Eigenvectors and eigenvalues computation using Neural Networks

Raghvendra Mishra (ME21B1075) and Ayush Agarwal (ME21B1076)

15th May, 2023

Computational Methods

- Neural Network Algorithm

The algorithm is defined as

$$\frac{dx(t)}{dt} = -x(t) + f(x(t))$$

$$f(x) = [x^T x A + (1 - x^T A x)I]x$$

Which converges through iteration of $x(t)$ with respect to time, starting with a $x(0)$.

Computational Methods

- Neural Network Algorithm
- Power Method

The algorithm is defined as

$$y_k = Ax_k$$

$$x_{k+1} = \frac{y_k}{\max(|y_k[1]| + |y_k[2]| + \cdots + |y_k[n]|)}$$

Which converges through iteration of $x(t)$ with respect to time, starting with a $x(0)$.

Computational Methods

- Neural Network Algorithm
- Power Method
- Jacobi Algorithm

The algorithm is defined as

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2a_{ij}}{a_{ii} - ajj}\right)$$

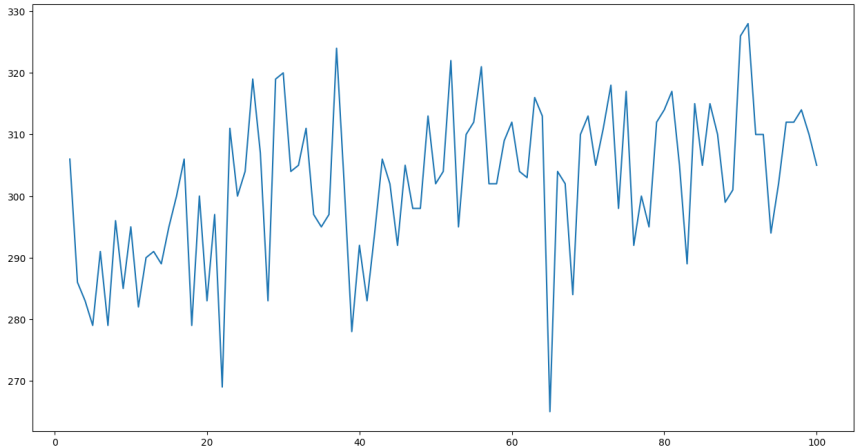
$$R[i][i] = R[j][j] = \cos\theta \quad R[i][j] = -\sin\theta \quad R[j][i] = \sin\theta$$

$$B = R^T AR$$

And the steps are repeated until B is obtained to be a diagonal matrix.

