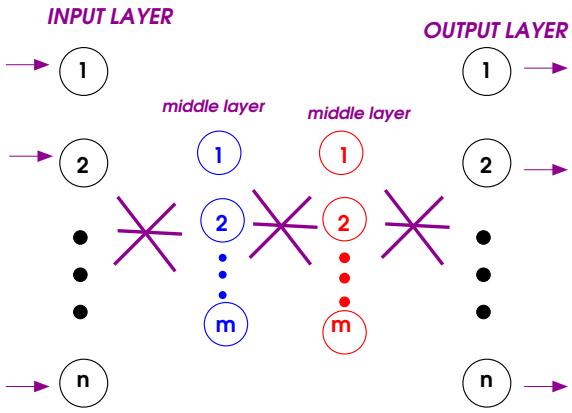
HW3, due October 31: Training a four network to predict asset returns

In this assignment we use the Russell data set (attached) with 947 assets over 504 days. Suppose we denote by $r_j(t)$ the returned attained by asset j on day $t+1$ (i.e. by looking at prices on days t and $t+1$). Let $r(t)$ be the vector with entries $r_j(t)$ for $j = 1, \ldots, 947$.

We want to use $r(t)$ so as to predict $r(t+10)$. We will use the first half of the data (i.e. the first 252 days) to train the network, and evaluate the trained network on the second set of 252 days. The network we will use is depicted next: In this picture, n = number of assets = 947. And m is a much smaller value. Your code should



work with m at least 50, but for debugging purposes you should try much smaller m and n. Each layer is fully connected to the next layer. You are free to use any of the activation units we discussed in class.

At any time t, if we feed to the input layer the vector $r(t)$, the output layer will produce the prediction for $r(t+10)$. Let us denote by $p(t)$ this prediction vector, i.e. the output produced by the network when the input is $r(t)$. It follows that the loss function is of the form

$$\sum_{t=1}^{242} \sum_{j=1}^{947} (r_j(t+10) - p_j(t))^2.$$

This is the function that we seek to minimize using the gradient-based method we discussed in class. The generalization error will likewise be:

$$\sum_{t=253}^{494} \sum_{j=1}^{947} (r_j(t+10) - p_j(t))^2.$$

You can implement this code in Python, for a maximum of 100 points. If you do it in C or C++ you get 30 extra credit points.

Questions welcome!