Latent space trajectories of biological and artificial neural-networks

Ash Robbins

December 10, 2020

Abstract

Untangling black boxes can shed insight into highly complex systems. One class of methods is dimensionality reduction, which takes high dimensional data, and transforms it into an encoded representation. How this encoding works has potential to relate to underlying patterns of the data. In this project, I use Tensor Component Analysis (TCA) which can reduce Nth order tensors into factors representing each axis. TCA is applied to both biological and artificial neural networks, shown to provide insights on both sets. Visualizing this complex data becomes, but advanced visualization techniques help uncover meaning.

1 Background

Analyzing complex neural firing patterns is an important issue which can shed light on human foundations of information processing and learning. As far as we know, human brains are the most complicated system in the known universe, and the pursuit of understanding them may help inform human-designed artificial intelligence.

In order to help understand the high-dimensional data, dimensionality reduction techniques such as PCA and ICA have been used to create a lower dimensional space. This "latent space" essentially can show the data in a way where structurally similar data points are nearby eachother. For biological neural-networks, we run into a problem where many methods only explore the latent dynamics linearly. Papers such as [1] explore nonlinear , and plot the dynamics a latent space. This shows how real neural data recorded from a Macaque in an arm reaching task can be separated based on positions, so the similarity between the tasks is captured in the latent space.

Other papers have explored dimensionality reduction techniques for neural recordings, and [2] provides a large review on different techniques. Even black box encoders such as Hierarchical Convolutional Neural Networks(HCNN) have shown biological relevance in predicting visual cortex activation from the higher level hayers in the HCNN [3].

2 Methods

2.1 Notation

We will generalize notation similar to that in [4]. Our data matrix $\mathbf{X}$ represents traces of neural activity through time which is size NxT, where N indicates the number of neurons and T indicates the total timesteps. For tensors, we only utilize 3rd order tensors in this work which are denoted by $\mathcal{X}$, indicating a size $NxTxK$ tensor, where K may represent different experiments, tasks, etc.

2.2 Principal Component Analysis

Starting with the matrix decomposition case, we assume to have a matrix $\mathbf{X}$ where each row vector x_i' for $i = 1, 2, ..T$ represents a single neurons activity trace through time. Principal component analysis (PCA) finds a reduced dimensionality representation of size R where the data $\mathbf{X}$ can be a approximated as $\hat{\mathbf{X}}$. In this case

$$\hat{\mathbf{X}} = \mathbf{W}\mathbf{B}^T$$

where $\mathbf{W}$ is of size NxR and $\mathbf{B}$ is size TxR.

PCA minimizes the error $\mathbf{X} - \mathbf{W}\mathbf{B}^T$ via the squared error, or Frobenius norm. However under PCA, this algorithm is minimized with the constraint that the vectors of both $\mathbf{W}$ and $\mathbf{B}$ are orthogonal.

2.3 Tensor Component Analysis

Tensor Component Analysis (TCA) can intuitively seem like an extension of PCA into n-th order tensors. The problem attempts to find a decomposition which decomposes data, such as the 3rd order tensor χ which is of size $NxKxT$ where N is neurons, K is experiments/trials, and T is time. The decomposition which minimizes:

$$||\chi - \hat{\chi}||_F^2$$

is desired. $\hat{\chi}$ is recreated through each component of it being determined with:

$$x_{ntk} \approx \sum_{r=1}^{R} w_n^r b_t^r a_k^r$$

where w^1 is the first neural factor, b^1 is the first time factor, and a^1 is the first experimental factor. Alternating least squares is used to minimize the reconstruction error.

3 PCA Toy Experiment

To start exploring results of observing these dimensionality reduction techniques, we employ them on both toy examples and real examples. We start with initial cases of techniques to elucidate what the latent variables (neuron

factors or temporal factors) can represent. Beginning in the easier cases and adding complexity can help ensure that we are visualizing and analyzing the data correctly.

3.0.1 Neural Factor Evaluation

Let us assume we have activity of 5 neurons. In order to pull out understandable variables in an unsupervised manner, we implement PCA via SciPy. The first toy example generates 5 neurons with traces which look like:

$$x = \begin{bmatrix} -2cos(t) \\ -cos(t) \\ .5cos(t) \\ cos(t) \\ 2cos(t) \end{bmatrix}$$

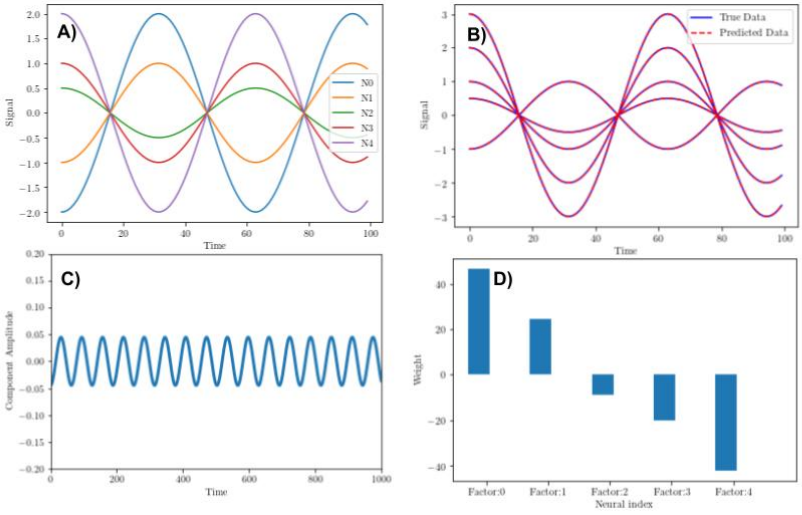

Figure 1: A) Traces of 5 simple signals. B) PCA Recreation of the traces. C) First temporal latent variable, accounting for 100% variance. D) Neural factors for the first latent variable, effectively scaling the amount of the neural factor.

Looking at Figure 1 we see that there is only one latent factor as composed, even though the analysis looked for three components. The other factors are visualized, but the line width is scaled by how much the factors account for the total variance. In this, the first variable accounts for 100% of the variance, thus there is one temporal factor. The neural factors as constructed account for the scaling of each variable.

3.0.2 Neural and Temporal Factor Evaluation

This time we take the case of having 5 neurons composed in the following manner:

$$x = \begin{bmatrix} -2sin(t) \\ -cos(t) + sin(t) \\ .5cos(t) + 2sin(t) \\ cos(t) - sin(t) \\ sin(t) \end{bmatrix}$$

Taking note that this time, there are two functions, where everything is a linear combination of both *sin* and *cos* functions. The prediction is that the PCA algorithm will fundamentally be able to pull out two temporal factors that can explain the data completely.