Neural Network Algorithm

Dimensions vs Iterations

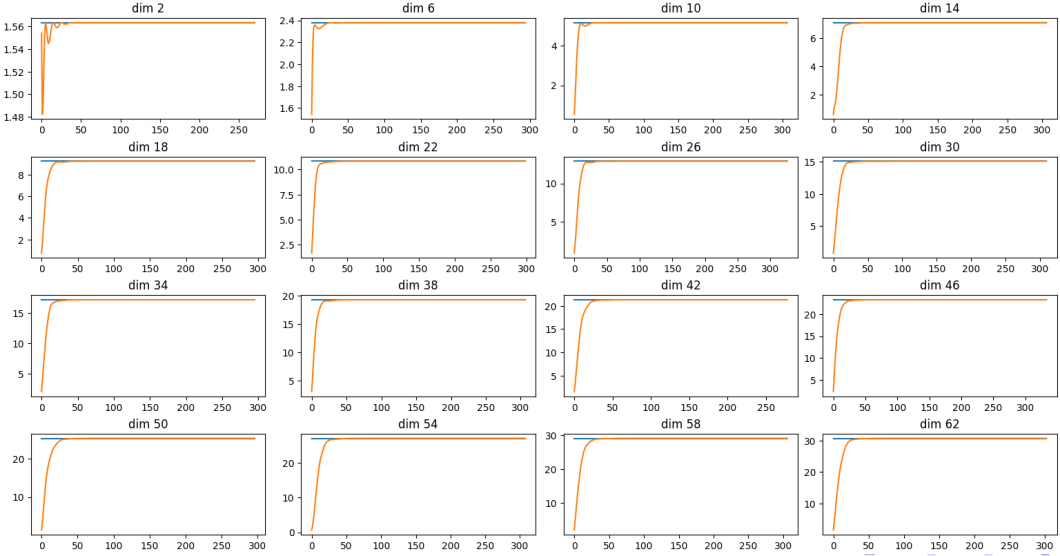


Neural network algorithm is pretty efficient in learning to produce the eigenvectors and eigenvalues even for 100×100 matrices way below the set limit of iterations which is 5000 for each of the methods. These are the values for the number of steps ranging from 2×2 to 100×100

$$\begin{bmatrix} 306, 286, 283, 279, 291, 279, 296, 285, 295, 282, 290, 291, 289, 295, 300, 306, 279 \\ 300, 283, 297, 269, 311, 300, 304, 319, 307, 283, 319, 320, 304, 305, 311, 297, 295, \\ 297, 324, 302, 278, 292, 283, 294, 306, 302, 292, 305, 298, 298, 313, 302, 304, 322, \\ 295, 310, 312, 321, 302, 302, 309, 312, 304, 303, 316, 313, 265, 304, 302, 284, 310, \\ 313, 305, 311, 318, 298, 317, 292, 300, 295, 312, 314, 317, 305, 289, 315, 305, 315, \\ 310, 299, 301, 326, 328, 310, 310, 294, 302, 312, 312, 314, 310, 305 \end{bmatrix}$$

