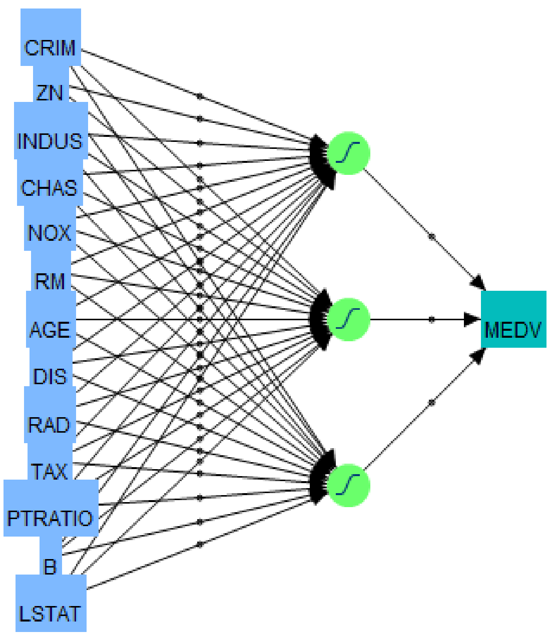
Neural Networks for Regression

This tutorial assumes that you know how to build ANN Classification model with SPSS Modeler
Please read the Appendix (description of the dataset) before going through this tutorial.

The Artificial Neural Network learning that we used for Classification can also be used for Regression. It is important to note that when learning for Regression, the activation function at the output layer cannot be a **Sigmoid** function because the model needs to predict a numeric value. In fact, the activation function at the output layer for ANN Regression is the **identity function**.

Diagram of a neural network for predicting the median home value in a census tract from the **Boston Housing Data** is shown below. Metadata about the Boston Housing dataset is posted in the Excel file that has the data. A summary of this dataset is in the appendix to this tutorial. As I mentioned in the class, the dataset that we are using has one less independent variable. Our dataset has **12 features** and **MEDV** as the target variable.

Note that this is a **Feed Forward** ANN.



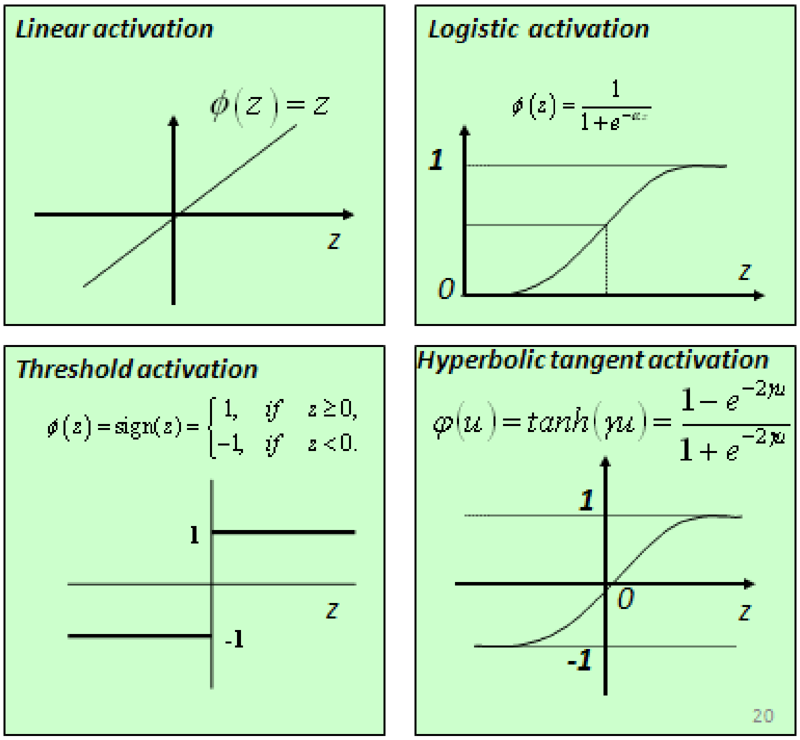
The neural network regression model looks like this:

$$y_i = \alpha + \underbrace{\sum_h w_h \phi_h(\alpha_h + \sum_{j=1}^{p} w_{jh} x_{ij})}_{f(\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_p})} + \epsilon_i$$

The neural network model in the above diagram has one hidden layer with 3 nodes, but it is possible to have additional hidden layers and more nodes per layer. The actual ANN that we will use is shown in the last section.

