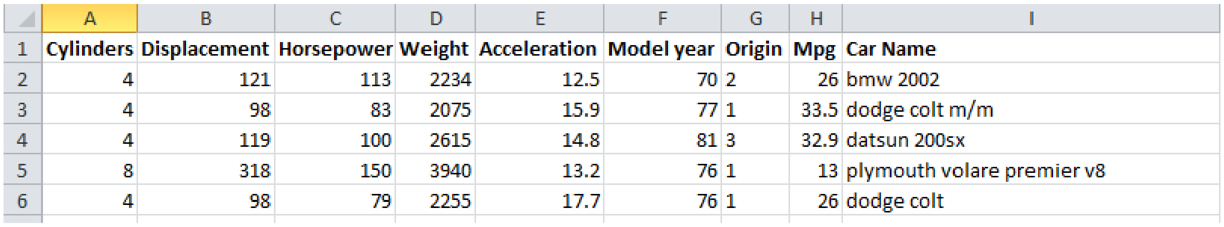
Something like this should we in your sheet.

NOW YOU HAVE TO CLEAN WELL YOUR DATA:

You can find a few rows with non-numerical values with an easy filter (Just a clue, they are identify whit an [“?”]). Delete those rows. We are going to ignore those values.

IN ORDER TO MAKE PROGRESS:

“Car name” will not be used during the training. You can delete it or leave it at the right. But MPG must be at the right of all the inputs. Something like this will work. Remember the configuration of the inputs! The formula is going to request it in the same order.



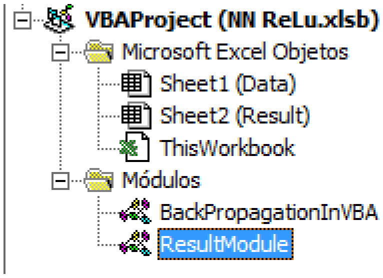
	A	B	C	D	E	F	G	H	I
1	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model year	Origin	Mpg	Car Name
2	4	121	113	2234	12.5	70	2	26	bmw 2002
3	4	98	83	2075	15.9	77	1	33.5	dodge colt m/m
4	4	119	100	2615	14.8	81	3	32.9	datsun 200sx
5	8	318	150	3940	13.2	76	1	13	plymouth volare premier v8
6	4	98	79	2255	17.7	76	1	26	dodge colt

VBA: WHERE THE MAGIC HAPPENS

Open the Visual Basic Application in the Programmer TAB. **You still don't have a Programmer TAB!** Don't worry; your secret is safe with me. Go to Options > Customize Ribbon and activate it.

Rename the sheets and paste the Code found in GitHub in two modules:

It should be something like this.



VBAProject (NN ReLu.xlsb)
Microsoft Excel Objetos
Sheet1 (Data)
Sheet2 (Result)
ThisWorkbook
Módulos
BackPropagationInVBA
ResultModule

SET THE HIPERPARAMETERS:

LET'S CHECK ONE BY ONE. (YOU CAN FIND THIS IN THE PRIVATE SUB INITIALSETTINGS)

n_hid_layer = 'Number of hidden layers

I recommend starting with only one. And start adding layers. It's much more stable.

x_features = 'Number of x features

There are 7. (Number of Cylinders, Displacement, Horsepower, Weight, Acceleration, Model Year, Origin)

y_features = 'Number of y features

Only one, MPG, if you use another with more than one output remember that it will return a range. Accept the formula with Ctrl-Shift-Enter.

n_neurons_hid_layer(x) = 'Number of neurons in the hidden layer x (change the x for as many layers you have set)

I will start with 7. Just because is a pretty number.

SizeOfDataBase = 'The size of the database

Once you clean it there are 392 rows. You can save some as Test with the same format in the "Test" Sheet

I will save 381 rows for training and 11 for Test.

In the low-dimensional input spectra, overfitting is not very common because with a smaller data set you will have better information density.

SizeOfTestSet = 'The size of the Test Set

See SizeOfDataBase

UploadOldModel = 'Train more an existing model.

It's the first time here. So the answer is False.

ChangeArquitecture = 'It's common, and recommended, start with a shallow neural network and start increasing/decreasing neurons/layers. If it's true, will recycle old weights and initialize new ones.

It's supposed to be False. But since UploadOldModel is False, it doesn't matter.

