# Research Progress Report

Zachary McNulty

July 2019

## 1 Initial Questions

When I first started this project, the goal was to apply some of the same techniques Stefano applied in his *paper* to natural scenes. Stefano showed that these RNNs, when trained in a predictive setting, seemed to be able to extract some of the latent features associated to the task at hand. By applying this to a natural scene, the hope is that we could find some underlying structure that otherwise was not apparent. Some of the relevant questions would be:

1. How can we expand Stefano's techniques to the processing of images?
2. If they exist, what are these relevant low dimensional structures in natural scenes?
3. Do the dynamics of the agent's movement throughout the scene or the task the agent is trained to perform influence the representations built? Are the representations mainly task-dependent or context-dependent?
4. What facets of predictive learning are advantageous to building/storing these representations?

The project started as a simple extension of Stefano's paper. We would have an "eye" perform a series of saccades over a fixed image, moving across the image in a stochastic manner. At each point in time, the "eye" would sample some of the pixels at its current location. Using this information and the direction of movement for the next saccade, it would be trained to predict what it saw in the future. I was hesitant to try this project as I did not think it would provide any new information. At each point in time, it seemed the best way for the network to predict the future would be again to just build a representation of the 2D space the image lives in. Then we would just get identical results to Stefano's paper, with most of the neurons acting as place cells. We could try not giving the agent any of the positional information (i.e. not tell it the direction of the next saccade and just put a restriction on how far the saccade could move), but I was worried this would be too complicated for a simple network to learn, especially with detailed images. With relatively continuous motion, the latter case just becomes a case of video processing, so I thought I might as well try prediction on videos.

Initially, I was worried natural scenes are simply too complicated to train in. I feared any network capable of successfully performing the predictive task would be far too complicated to open up and look for low-dimensional representations. Furthermore, even if we tried it would not be clear what kind of low-dimensional representations we expect to see and this would make it difficult to analyze these representations in any meaningful way.

## 2 Simplification

Due to the above concerns, I wanted to first study these questions in a simpler setting. This would hopefully allow me to build a network architecture that could handle a sequence of images, but that would still be simple enough to analyze. I wanted to choose an image set where the underlying structure was more or less obvious, and to look for representations of this structure in the network. I chose to make my videos revolve around dynamical systems as these can often be parameterized by only a couple of variables, and thus should be conductive to some kind of low-dimensional representation. Of course, I could just feed in the relevant parameters of the dynamical system to the network (e.g. the position and velocity), but this would not provide any information about how I could transfer this information to video processing. Instead, I chose to create a simple video of a simple object undergoing the desired motion.

## 3 Designing A Network

We wanted to try to find a balance between the simplicity of the network and the network's generalizability to more complicated tasks. The videos had frames that were $61 \times 61$ pixels and often contained a lot of redundant/invariant information (i.e the white background, the shape/color of the object). If we wanted to apply just a simple recurrent network to this, it would have to be so massive its inner workings would be beyond understanding. I tried, and the network was not able to learn the task. Instead, we chose a paired autoencoding and recurrent structure. The autoencoder would be trained to capture the invariants in the image and encoder the state of the system in a low-dimensional representation. From there, the recurrent network would be trained to learn the dynamics in this low-dimensional representation built by the autoencoder. This allowed us to keep the recurrent layer relatively small, only 64 neurons.

## 4 Problems

When we first started, I just had the object oscillate through the center of the screen along some angle. After some effort, I was able to get the network to successfully complete the task with a simple recurrent layer (flattening the autoencoder's representation in the middle). But when I analyzed the network, I saw no clear tuning of the network towards the parameters of the dynamical system. Furthermore, the recurrent network seemed to have a very binary response and even the autoencoder showed some tuning to the angle of oscillation (I wrote about this in a report for one of my classes, see Figure 8). To help fix this, we initialized the recurrent weights to the identity to help promote high autocorrelation within the recurrent layer. We also decided to change the videos so that the object could oscillate anywhere on the screen, not just the center. It was hoped this would help separate the roles of the autoencoder and the recurrent layer, removing the autoencoder's apparent tuning to the angle of oscillation. However, upon doing this the network consistently fails to predict the future state of the system, regardless of how much training data we give it. I thought this may just be because the recurrent layer had to flatten the output of the autoencoder and this loses too much of the local information stored in the autoencoder, but even when I swapped out the simple RNN for a convolutional RNN, the network still fails.

## 5 Other Papers

While there is quite a lot of work on predictive learning on videos, none of the papers I have read so far have accomplished the task with a simple recurrent network. This paper by Shi used three convolutional LSTM layers, but could only obtain mediocre results on the moving MNIST problem, a far simpler problem than prediction in natural movies. This paper by Cox used PredNet, a deep neural network using both convolutional, and convolutional LSTM layers, did a pretty good job at video prediction on a natural movie, but the design is very complicated and requires many recurrent layers. In this paper (Figures 4 and 5), Cox also showed the network representation stored some information on a relevant latent variable (steering angle of a car in the video): a linear model could be built to fairly accurately predict the steering angle from the network representation at each frame in the video. This gave us hope that these networks store some kind of representation of these latent variables.

## 6 Conclusions

I think it is unlikely we would be able to discover latent structures in natural scenes using this technique as it is. While these recurrent neural networks clearly are capable of extracting these latent variables, it will be hard for us to determine specifically what the variables are. In both Stefano's and Cox's papers, the underlying latent variables are known beforehand. This allows them to do something along the lines of supervised learning: either breaking apart the network neuron by neuron and checking for tuning against the latent variables, or building a separate model to predict the latent variables from the neural representations. What I am currently trying to do is a lot more like unsupervised learning. These techniques like PCA and clustering will look mostly at the global representation of the network rather than the activities of single neurons, so they won't be as effected if we choose to complicate the network.

## 7 What Next?

I think we should move away from trying to analyze this network from a supervised setting. While these are simple and powerful techniques, I do not think they will scale well to our future goal involving complex natural scenes and feature extraction. In the future, we will inevitably have to make the network more complicated and these neuron by neuron analysis techniques will no longer suffice. Furthermore, it is our desire to use these techniques that is restricting the complexity of the network and this simplicity is starting to cause the network to fail to complete its task. To put it shortly, I think we should make the network more complicated to begin with. PredNet, the network from Cox's paper, is available on GitHub and implemented as a custom Keras network layer. This should make it fairly easy to use, and it has already been proven to work on complicated videos. PCA will still be a powerful tool we can use, and we can use various unsupervised clustering techniques to group different stimuli (videos) and compare them for similarity. Some downsides of this would be that it would make the research beyond the realm of mathematical analysis: we could no longer make any of the kinds of claims along the lines of the math found in Stefano's paper. And of course as we make the network more difficult, its behavior gets more difficult to visualize. Furthermore, it seems the authors of PredNet have not done anything with it since the released the

paper two years ago, even though they had similar goals to my current line of research. There was also a paper released recently (albeit I question the paper's thoroughness) that criticizes PredNet, claiming it is not truly following the predictive coding philosophy. I am a bit concerned the paper may be right. The PredNet paper provided little evidence its recurrent layers were encoding latent variables in a meaningful way (Figure 3 of the paper seems to suggest the trained representations could only predict the latent variables slightly better than the representations under random initial weights). Still, I think PredNet might be worthwhile to explore.

# 8 End Results

I did not find any more success after starting to use the PredNet architecture. While there seemed to be some clustering of the latent variables within the principal components of the neural representation, this was far from conclusive. Furthermore, the PredNet architecture was somewhat cumbersome for doing multistep predictions.

I tried leveraging this clustering to discover the latent variables. The main idea was to collect a sample of videos which had similar neural representations (within the first three principal components) and compare the videos for similarities to extract features relevant to prediction within that particularly set of scenes. However, I could find no good video comparison algorithms that allowed for the simple extraction of features. If this project is to go anywhere, I think it should simply aim to expand on Stefano's findings and show that convolutional layers do not interfere with the RNNs ability to extract low dimension structure within the predictive setting.