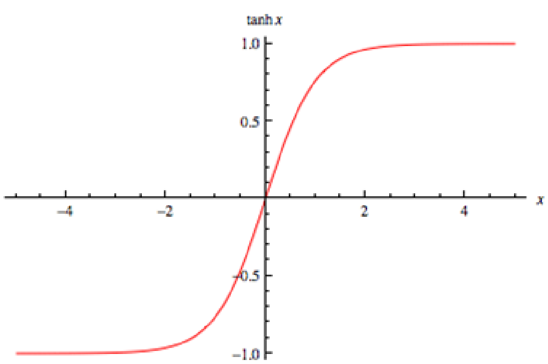
Activation Functions - $\phi(z)$

The $\phi(z)$ functions used are usually chosen from the following choices. In Deep Learning, RELU (Rectified Linear Unit – not shown here) is a choice that has become popular.



The hidden layer activation function used by SPSS Modeler is the **hyperbolic tangent function.** For regression problems it is important that the <u>final outputs be linear</u> as we don't want to constrain the predictions to be between 0 and 1.
The hyperbolic tangent function (used for hidden layer activation in SPSS modeler) is given by:

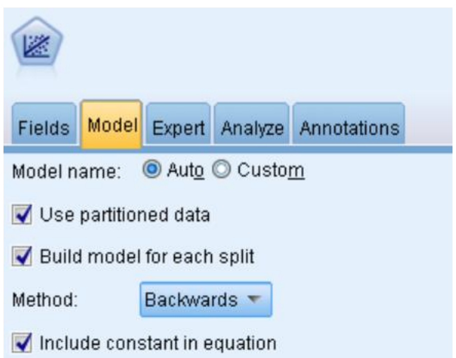
$$\phi(z) = \tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}}$$



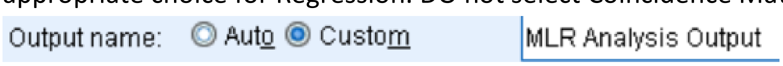
Multiple Linear Regression (MLR) Model in SPSS Modeler

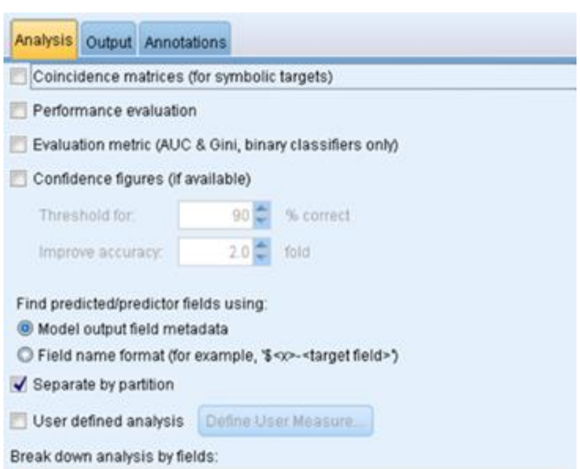
Any time you have a dataset where Neural Net Regression is applicable, MLR is also applicable. In fact, you are advised to try both and compare. If the performance of both the models is about the same, then you must go with MLR, which is a much simpler model, and much easier to explain to others. Fortunately, you can build both these models off of the same stream. The MLR node looks like.

In the Regression node- in the model tab-you have the option of specifying how the regression model should be built. The **Backwards** option corresponds to the backward stepwise regression that you learnt with excel. In this example we will use a **75%-25%** partition to specify Training-Testing partition. So, select the **use partitioned data** option.