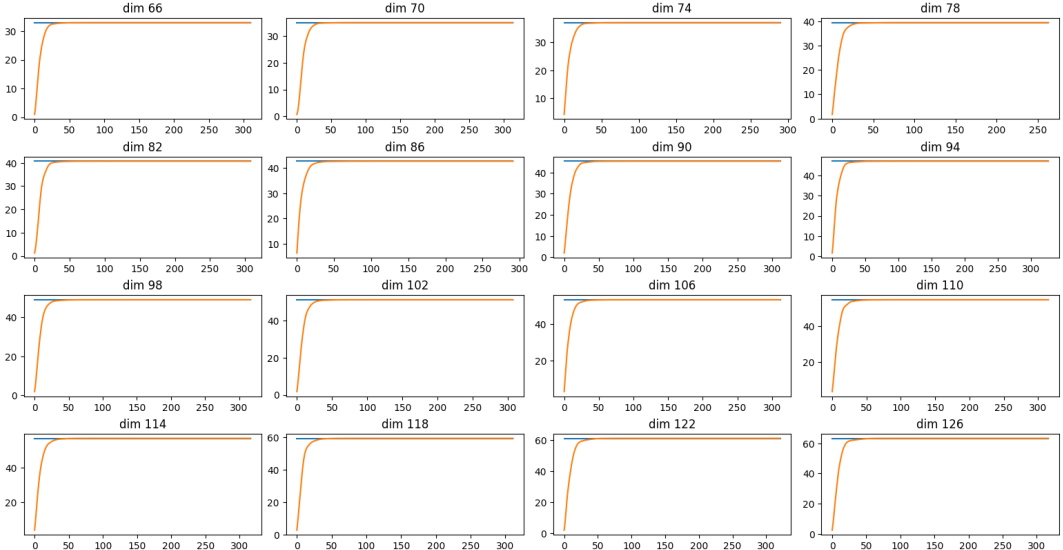
Neural Network Algorithm

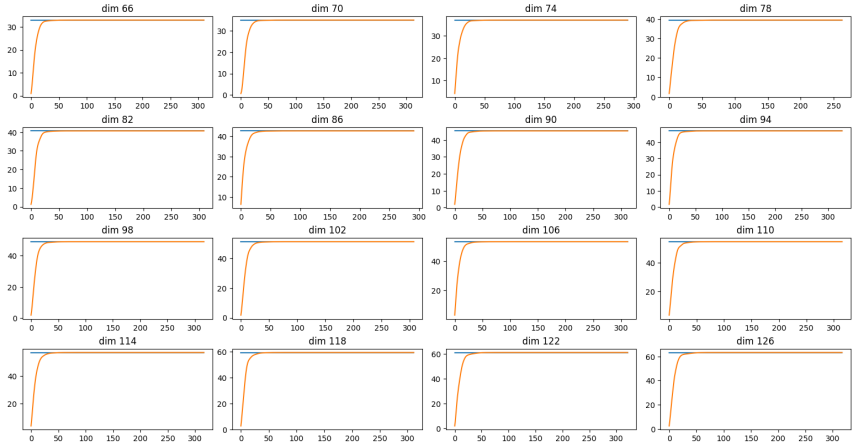
Error Analysis and Convergence



Neural Network Algorithm

Error Analysis and Convergence

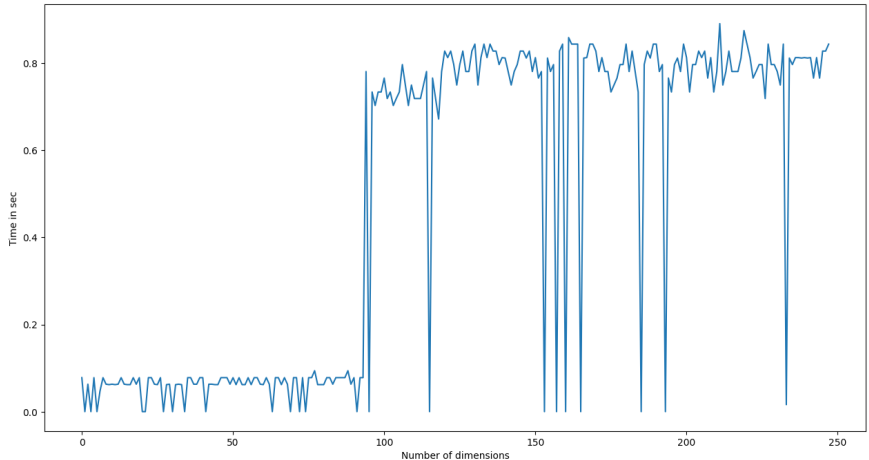


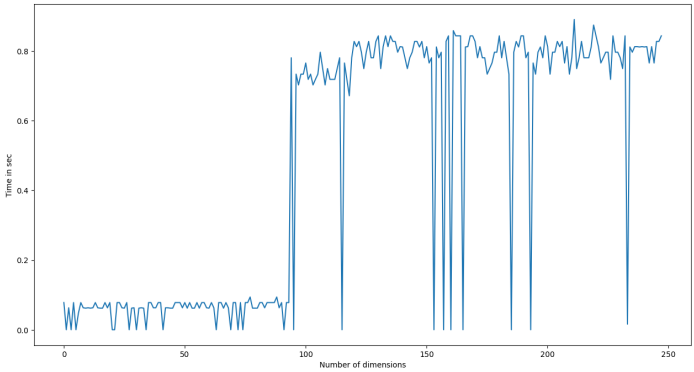


The above give the error analysis of calculating the eigenvalues using a neural network for dimensions ranging from a 2×2 matrix to a 126×126 matrix.