SamplesInBatch = 'Number of Samples on each Batch. Maximum of 1024

Leave it at maximum. If you want to try mini-batch GD change it. I'm afraid for those stochastic lovers that the value of one it's not available.

NumberOfIterations = 'Number of iterations

I will start with 500. To see how everything is going.

Beta1 = 'First parameter of ADAM optimization

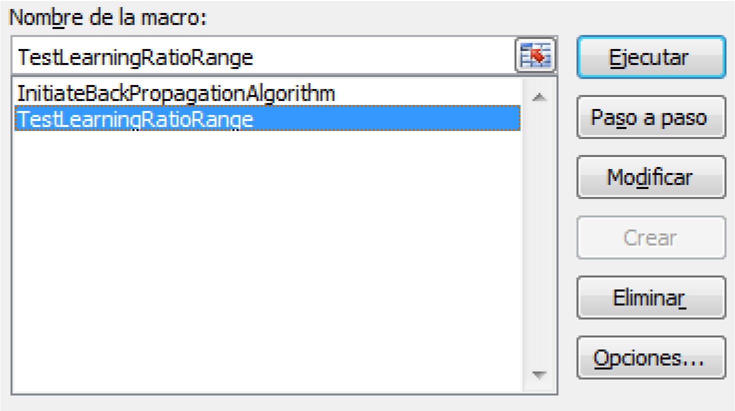
0.9 is recommended.

Beta2 = 'Second parameter of ADAM optimization

0.99 is recommended.

LearningRatio = 'You can perform a range test every time you want. This value will depend on the dataset and NN architecture.

Let's perform a Range Test. It's very important to study the response of the learning rate in your model.

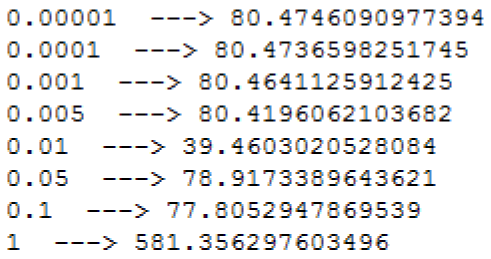


Nombre de la macro: TestLearningRatioRange

The result of the Test Range will be displayed in the immediate window. If you don't have it active press Ctrl + G.

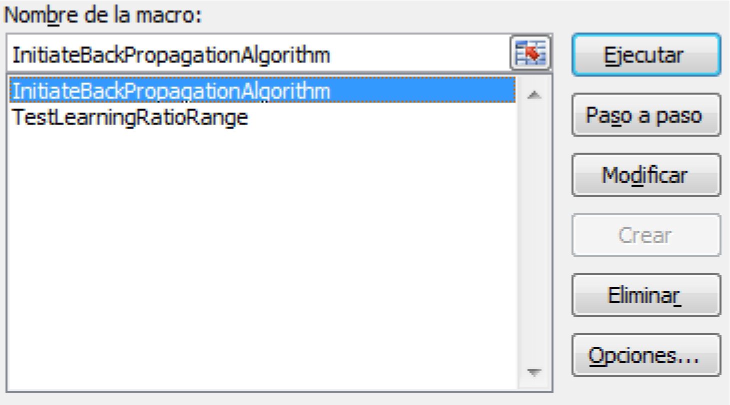
With a Test Range you can see the relationship between the squared error and the learning ratio in only 10 iterations. Our best result was in 0.01. A common rule of thumb is to divide this value by 10.

0.001 will be stable and effective.



```
0.00001  ---> 80.4746090977394
0.0001  ---> 80.4736598251745
0.001  ---> 80.4641125912425
0.005  ---> 80.4196062103682
0.01  ---> 39.4603020528084
0.05  ---> 78.9173389643621
0.1  ---> 77.8052947869539
1  ---> 581.356297603496
```

Let's set the learning rate and start training.