Once the model is built, you can use **Analysis** node to evaluate the model. Coincidence Matrices is not an appropriate choice for Regression. DO not select Coincidence Matrices. Give a name to the Output Sheet.

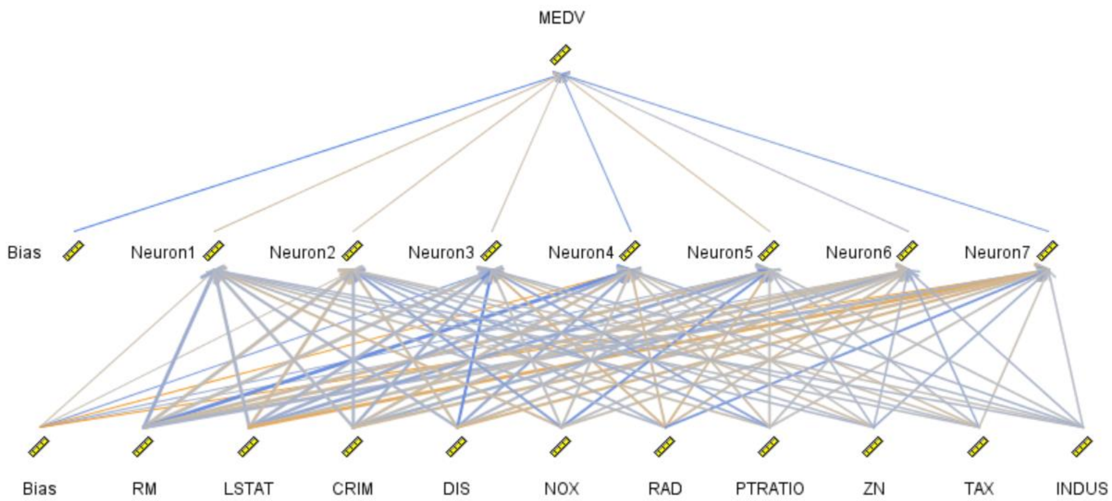




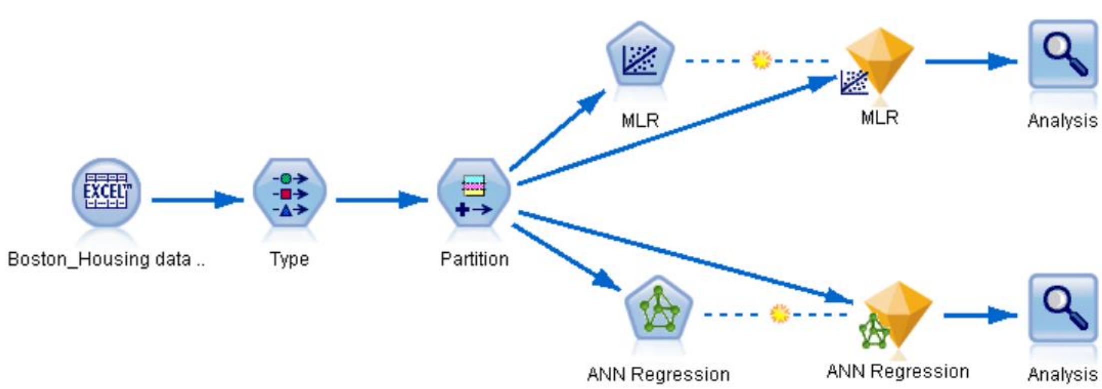
Choosing Neural Net Regression in SPSS Modeler

IN SPSS Modeler, once you choose a continuous variable as the **Target**, the software automatically builds a Regression model. The default network architecture (with a single Hidden layer) is chosen by SPSS modeler, and it makes sure that the Activation function on the output node is the Identity function. Whether to build a Regression model or a Classification model is determined by the data type of the **Target**. Rest of the model building process is the same for both classification and regression.