Power Method

Dimensions vs Time

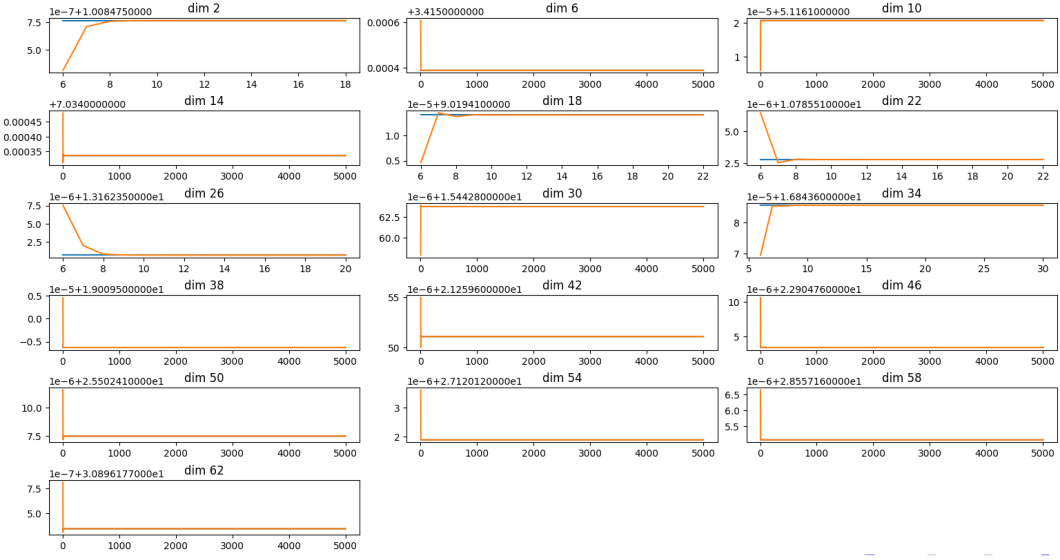




From the above graph we can see Power Method requiring less than a second to get the eigenvalues of a given matrix, even for a matrix of 250×250 size.