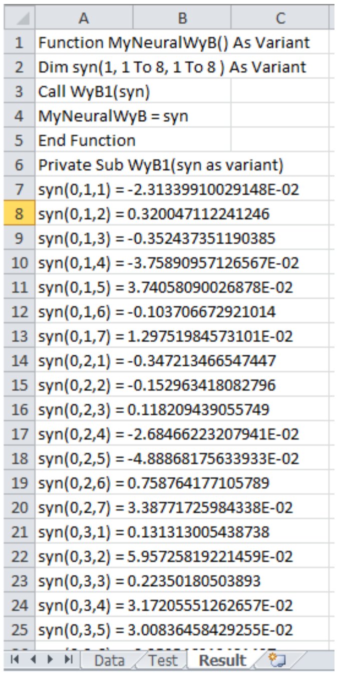
Nombre de la macro: InitiateBackPropagationAlgorithm

- InitiateBackPropagationAlgorithm
- TestLearningRatioRange

In the immediate window you will see the squared error evolution. If everything is Okie Dokie the number will decreasing.

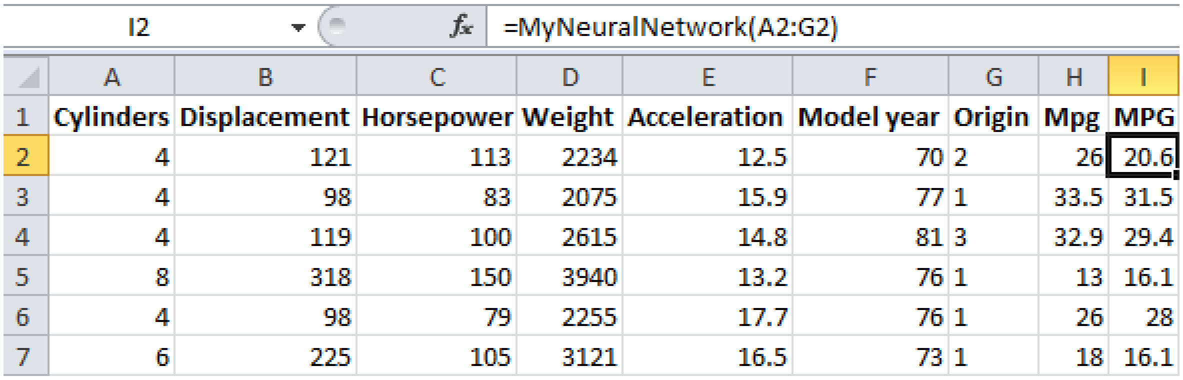
```
Train Error 1203.16823190972     Test Error 31.7404955385136     Iter: 0
Train Error 1102.56900753557     Test Error 29.2180229967748     Iter: 100
Train Error 1083.14751123235     Test Error 28.7137569474042     Iter: 200
Train Error 1052.30102764346     Test Error 27.924070312828     Iter: 300
Train Error 1004.29297796365     Test Error 26.6540595809408     Iter: 400
Train Error 931.96500783596     Test Error 24.7138377483672     Iter: 500
Train Error 827.546103564878     Test Error 21.9049957728768     Iter: 600
Train Error 682.500944731316     Test Error 18.0334914158763     Iter: 700
Train Error 495.752188986818     Test Error 13.2302902596418     Iter: 800
Train Error 300.050571434527     Test Error 8.07068876673788     Iter: 900
Train Error 144.952388530535     Test Error 3.82486820760773     Iter: 1000
```

You heard 3 beeps? You see how the squared error has decreased? That mean that we are ready! Go to the “Result” Sheet and copy the content in the module “Result Module”.



	A	B	C
1	Function MyNeuralWyB() As Variant		
2	Dim syn(1, 1 To 8, 1 To 8) As Variant		
3	Call WyB1(syn)		
4	MyNeuralWyB = syn		
5	End Function		
6	Private Sub WyB1(syn as variant)		
7	syn(0,1,1) = -2.31339910029148E-02		
8	syn(0,1,2) = 0.320047112241246		
9	syn(0,1,3) = -0.352437351190385		
10	syn(0,1,4) = -3.75890957126567E-02		
11	syn(0,1,5) = 3.74058090026878E-02		
12	syn(0,1,6) = -0.103706672921014		
13	syn(0,1,7) = 1.29751984573101E-02		
14	syn(0,2,1) = -0.347213466547447		
15	syn(0,2,2) = -0.152963418082796		
16	syn(0,2,3) = 0.118209439055749		
17	syn(0,2,4) = -2.68466223207941E-02		
18	syn(0,2,5) = -4.88868175633933E-02		
19	syn(0,2,6) = 0.758764177105789		
20	syn(0,2,7) = 3.38771725984338E-02		
21	syn(0,3,1) = 0.131313005438738		
22	syn(0,3,2) = 5.95725819221459E-02		
23	syn(0,3,3) = 0.22350180503893		
24	syn(0,3,4) = 3.17205551262657E-02		
25	syn(0,3,5) = 3.00836458429255E-02		

