Recurrent Autoencoder Networks for Koopman Spectral Analysis

Sam Otto*, Bill Eggert†
Princeton University
Department of Mechanical and Aerospace Engineering
Email: *sotto@princeton.edu, †weggert@princeton.edu

***Abstract*—A novel neural network architecture is presented for data-driven analysis of high-dimensional nonlinear dynamical systems. The architecture consists of two networks – an encoder and a decoder together with linear recurrence in the hidden encoded state. Consequently, the network learns nonlinear transformations into and out of a low-dimensional intrinsic space in which the time evolution is linear. The transformations learned by this architecture are deeply connected with Koopman operator theory. The encoder network learns an invariant subspace of the Koopman operator consisting of sufficiently rich features to reconstruct the full state through a nonlinear decoding transformation. This relaxes the typical strong assumption that the full state be linearly reconstructible in the subspace, allowing the subspace to be very low-dimensional and learned from the available data. The network was trained and tested on two artificial data sets possessing underlying linear dynamics nonlinearly embedded in high-dimensional images. Once trained, the network was used to find the leading Koopman eigenvalues, eigenfunctions, and approximate Koopman modes of the systems. Low-dimensional linear dynamics in the hidden state enabled efficient and accurate predictions to be made many time steps into the future.**

I. Introduction

An important aim of science is to understand and make predictions about the evolution of systems found in nature. This understanding is typically manifested in a mathematical model whose terms relate to some underlying theory. While theory allows us to gain intuition about the system, the model is used to make predictions and perform qualitative analysis. Provided with a new system having unknown characterstics, we begin by making observations and proposing an explanation. Using this hypothesis, a candidate model is formulated and compared to the data. If the model's predictions agree with the data, then we accept the model and the corresponding expanation. This approach, known as the scientific method, is highly successful when the system in question is composed of simple parts whose behavior is understood. Such systems enable hypotheses appealing to the underlying physics of their component parts. Consequently, scientific models are most useful for small-scale systems with few component parts or large-scale systems with so many parts that averaging or renormalization may be employed.

In many cases, these exists an intermediate regime where small-scale interactions are still relevant, but the theoretical models are too costly to be useful in practice. This is the regime where physical systems tend to exhibit complex and highly nonlinear emergent behavior. The study of emergence and dynamics of systems in the intermediate regime is at the fronteir of modern science and new approaches are required. At our disposal, we often have a large quantity of data consisting of rich measurments coming from experiment or meticulous simulation of the small-scale physics. Therefore, a data-driven approach for efficiently modeling the approximate behavior of the system is desired. Furthermore, a decomposition of the system into simple component parts is sought in order to constitute a satisfactory theory.

It is natural to consider a dynamical system which carries the measurements to their values at a future time. Most techniques for discoving the dynamical system rely on nonlinear regression. While regression allows for predictions to be made, the model is a black box which gives no further intuition about the system. Instead, one may ask whether the high-dimensional nonlinear system in question is actually a low-dimensional linear system in an intrinsic feature space. If such a linear system and corresponding nonlinear transformation can be found, standard techiques for linear system analysis can be applied in the transformed space. Eigenvectors and eigenvalues of the underlying linear system will correspond to nonlinear modes of the original system along with their frequencies and rates of decay. These modes are akin to the persisitent discrete parts which are required for a satisfactory theory of the system. Standard techniques for linear model reduction may also be applied to remove transient and low-energy modes. This effectively neglects less salient parts of the system which do not contribute significantly to the observed variations.

Discovering an underlying linear dynamical system using observational data is the idea behind Dynamic Mode Decomposition (DMD) and its variants. These techniques construct a finite-dimensional approximation of the Koopman operator which carries any observable function of the state to its value one time step in the future. The Koopman operator is linear, but infinite dimensional since it acts on the space of all possible functions of the state. Extended Dynamic Mode Decomposition (EDMD) [2] uses a dictionary of chosen observable functions to construct a finite-dimensional approximation of the Koopman operator on the subspace spanned by these functions. This may also be accomplished in a high-dimensional Hilbert space of observables using Kernel Dynamic Mode Decomposition (KDMD) [3]. Here, the subspace

used to approximate the Koopman operator is implicit in the choice of kernel. The subspace used is critical to the performance of EDMD and KDMD, yet there are no known methods for selecting an appropriate dictionary. It can be difficult to choose a sufficiently rich dictionary when using EDMD to capture highly nonlinear dynamics. On the other hand, we may choose a very rich dictionary with KDMD. However, due to the shallow architecture, many functions are needed and may over-fit the data, producing spurious modes. Additionally, an excessively high dimensional subspace must be kept in order to linear reconstruct the state from the rich feature space embedding.

This paper investigates using deep neaural networks to discover underlying linear dynamics in an intrinsic feature space from collected data. A new neural network architecture combining an autoencoder with linear recurrence in the hidden state is proposed for this purpose. The autoencoder provides the transformation into and out of the intrinsic space. Conseqeuntly, the encoder defines a small dictionary of nonlinear observables whose evolution is linear. Furthermore, these observables contain sufficiently rich information to approximately reconstruct the state using the nonlinear decoder. Once trained, the dynamics of the system may be predicted by encoding the current state and applying the linear update for each desired time increment. The prediction is formed by decoding the predicted hidden state. Using this approach, complex nonlinear features of the data are decoupled from underlying linear dynamics enabling theoretical analysis beyond what is possible with regression-based state updates.

In this paper, we implement this technique and use it to analyze two preliminary artificially constructed data sets. These data sets are made by embedding an underlying low-dimensional linear dynamical system nonlinearly in a high-dimensional space. The aim is to recover the underlying linear system and nonlinear transformation using the generated data and the proposed neural network. Embeddings and linear dynamics of varying complexity are investigated. Spectral analysis is performed on the learned approximation of the Koopman operator in order to recover its eigenvalues, eigenfunctions, and modes.

II. The Koopman Operator

The Koopman operator forms the theoretical basis for our investigation of nonlinear dynamical systems in an intrinsic feature space. Suppose that we have a discrete time autonomous dynamical system on state space $\mathcal{M} \subset \mathbb{R}^n$ given by the evolution function

$$x_{t+1} = f(x_t) \tag{1}$$

An observable is defined to be any function of the state space $\phi : \mathcal{M} \to \mathbb{R}$. Observables may include direct measurements of the system's state as well as nonlinear functions of them. The Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ acts on the space of observable functions by composing them with evolution function. Taking any $\phi \in \mathcal{F}$ and applying the Koopman operator, we get

$$\mathcal{K}\phi = \phi \circ f \tag{2}$$

We observe that the Koopman operator is linear and captures all of the information about the dynamics. However, it is infinite dimensional since it acts on the space of all observables. If the action of the Koopman operator is known on a subspace of observables containing the individual state variables, then the dynamics of the system can be reconstructed

$$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_n(x) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(n) \end{bmatrix} \Longrightarrow \mathcal{K}g = f \tag{3}$$

The full state observable $g \in \mathcal{F}$ links the Koopman operator on the space of observables $\mathcal{F}$ to the dynamical system on the state space $\mathcal{M}$.

In order to approximate the Koopman operator using a computer, we must choose a dictionary $\mathcal{D} = \{\psi_1, \psi_2, ...\}$ of observable functions spanning a subspace $\mathcal{F}_\mathcal{D} \subset \mathcal{F}$. