Assignment 3 - Report

STAT497

Presented to

Dr. Frederic Godin

By

Nam Phuong Nguyen (40011040)

Tony Yuan (40029336)

Spyros Orfanos (40032280)

Concordia University

April 15, 2018

QUESTION 1

**Part (a)** – See the R code

**Part (b)**

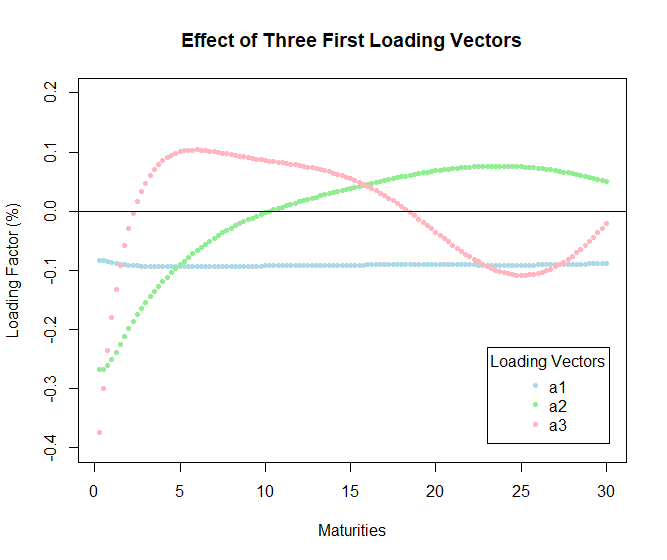
First, we find the eigenvalues and eigenvectors of the Variance-Covariance matrix using the eigen function. Then, the percentage of variance explained by each principal component is given as

. Therefore, the cumulative percentage of variance explained by each principal component is the cumulative density of the PVE, from which we get the formula .

Table 1: Cumulative percentage of variance explained by the six first principal components

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 0.9793518 | 0.9961754 | 0.9980159 | 0.9991078 | 0.9995772 | 0.9997982 |

**Part (c)**



In PCA, the first loading vector is associated with a parallel shift of the data, meaning that all scores are shifted up or down by the percentage presented. In this case, it is shifted by 0.1 downward. From part(b), this shift represents 97.94% of the variability in our yield curve.

The second loading vector is generally associated with the change in the steepness of our yield curve. Hence, an increase in the score will also increase the steepness of the yield curve. In this case, as the bond is closer to maturity, the slope of the yield curve is more affected by the score than for longer maturity bonds. From part (b), this factor represents about 1.72% of the variation in our yield curve.

The third loading vector explains the curvature in our yield curve. It reflects the up movement of the closest maturity dates and furthest maturity dates relative to the down movement of the middle portion of maturity dates and vice-versa. From part (b), this only represents less than 1% of the variability in our yield curve.

Reference used for part (c)

https://www.moodysanalytics.com/-/media/whitepaper/2014/2014-29-08-pca-for-yield-curve-modelling.pdf

**Part (d)**

To recover our initial data, we use the following formula to validate our method

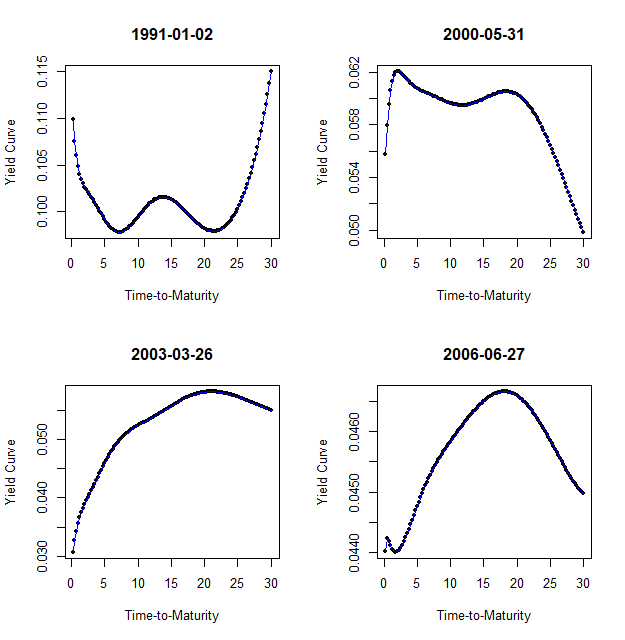
where evec are the eigenvectors of the variance-covariance matrix, YCmatrixdm are the demeaned original data and Allmeans are the means of each column of our original data. We perform a small validation step by subtracting each observation of allcomp (matrix containing the data calculated using all principal components) to our initial dataset called YCmatrix. Each value that we get are very close to 0 meaning that the yield curve from our principal component analysis and the actual dataset are extremely close and our method was well performed.

