The weights and triggers of a neural network are critical pieces of the model that if not properly elucidated can introduce significant uncertainty into the learning operation. To start we consider a single weight acting on an input of information that passes along an activation function within a hidden layer in our structure.

An initial variable will be labeled “target\_stock” to start, although eventually additional ones can be called “target\_stock\_2” and so forth. This variable will be found by parsing through the incoming data and finding from a list of candidates the one that meets the conditions to be bound to this initial variable(s). This process will be called the weighing process, and it will made up of a series of filters through which the incoming list will be cleaned up with. These filters will be made up of the following 4 conditions:

* Where *S* is the stock pool, i.e DJI
* Where *p* is the price range = [5-45]
* Where *v* is the volume which is a maximum vertex
* Where *MA* is the moving average technical indicator

The function flow of this program will be quite straightforward and simple. At its core will be the triggering conditions which will weigh a list by filtering and sorting function calls.

While :

If *S <MA:*

By > then **sell**

Else If *S <MA:*

By > then **buy**

Where and are the respective sell and buy target rates above or below the moving average.

The get\_select\_execute master function will be called g\_s\_e. It will take a list of stocks streamed by a data provider and select a few of them that match our given and variables. For example, if our variable of 25% is broken through when the difference between the stock price and the moving average is greater than , our program would sell the stock. This is of course if our initial condition of a flat slope for our price function is met.

The function will trail by a fixed amount of given days, for example a month. If during this month the slope of the price function with respect to time flattens then we can look at our target buy and sell rates to trigger.

The String Library

In addition to our target variables, there will be label variables as well as variations of these in static or dynamic form. Examples of dynamic variables are broker tokens that refresh and update as they are issued through the authorization function. The initial target\_stock variable will be dynamic since its value will depend on the output of our activation function. “Target list” variables are the initial parameters and will be fixed. Examples of “real” variables would be the real time incoming market data being archived into static form for retrieval and analysis. As we can see our string library is starting to grow into a sizeable dictionary.

Computational Flow

The general flow of our algorithm is to get a list, set targets, select from the list based on our targets, and to execute a trade. The last portion of the learning environment will need to be discussed more broadly in another section. This is because the results will need to be broken down and the risk will need to be redistributed depending on our desired and variables, which are also fixed.

An XLS file of our program summarizes all of the above.