The following image shows the ANN for the Boston Housing Regression model. The input data provided **12** features (independent variables) and of course **MEDV** as the Target. The image below shows the network after the model was built. Two input nodes (**AGE**, **CHAS**) have been eliminated in the model building process.

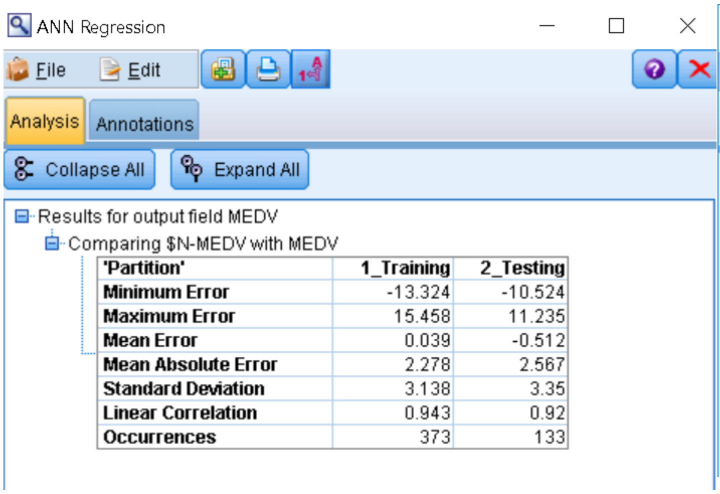


You can build both the MLR and ANN regression models as part of a single stream, as shown below.

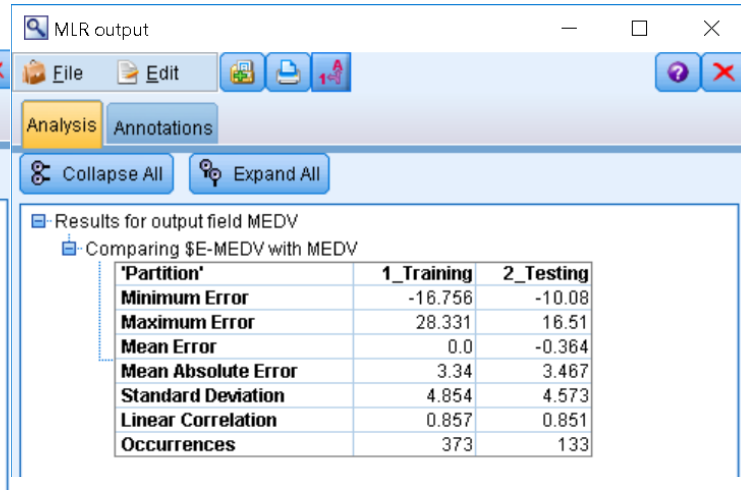


75% - 25% partition for Training-Testing was used in the above model

Evaluation:



'Partition'	1_Training	2_Testing
Minimum Error	-13.324	-10.524
Maximum Error	15.458	11.235
Mean Error	0.039	-0.512
Mean Absolute Error	2.278	2.567
Standard Deviation	3.138	3.35
Linear Correlation	0.943	0.92
Occurrences	373	133



'Partition'	1_Training	2_Testing
Minimum Error	-16.756	-10.08
Maximum Error	28.331	16.51
Mean Error	0.0	-0.364
Mean Absolute Error	3.34	3.467
Standard Deviation	4.854	4.573
Linear Correlation	0.857	0.851
Occurrences	373	133

The evaluation output on the left is from the Neural Net Regression. The Output on the right is from MLR regression. When there is more than one model, it is useful to give a name to your output report, which appears at the top left corner.

Mean Absolute Error: When you build a regression model and predict the value of the Target, there is always an error (or deviation), even if it is small. If these errors are small, then the predictions are strong.

> **Mean Absolute Error** is an important metric that is often used, and easy to understand. Take the absolute value of all the errors, and then find the average of the absolute errors.