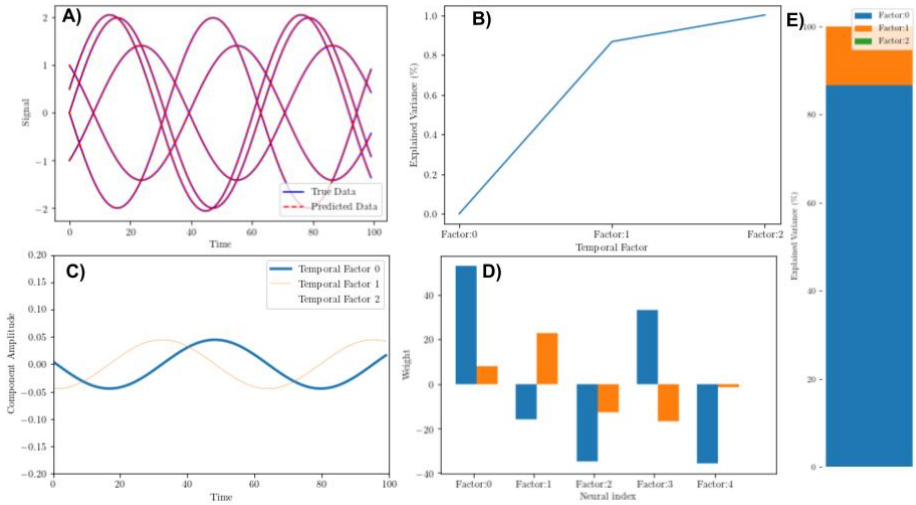

Figure 2: A) Traces and recreation of 5 compound signals. B) Variance explained by the principal components C) Temporal components with line width indicating the variance explained D) Neural factors for the latent variables, effectively scaling the amount of the neural factor. E) Visually displaying the percentage of variance explained by components.

Figure 2 shows the results from this experiment. We see that PCA can recreate the data sufficiently, and that two main temporal factors are determined. Figure 2C shows the temporal factors with their line width identifying the percent of variance accounted for(also present in 2B and E). Figure 2D shows how each temporal component is weighted for the corresponding neurons, which shows relative similarities to the equations crafted.

3.0.3 Trajectory in Reduced Dimension

In order to see intuitively see how trajectories can be reduced into a lower-dimensional space, we begin with a toy problem which has a follows a 3D path. The path is defined as:

$$x = \begin{bmatrix} 10 + .04t^2 + cos(22t) \\ t^2 + cos(5t) + 5 \\ -.1t^2 - cos(10t) \end{bmatrix}$$

The dimension of x is 3 initially, but it can be reduced to 2 dimensions using PCA and selecting the 2 components which account for the most variance, seen in Figure 3.

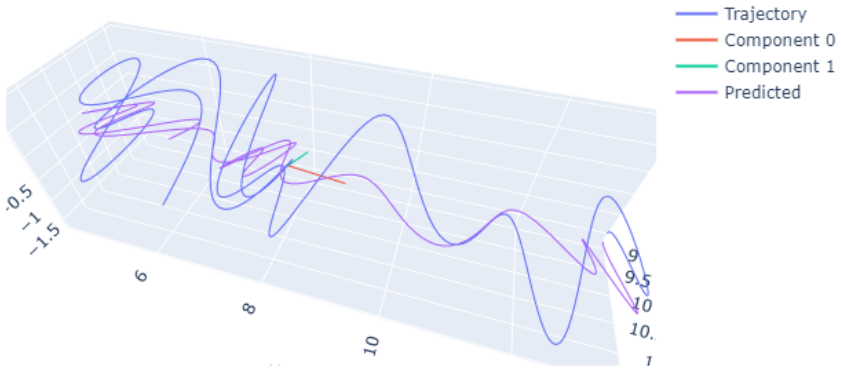

Figure 3: Trajectory and projection of the toy problem, along with the major components showing their corresponding direction

4 Results

4.1 Biological Neural Data

Taking the same TCA approach, data is taken this time from true biological samples. Primary neurons are dissociated and placed on a 6 well plate, adhered to a microelectrode array (MEA). This MEA has 64 channels for recording for

each well. The experiment conducted involved stimulating the neurons every 20 seconds for 13 total minutes. The data processing pipeline is seen if Figure 4.