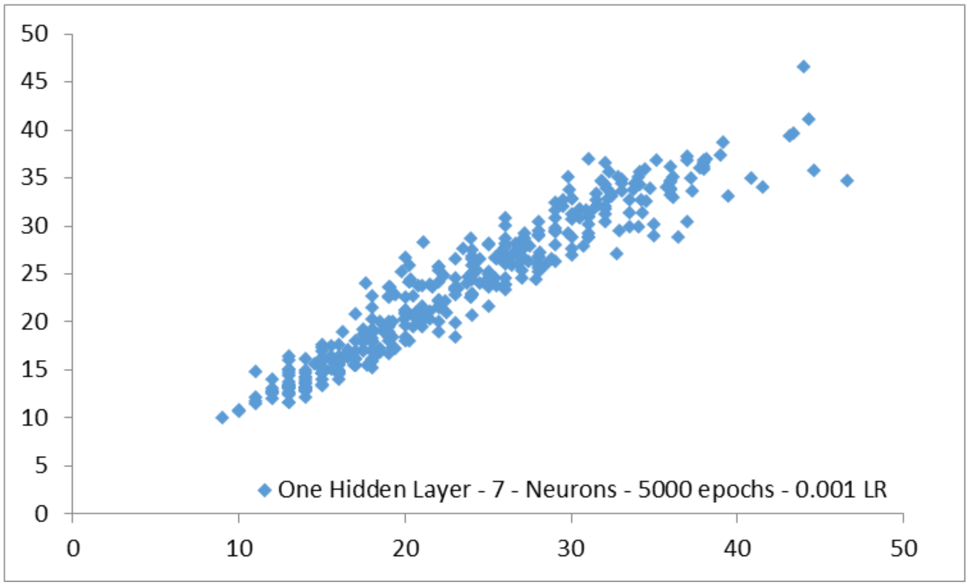
Now a Formula named **myneuralnetwork** is ready to use in your spreadsheet!



I2 f_x =MyNeuralNetwork(A2:G2)

	A	B	C	D	E	F	G	H	I
1	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model year	Origin	Mpg	MPG
2	4	121	113	2234	12.5	70	2	26	20.6
3	4	98	83	2075	15.9	77	1	33.5	31.5
4	4	119	100	2615	14.8	81	3	32.9	29.4
5	8	318	150	3940	13.2	76	1	13	16.1
6	4	98	79	2255	17.7	76	1	26	28
7	6	225	105	3121	16.5	73	1	18	16.1

Let’s compare the prediction with the given values:



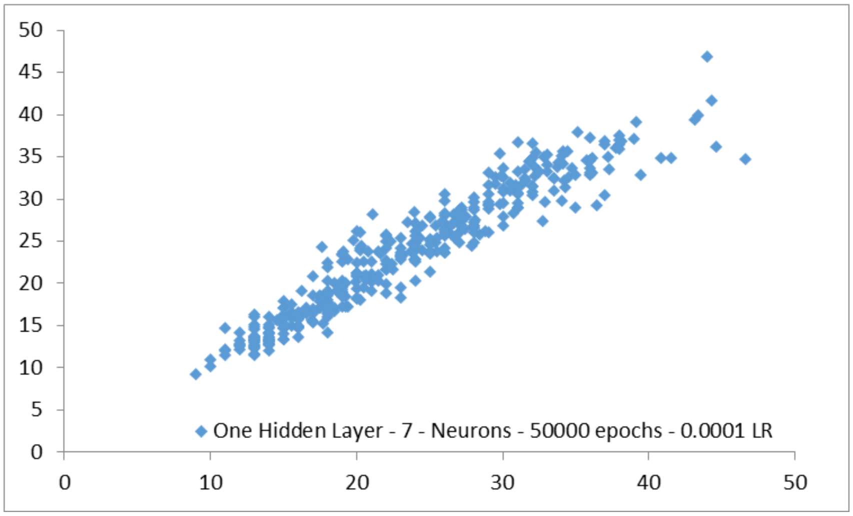
YOU WANT TO TRAIN MORE? NO PAIN NO GAIN?