Standard Deviation: is also known as Root mean squared error. It is the same measure you get with any MLR output.

Linear Correlation: is the linear correlation (correlation coefficient) between actual Target values and predicted Target values.

Metric	**MLR Regression**	**ANN Regression**
Mean Absolute Error	3.467	2.567
Standard Deviation	4.573	3.35
Linear Correlation	0.851	0.92
R Square	07242	0.8464

When you compare the two models, ANN Regression has performed better w.r.t. all the three metrics'. If you compare R-Square, ANN Regression is significantly superior.

Appendix: The Boston Housing Dataset

This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. (http://lib.stat.cmu.edu/datasets/boston), and has been used extensively throughout the literature to benchmark algorithms. The dataset is small in size with only 506 cases. Each case describes the characteristics of housing in a town (neighborhood) of Boston. It is worth noting that each case is not about an individual house. We have one record per town (neighborhood) and the target variable is the Median Price of houses in that town (in 1000s of dollars). The dataset is from a long time ago and the prices today are considerably higher.

The data was originally published by Harrison, D. and Rubinfeld, D.L. `*Hedonic prices and the demand for clean air*', J. Environ. Economics & Management, vol.5, 81-102, 1978.

There are **13** attributes in each case of the dataset. (The original dataset has one more attribute, which I removed). The attributes (features) in the dataset that we will use are:

1. CRIM per capita crime rate by town
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS proportion of non-retail business acres per town.
4. CHAS Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX nitric oxides concentration (parts per 10 million)
6. RM average number of rooms per dwelling
7. AGE proportion of owner-occupied units built prior to 1940
8. DIS weighted distances to five Boston employment centres
9. RAD index of accessibility to radial highways
10. TAX full-value property-tax rate per $10,000
11. PTRATIO pupil-teacher ratio by town
12. LSTAT % lower status of the population
13. **MEDV** **Median value of owner-occupied homes in $1000's ←-------- TARGET Variable**

The **Target** Variable is the last attribute, namely **MEDV**.

There are several quality problems with this dataset. We will only use the dataset as an example for MLR and ANN.

It is important to carry out a correlation analysis before conducting a Regression Analysis. The correlation matrix is as follows.

	CRIM	*ZN*	*INDUS*	*CHAS*	*NOX*	*RM*	*AGE*	*DIS*	*RAD*	*TAX*	*PTRATIO*	*LSTAT*	*MEDV*
CRIM	1												
ZN	-0.20046922	1											
INDUS	0.406583411	-0.533828186	1										
CHAS	-0.055891582	-0.042696719	0.062938027	1									
NOX	0.420971711	-0.516603708	0.763651447	0.091202807	1								
RM	-0.219246703	0.311990587	-0.391675853	0.091251225	-0.302188188	1							
AGE	0.352734251	-0.569537342	0.644778511	0.086517774	0.731470104	-0.240264931	1						
DIS	-0.379670087	0.664408223	-0.708026989	-0.09917578	-0.769230113	0.205246213	-0.747880541	1					
RAD	0.625505145	-0.311947826	0.595129275	-0.007368241	0.611440563	-0.209846668	0.456022452	-0.49458793	1				
TAX	0.582764312	-0.314563325	0.72076018	-0.035586518	0.6680232	-0.292047833	0.506455594	-0.534431584	0.910228189	1			
PTRATIO	0.289945579	-0.391678548	0.383247556	-0.121515174	0.188932677	-0.355501495	0.261515012	-0.232470542	0.464741179	0.460853035	1		
LSTAT	0.455621479	-0.412994575	0.603799716	-0.053929298	0.590878921	-0.613808272	0.602338529	-0.496995831	0.488676335	0.543993412	0.374044317	1	
MEDV	-0.388304609	0.360445342	-0.48372516	0.175260177	-0.427320772	0.695359947	-0.376954565	0.249928734	-0.381626231	-0.468535934	-0.507786686	-0.737662726	1