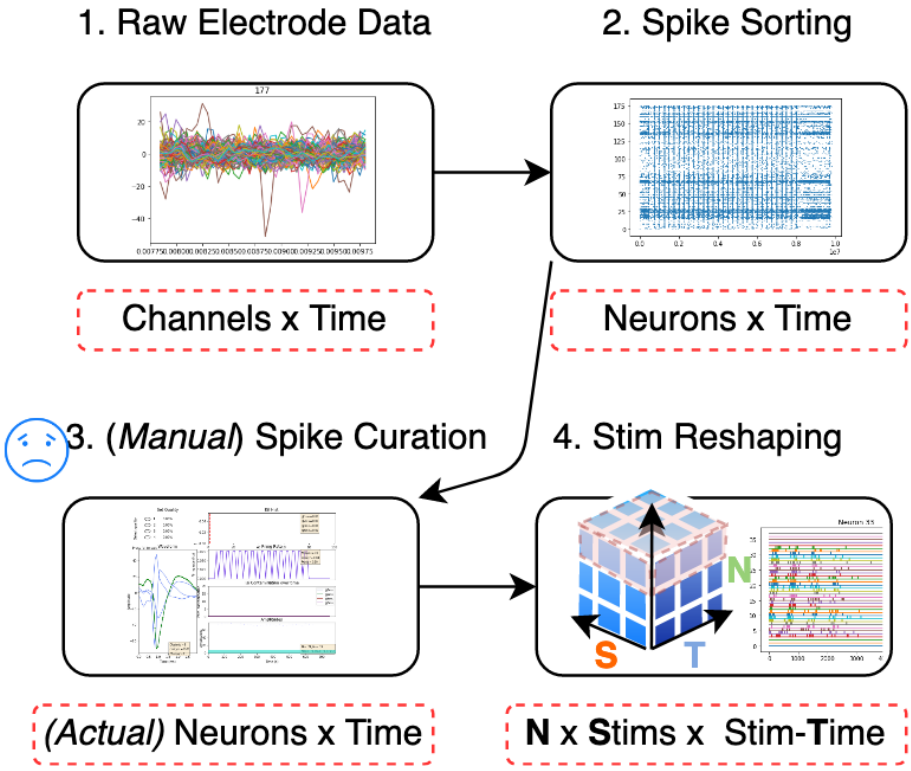

Figure 4: Data flow from raw electrode to a 3rd order tensor (NxKxT).

Beginning with the raw data, the neural spikes are sorted and turned into singular binarary events. Spikes are manually curated to be deemed correct, then the data is wrapped around the periodicity of the stimulations. This gives the 3rd axis of the tensor, resulting in an NxKxT tensor. The factors identified with TCA can be analyzed to show patterns of which neurons show specific temporal patterns, and how those patterns change through time. In this case, it would show how neurons show different activity types as they are stimulated more and more. In the long term, experiments may show underlying patterns which represent learning, such as two neurons having a pattern of activity which strengthens through experiments. The visualization reducing to 3 factors is shown in Figure 5.

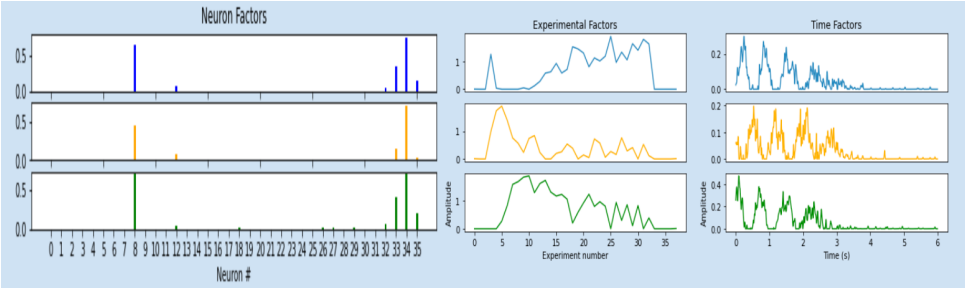

Figure 5: Factors for the primary cortical neurons. Rows indicate factor number, columns indicate factor type.