Just change UploadOldModel to true and keep training.

A learning ratio reduction is commonly recommended. Or just perform another “RangeTest”!

I will go with 0.00005. Because I feel lucky!

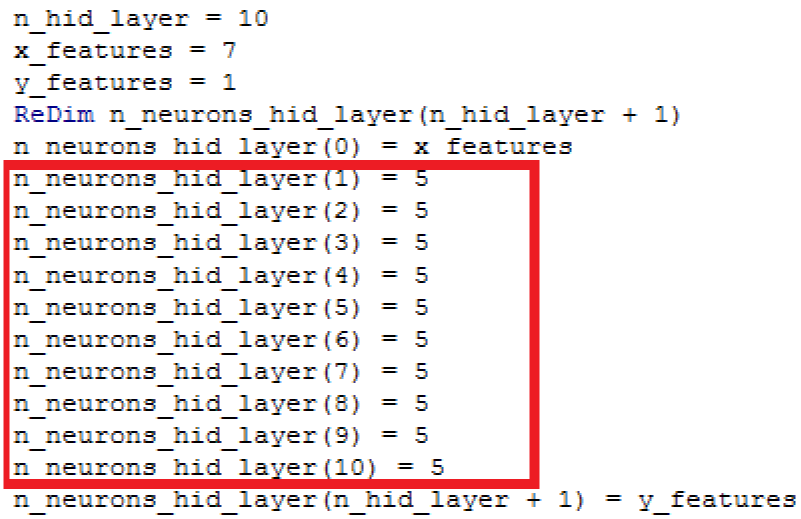
Let’s see after 45500 iterations! Apparently we have reached this model full capacity or at least, a local minimum.



YOU WANT TO GO DEEPER? WE ARE NOT HERE TO SEE ONLY ONE HIDDEN LAYER!!!

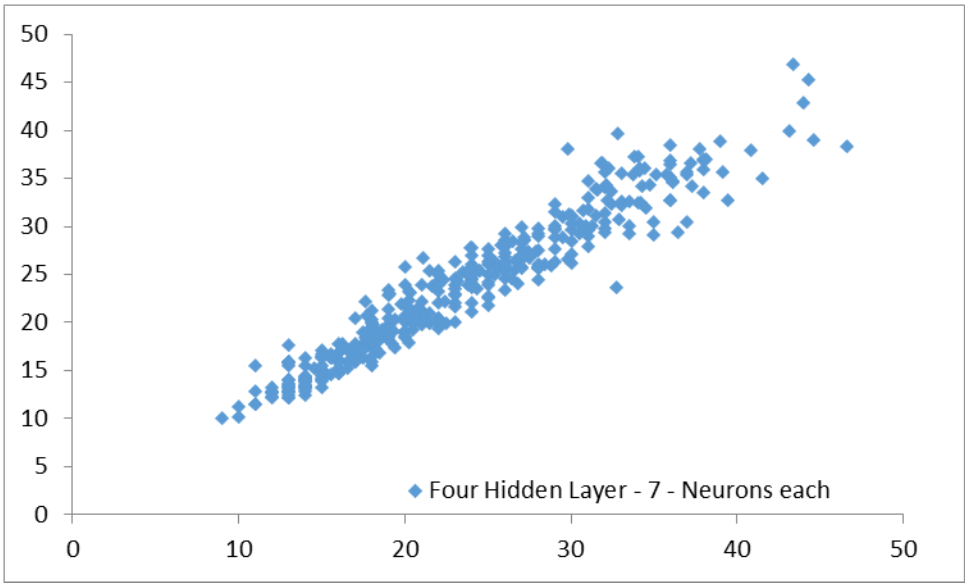
Make sure that “ChangeArquitecture” is true and change “n_hid_layer”. If you don’t want to explode your neural network adding layers progressively is recommended.