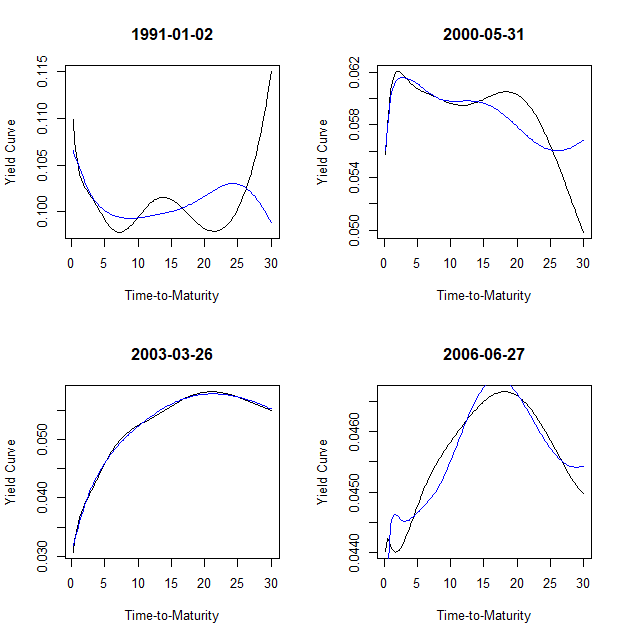
Before plotting the data with 3 and 5 principal components, we plot the original yield curve (black dots) with the PCA calculated yield curve (blue line). From Figure 1, we can see that they overlap perfectly, which again, supports our methodology.

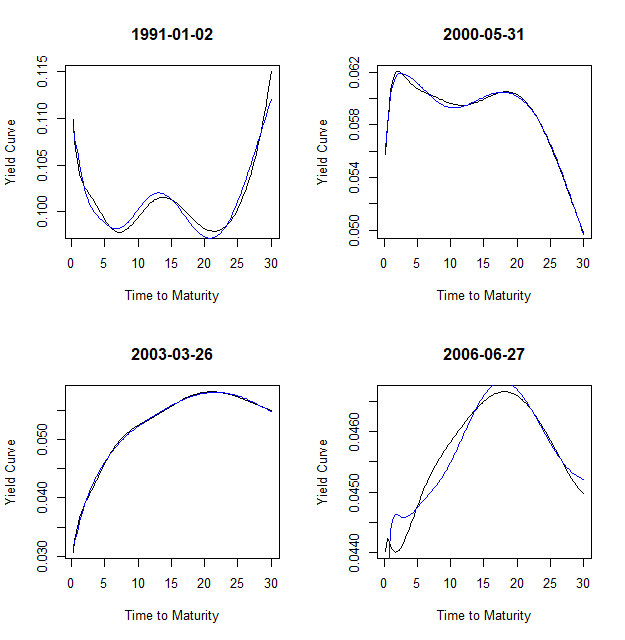
To get our predicted yield with 3 (and 5) principal components, we use the same formula as to validate but using only the first 3 (and 5) columns of our evec matrix, YCmatrixdm and Allmeans.

Using the first 3 principal components (Figure 2), we get these yield curves (blue) in comparison to the real yield curve (black dots) for each date. We can see that it performed better for more constant change as in 2003-03-26. For 1991-01-02 our predicted yield curve was way off compared to the actual data for most of the time-to-maturity. For 2000-05-31, the predicted yield curve resembles the actual data for most of the time-to-maturity but differs greatly for later maturity dates. 2006-06-27 has a predicted yield curve that imitates the shape of the real yield curve but is still a little bit off.

Using the first 5 principal components (Figure 3), we can see that it is already much closer to the real yield curve than using the first 3 principal components. The predicted yield curve is very similar to the actual yield curve when we look at 1991-01-02, 2000-05-31 and 2003-03-26. Again, as we can see for 2006-06-27, it varies more from the original than the other three but still follows the overall shape of the original curve.

Figure 1: Validation of the Principal Component Analysis

Figure 2: Prediction yield curve using three principal components

Figure 3: Prediction yield curve using 5 principal components

QUESTION 2

**Part (a)**

The seed number 1980 is used to generate 60 random points uniformly between the interval -2 and 2. We then generate 60 error terms that are Gaussian with mean 0 and variance 1. We calculate our response as the function.

**Part (b)**

We create the activation function and the derivative of this activation function that is used in our neural network. We define the activation function as and the derivative of the activation function as

Refer to code part B)

**Part (c)**

Two functions are made in our code to calculate the predictions.

Nneteval is a function that using the weight matrix w1 and w2 and the predictors, is able to make the prediction of the input vector.

nnetSSE will calculate the sum of squared error of the predictions using the Nneteval function described above.

Refer to code part C)

**Part (d)**

The gradient function that uses our wmat1, wmat2, responses and x vector to find the gradient. Now the gradient must be calculated using certain formulas found in the course notes.

Now this equation can be simplified to our specific situation. Our specific situation has.

1. “i” denotes the ith observation in the data set
2. w1  is a 5x2 Matrix
3. w2  is a 1x6 Matrix
4. a1  is a 5x1 Matrix
5. z is a 6x1 Matrix
6. a2  is a 1x1 Matrix
7. L = 2 since there are 2 layers
8. J = {1,2} since we have two weight matrices.
9. For J = 1, j is between 1 and 5 while k is between 0 and 1
10. For J = 2, j is 1 and k is between 0 and 5
11. g(x) = x thus g’(x) = 1.

Now the partial derivative can be decomposed as well.

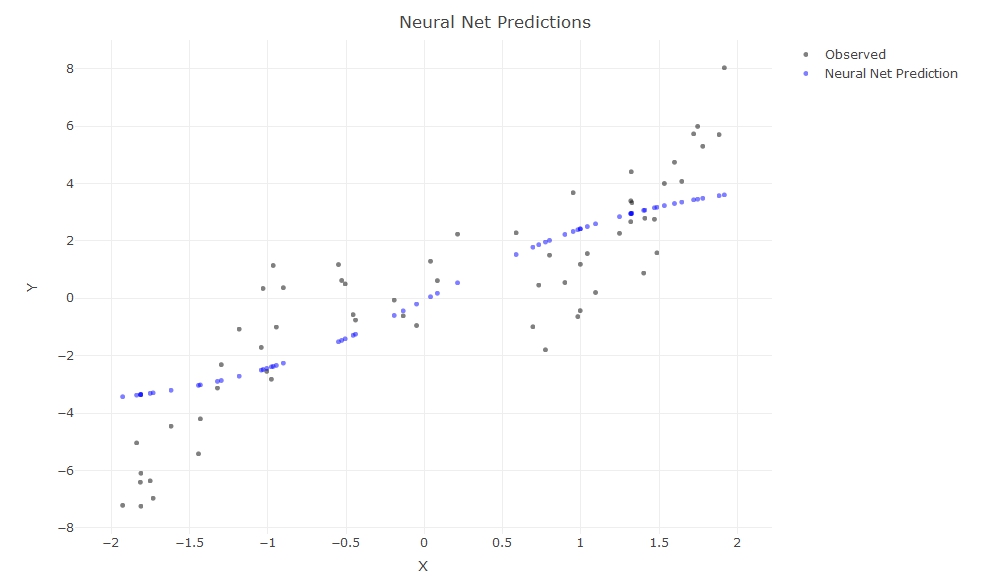
We were able to determine what the values were for these 2 partial derivatives based on the functions we were able to determine. These 4 functions for the different cases that will be used in computing our backpropagation algorithm

Refer to code in part D) to see the use of algorithm.

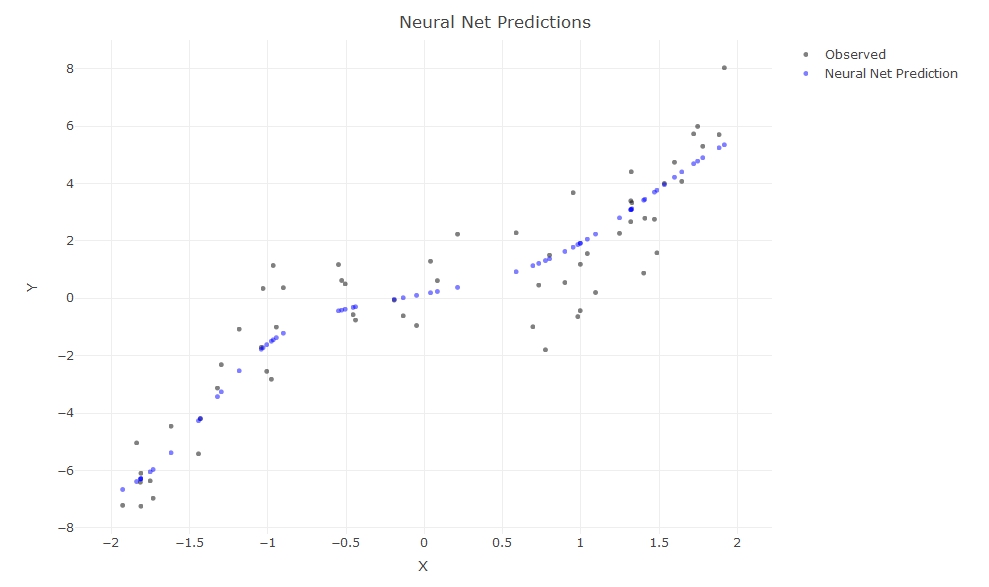
**Part (e + f)**

The gradient is used with a learning rate of 0.01 and 200 iterations. After graphing using the weights that have been optimized with 200 iterations, I can see that the weights have not fully converged to the best values based on the graph but it can still be seen that neural net is able to predict the curve.

Refer to code part e) and f)