Observing the neural factor shows us which neurons are active together, and can tell about their underlying connectivity. The experimental factors in the middle tell a story which may align with two neural properties of short term potentiation and short term depression. We would see these occur as the activity increasing or decreasing respectively with each stimulation. The neuron factors do show this, with the first one exhibiting increasing strength through trials, and the second two losing strength through more stimulations. The activity itself is seen in the time factor, which shows that there are dampened oscillations occurring which seem to be at differing phases. A fourier transform is visualized in Fig. 6

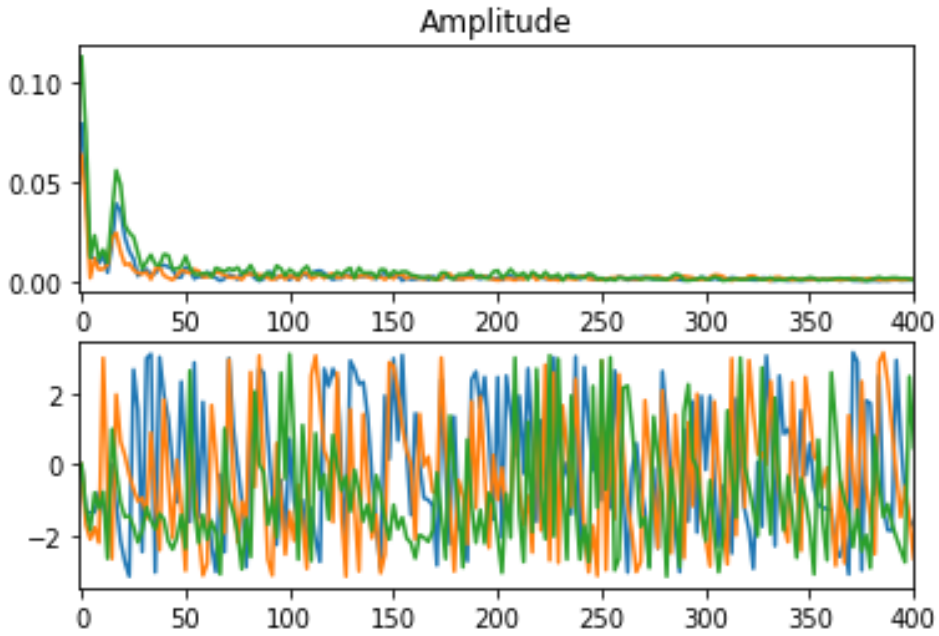

Figure 6: A fast fourier transformed is used on the neural factors, showing that there is a peak in the low frequency range around 12 Hz. Phase angles around that peak are differing, while the rest seems to be noisy.

Finally, the neural factor is used to reduce the problem to 3 dimensions, and a trajectory is plotted through time. This is shown through the gif file Supplementary Figure 1.

4.2 Artificial Neural Network

A final case study was completed to analyze more interpret able trajectories. 30 machine learning models are trained on the MNIST dataset which consists of 60000 handwritten digits. The model involved has one fully connected layer, so each input pixel (size 784) connects to each possible output digit class (10 digits). Together this results in a tensor of shape (MxDxNxT) or (models x digits x neural weights x timesteps). Once again we identify factors for each component, shown in Figure 7