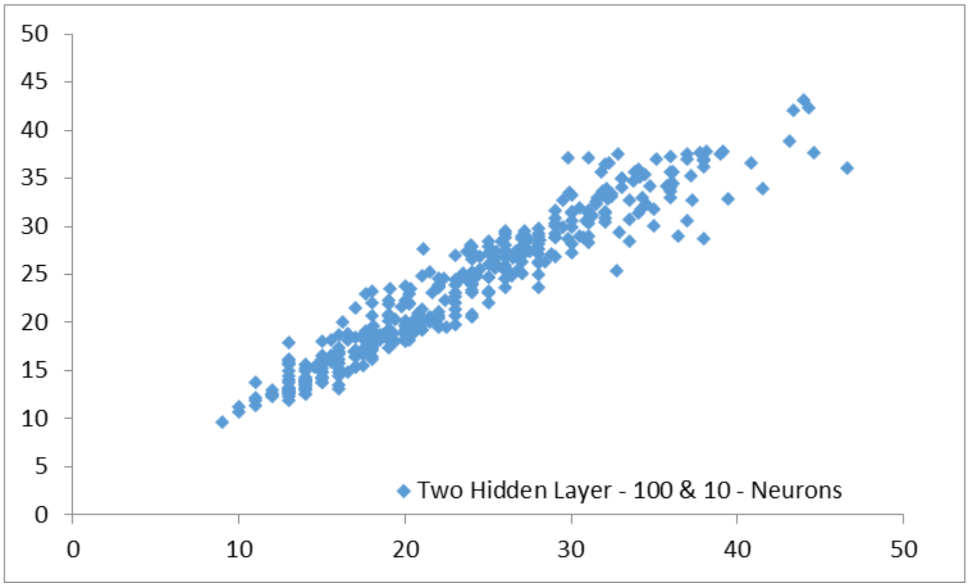
ATTENTION: REMEMBER TO ADD THE NUMBER OF HIDDEN LAYER OF EVERY LAYER!

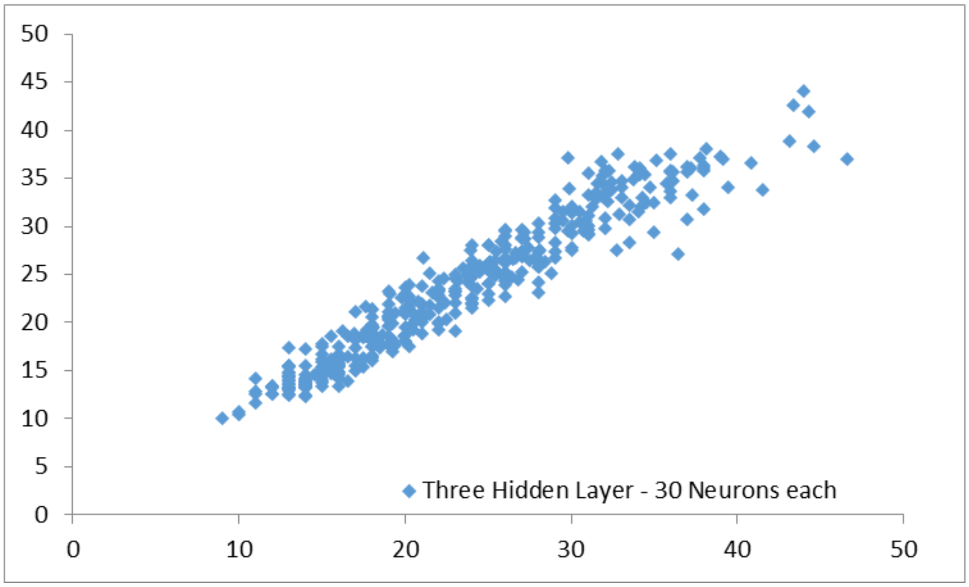


```
n_hid_layer = 10
x_features = 7
y_features = 1
ReDim n_neurons_hid_layer(n_hid_layer + 1)
n_neurons_hid_layer(0)  = x_features
n_neurons_hid_layer(1)  = 5
n_neurons_hid_layer(2)  = 5
n_neurons_hid_layer(3)  = 5
n_neurons_hid_layer(4)  = 5
n_neurons_hid_layer(5)  = 5
n_neurons_hid_layer(6)  = 5
n_neurons_hid_layer(7)  = 5
n_neurons_hid_layer(8)  = 5
n_neurons_hid_layer(9)  = 5
n_neurons_hid_layer(10)  = 5
n_neurons_hid_layer(n_hid_layer + 1)  = y_features
```

Here are some predictions. They were taken on a rush. No optimum intended.




50
45
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
◆ Two Hidden Layer - 100 & 10 - Neurons


50
45
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
◆ Three Hidden Layer - 30 Neurons each