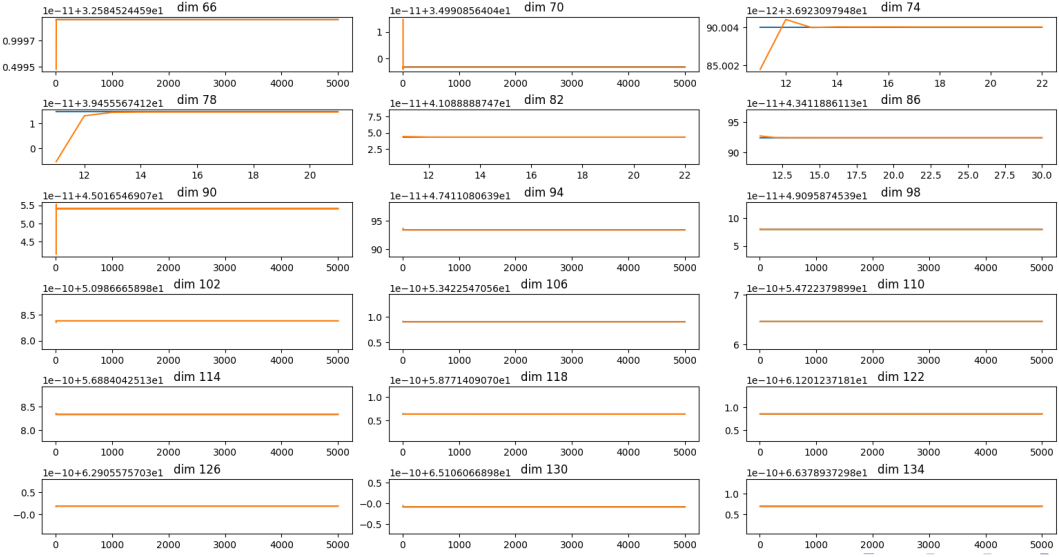
Power Method

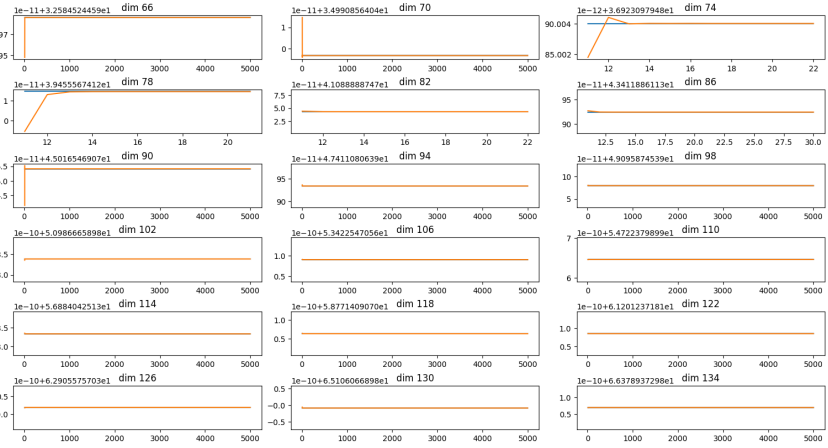
Error Analysis and Convergence



Power Method

Error Analysis and Convergence

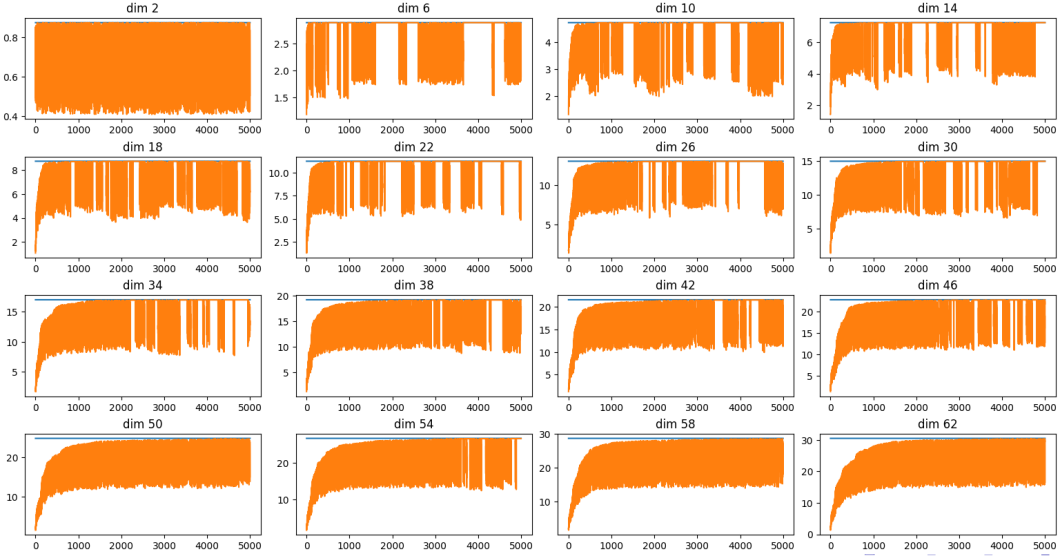


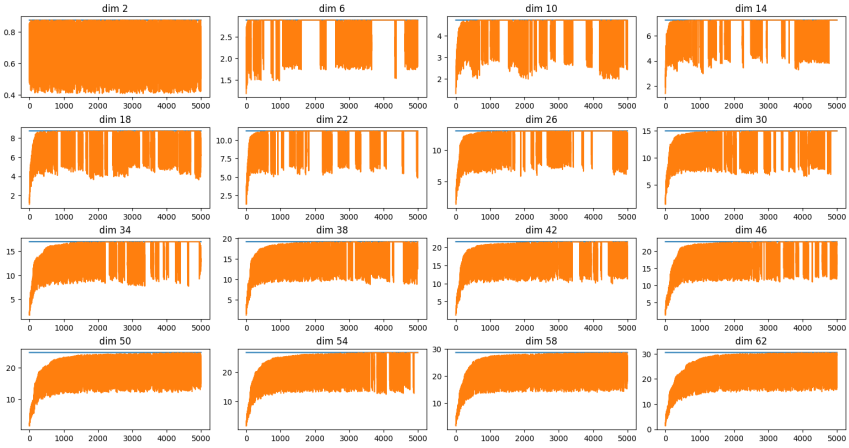


The above give the error analysis of calculating the eigenvalues for the power method algorithm for dimensions ranging from a 2×2 matrix to a 134×134 matrix. Power Method almost instantly achieves convergence but it never touches the threshold value of 1e-16.