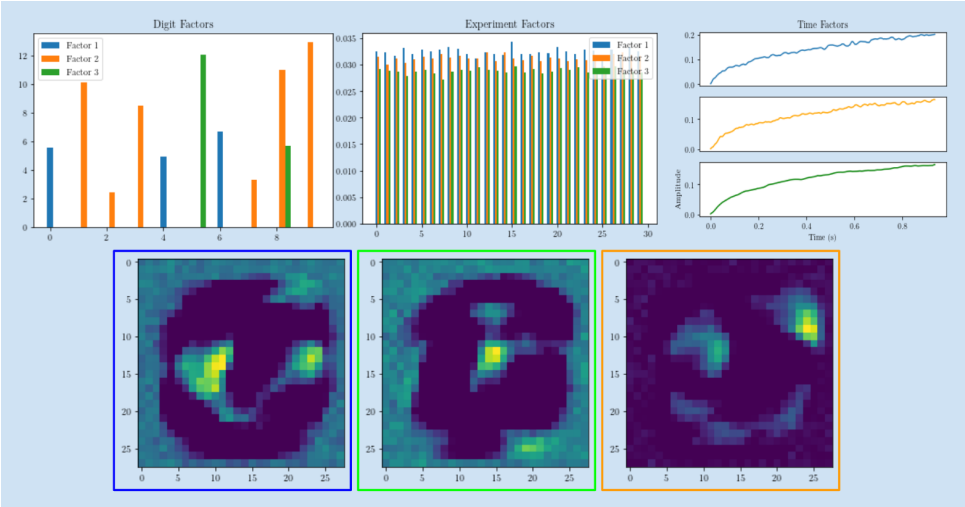

Figure 7: Three factors for each axis are displayed. Digit factors represent how much each digit displays the corresponding other factors (blue factors exhibit the qualities of the other marked blue factors, or factor 1). The experimental or model factor displays that each model has a similar weighting. The temporal model strictly display that the most pertinent time change are weights increasing. Finally, the weight factors can be reshaped to visibly show a representation. The blue factors from the digit factors mostly show the corresponding blue weight representation.

These factors can provide great insights on what is happening under the hood. Finally, we can see how their trajectories move in the reduced dimensional space. Since 3 factors have been determined, the weight space is reduced from 784 to 3, and are visualized through time. Each digits weight is color coded, and we can successfully see that the digits share separate trajectories through this low dimensional space. Figure 8 and Supplemental Figure 2 show the trajectory.

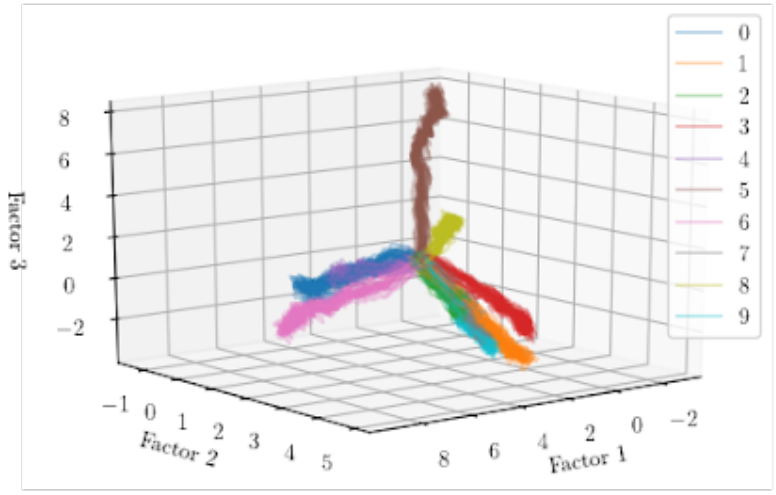

Figure 8: Trajectories of ANN models are graphed through the reduced dimensional weight-space, with digits color coded. We see definite separation for differing digits.

5 Conclusion

Dimensionality reduction techniques such as Tensor Component Analysis can help provide insights into black boxes which have inputs and outputs. Large amounts of data can be used to find smaller dimensional factors which can recreate the full data. These factors can have meaning in how underlying computation is done. Through analysis of biological data, different activity increases or decreases through multiple stimulations, and primary activity patterns are identified. In the case of an artificial neural networks, we can see easily discernible trajectories in a low dimensional space corresponding to how the algorithm was trained. Visualization techniques are absolutely necessary in order to find interpretations of this complex data.

References

[1] Y. Gao, E. Archer, L. Paninski, and J. P. Cunningham, "Linear dynamical neural population models through nonlinear embeddings," *arXiv:1605.08454 [q-bio, stat]*, Oct. 2016. arXiv: 1605.08454.

[2] J. P. Cunningham and B. M. Yu, "Dimensionality reduction for large-scale neural recordings," *Nature Neuroscience*, vol. 17, pp. 1500–1509, Nov. 2014. Number: 11 Publisher: Nature Publishing Group.

[3] D. L. K. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature Neuroscience*, vol. 19, pp. 356–365, Mar. 2016. Number: 3 Publisher: Nature Publishing Group.

[4] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli, "Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis," *Neuron*, vol. 98, pp. 1099–1115.e8, June 2018.