* Non-Linear Regression Models
  + Neural Networks (Anthony)
    - Description
      * Modeled by Neural network of the brain
        + Input-Dendrite (Artificial NN Input values)
        + Nucleus – (Inputs with unique weights and summed together, sigmoid functions are applied and passed threshold in 1 to many hidden layers
        + Output – Axon and terminal (Based on about becomes a 0 or 1 with Sigmoid function, Passthrough Synapse to next Dendrite (passed to next neuron)
      * Intermediate Observed or Hidden Variables
        + Linear combinations transformed by sigmoidal function

Sigmoidal functions

Transformed by a nonlinear function g()

* + - * + For P predictors there are H(P+1)+H+1 parameters

3(228+1)+3+1=691 parameters

**Neural networks were inspired by the way neurons work in the brain. At a high level the neuron Input dendrites work like the input values for a neural network. The Nucleus does the work and would be similar to where data is being passed through function and additional layers before it passed through to the output into the next neuron. The number of parameters can be calculated as listed. So if you had 228 predictors and 3 hidden networks that would equate to 691 parameters.**

* invvalue<-input1value + eachvar\*weight+...
* outvalue<-1/(1+exp(-iv))
* invvalue<-input2value+ov\*weight
* outvalue<-1/(1+exp(-iv2))

**Input values are passed along to be summmed and weighted. This is passed the sigmoid function in 2nd bullet point to generate the output value. The output data is then get passed to another network or neuron depending on configuration of the model and continues. If there are multiple input values they are all passed to all hidden layers and continues until data is passed to the output layer.**

* + - Computing examples

**In the computing example we are using the neuralnet package. The hidden values allows you to select the number of layers and neurons. A value by itself just give number of hidden layers. A vector value of c(2,1) produces 2 layers with 1 node. You can use the rep parameter to decide how many time you wan to run the model and when you plotting you can add the model and the rep number you want to view.**

**In the plot we are showing rep 2 of 3. The netresults gives us the probability of**

Ref: <https://www.youtube.com/watch?v=-Vs9Vae2KI0>

* + - Pros
      * Robust with noisy data
    - Cons
      * Less interpretable
      * Need longer training times
      * Neural Networks have a tendency to over-fit the relationship between predictor and response due the large coefficients
        + Fix

Early stopping

Weight Decay with regularization with lambda values 0-.1

Pre-filter the predictors

Use PCA prior to modeling to eliminate correlation

**One positive on Neural Network is that works well with noisy data which also makes it less interpretable and takes it longer to train. A major negative is its tendency to over fit relationships but there are a few ways to avoid this. Early stopping halts the optimization when errors increase. Weight decay which penalizes large regression coefficients. As regularization increase the model becomes smoother. Another approach is to prefilter predictors and remove any predictors with high correlation , since these models are usually adversely affected by this. We can also use PCA to eliminate predictors with high correlation.**

* + Multivariate Adaptive Regression Splines (Anthony)
    - Description
      * Creates 2 contrasted versions to of a predictor
      * 1 or 2 predictors at a time
      * Breaks predictors to 2 groups and models between
      * Hockey-stick(hinges)
        + Left-hand – values > 0 than cut point
        + Right-hand - values < 0 than cut point
        + Piece-wise linear model isolated portion of original data
        + Predictor/cut-point with smallest error
        + *Hing functions:X<a, h*(*x − a*) and *h*(*a − x*)

**For spline we begin with Multivariate Adaptive regression Splinesor MARS. Splines modeling breaks predictors into 2 groups and models the linear relationship between the 2 groups predictors and outcomes. It uses a cut point to create a hinge and a hockey stick functions of the original data.. The left side has have values greater than the cut point and the right has values less than the cut point. The new values of the functions are added to the linear model to estimate slopes and intercepts. Each function models an isolated portion of the data.**

**You use A linear regression model to determine a cut point from each predictor point. The predictor value with the lowest error is used as the cut point. This process can be repeated after the initial model is run to find another best fit until the user stop point is reached.**

* + - * Pruning used to remove parameters
        + The degree of the features that are added to the model and the number of retained terms.
        + The latter parameter can be automatically determined using the default pruning procedure (using GCV), set by the user or determined using an external resampling technique.
      * Pros
        + Model automatically conducts feature selection
        + Interpretability, each hinge feature is responsible for modeling a specific region in the predictor space using piecewise linear model.
        + MARS require very little pre-processing, transformation and filtering not needed.
      * Cons
        + Speed, other models may run faster

**The pruning process removes features that are not significant to the model. The big advantage of MARs is its ability to do automatic feature selection , but it comes at expense of speed.**

* + - * Other Splines

Ref: <https://www.youtube.com/watch?v=UDDXkffB-aE&t=329s>

* + - * + Polynomial Splines- continuous at the knot
        + Cubic Splines- continuous at the knot. Same as linear splines instead of power of 0 its 3.
        + Smoothing Splines

Splines without knots

Use smooth.splines() function in R

Find the function *g* that minimizes where *λ* is a nonnegative *tuning parameter*.

* + - * + GAM

The addition Spline model typ

* **Polynomial regression**. This is the simple approach to model non-linear relationships. It add polynomial terms or quadratic terms (square, cubes, etc) to a regression.
* **Spline regression**. Fits a smooth curve with a series of polynomial segments. The values delimiting the spline segments are called **Knots**.
* **Generalized additive models** (GAM). Fits spline models with automated selection of knots.
  + - Computing examples

**Here are some Spline examples. In this one we are using a smooth spline which . This uses a cubic polynomial linear with 3 knots Each vertical line is a knot is a knt. The greenline is the linear model with a continuous lin,e the red line is the smoothing spline with a degrees of freedom 16 which puts waves on the line. The purlple line show the cross validation information to pick the tuning parameter.**

**In the GAM example we are using gam with smoothing splines with the s parameter. This plot prodcues a plot for each of the terms in the GAM. And uses the degrees of freedom of for each level of the plot.**

**In this model you are using just a linear model with natural splines and lastly we use anova to compare 2 models.**