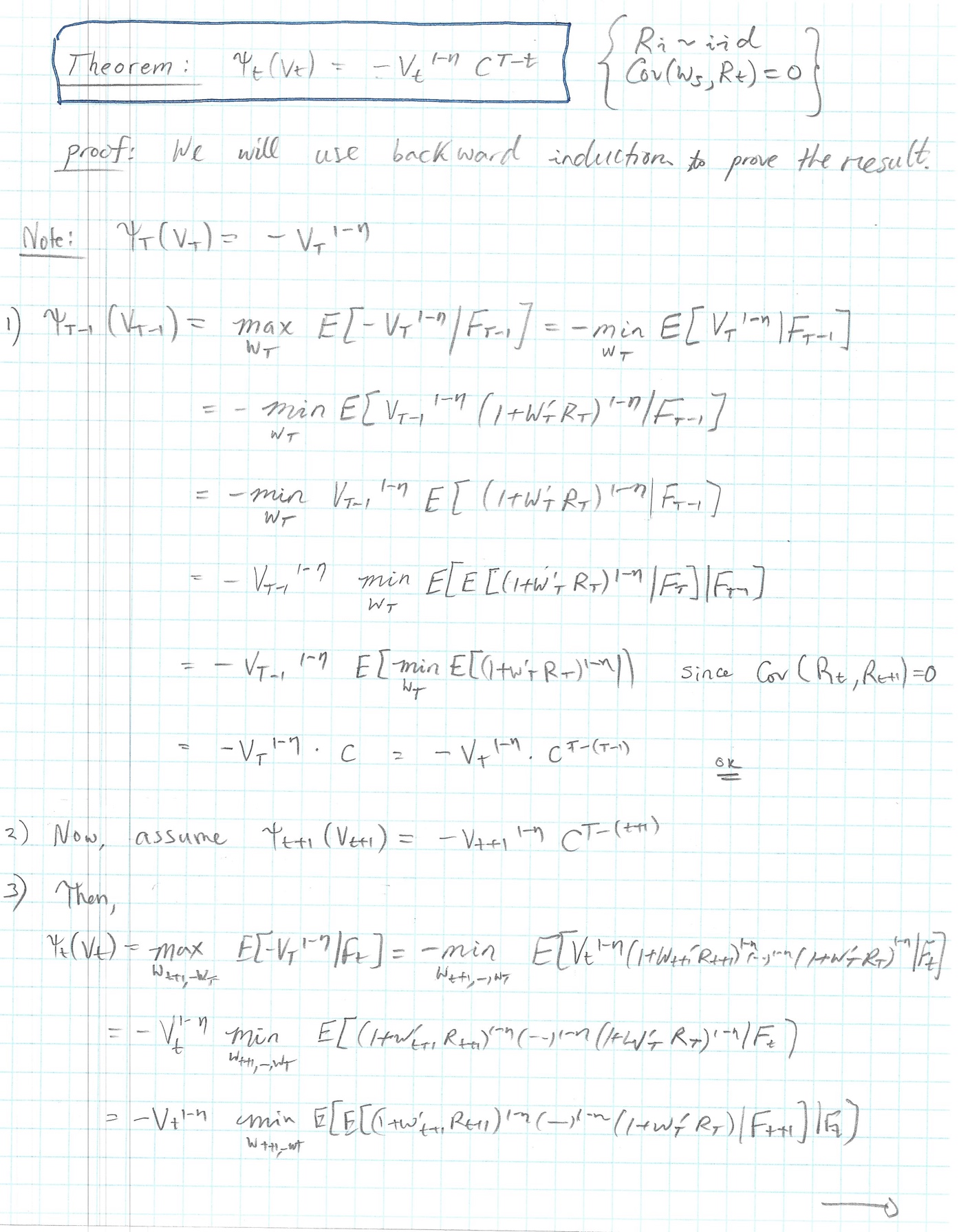


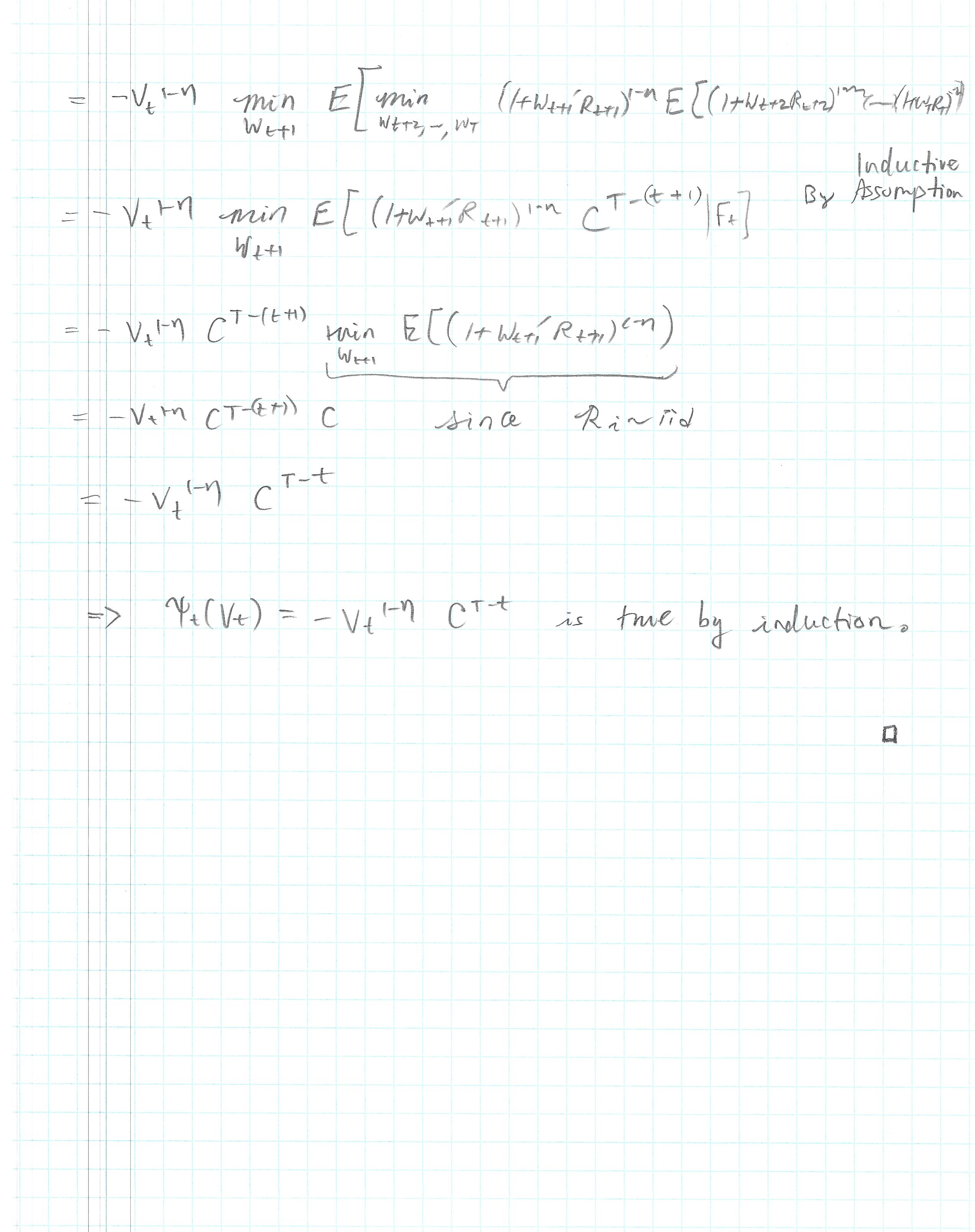
We can see that the first neural net does start to show that it has mimicked the original curve, however once the iterations have been increased more we can see that the neural net is closer to the original x^3 curve and closer to the original data.