Jacobi Algorithm

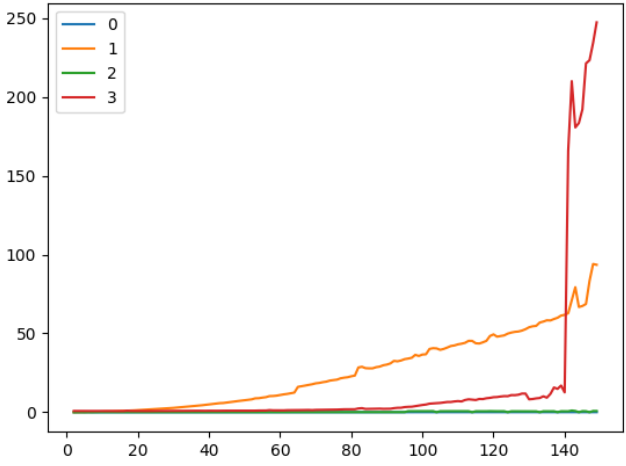
Error Analysis and Convergence





As you can see, the convergence for Jacobi algorithm is really erratic and varies quite much from the blue line that is the true eigenvalue and doesn't really converge onto it.

Time vs dimensions



This is a time vs number of dimensions of symmetric matrix A where time is in seconds on the y-axis and the dimensions are on the x-axis.

0 - Numpy values
1 - Jacobi algorithm
2 - Power Method
3 - Neural Network

As the graph suggests, Jacobi is the slowest of the 3 methods until the matrix size exceeds 140 where Neural net algorithm starts to fail and go way beyond the 200 seconds mark. In terms of time, **Power Method** is the most efficient.

Conclusions

The conclusions can be drawn that Neural Network is an accurate and efficient way of calculating eigenvalues and eigenvectors only upto a certain dimension of less than 150. It may require some changes, like changing the size of the neural network for bigger matrices, to make the convergence faster but for the scope of this paper, the time exceeds a minute as the number of dimensions reach 100.

Conclusions

The conclusions can be drawn that Neural Network is an accurate and efficient way of calculating eigenvalues and eigenvectors only upto a certain dimension of less than 150. It may require some changes, like changing the size of the neural network for bigger matrices, to make the convergence faster but for the scope of this paper, the time exceeds a minute as the number of dimensions reach 100.

But, neural network converges to the desired accuracy faster than any other method in under 500 steps for matrices of even 150 dimensions.

Conclusions

The conclusions can be drawn that Neural Network is an accurate and efficient way of calculating eigenvalues and eigenvectors only upto a certain dimension of less than 150. It may require some changes, like changing the size of the neural network for bigger matrices, to make the convergence faster but for the scope of this paper, the time exceeds a minute as the number of dimensions reach 100.

But, neural network converges to the desired accuracy faster than any other method in under 500 steps for matrices of even 150 dimensions.

So, for faster computations we may use Power Method, with the computational speed of less than a second, but for higher accuraies, we require the power of a neural network to help us.

Conclusions

The conclusions can be drawn that Neural Network is an accurate and efficient way of calculating eigenvalues and eigenvectors only upto a certain dimension of less than 150. It may require some changes, like changing the size of the neural network for bigger matrices, to make the convergence faster but for the scope of this paper, the time exceeds a minute as the number of dimensions reach 100.

But, neural network converges to the desired accuracy faster than any other method in under 500 steps for matrices of even 150 dimensions.

So, for faster computations we may use Power Method, with the computational speed of less than a second, but for higher accuraies, we require the power of a neural network to help us.

Meanwhile, Jacobi is just bad.

Thank You!