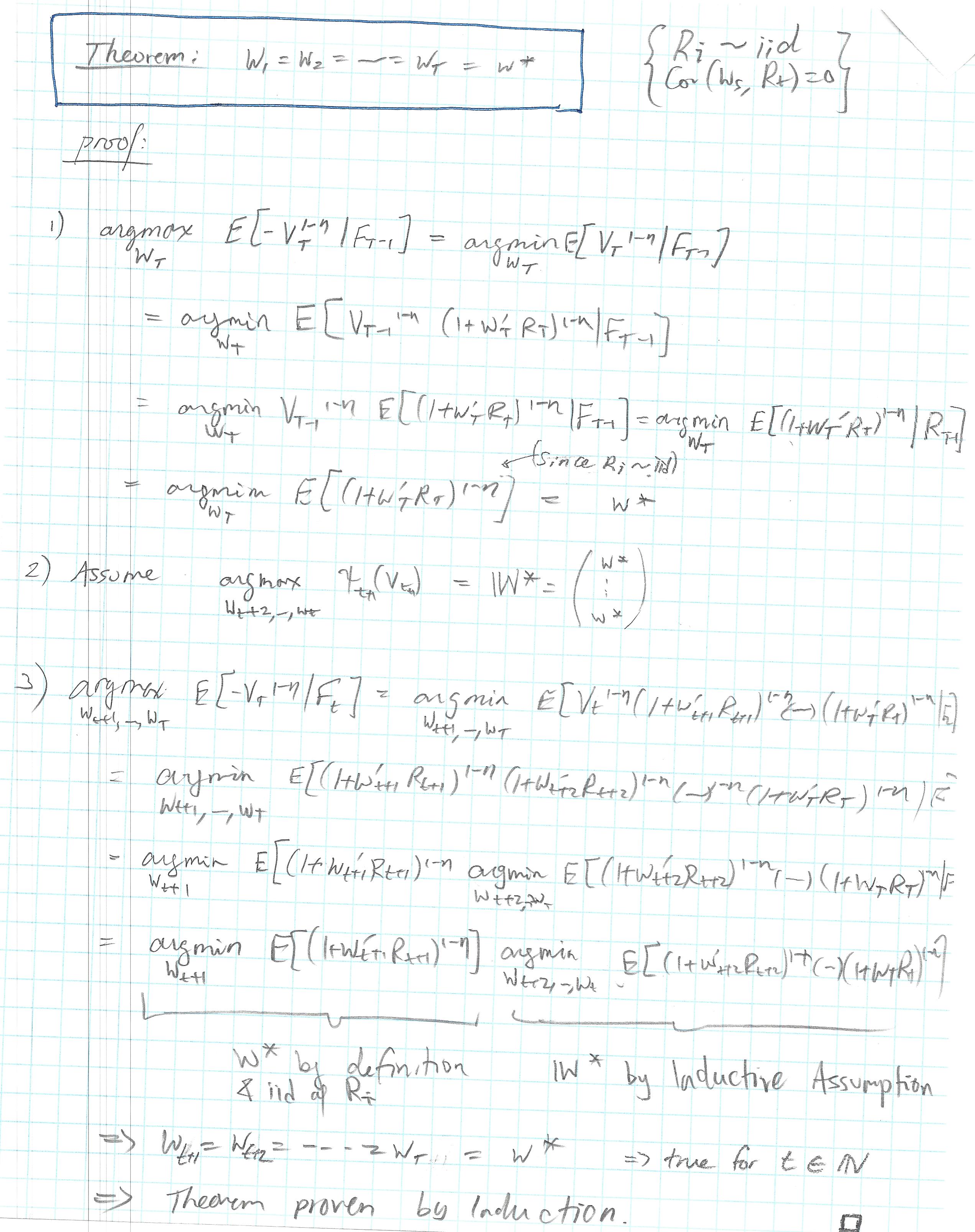


QUESTION 3

**Part (a)**







**Part (b)**

* Estimates of mean annual log-returns:

|  |  |
| --- | --- |
| SP500 | 7.303E-02 |
| WFBIX | 4.850E-02 |

* Estimate of covariance matrix of annual log-returns:

|  |  |  |
| --- | --- | --- |
|  | SP500 | WFBIX |
| SP500 | 2.097E-02 | -1.698E-05 |
| WFBIX | -1.698E-05 | 1.597E-03 |

* Correlation between annual log-returns of SP500 and WFBIX:

***ρ* = -2.933E-03**

* Standard deviation of annual log-returns

|  |  |
| --- | --- |
| SP500 | 1.448E-01 |
| WFBIX | 3.998E-01 |

**Part (c)**

* When we increase η, the optimal portfolio weights approach (0.5, 0.5).
* η can be seen as the level of risk aversion. As η increases, the investor is more risk averse meaning he or she dislikes uncertainty and prefers having smaller returns with lower variance than having larger returns on average but with higher variance. From the graph we see that when η is small, the investor is willing to favor one asset over the other in order to maximize utility. However, as η increases, the investor does not want to ‘bet’ on which asset will perform better, so he or she invests equally in both assets.

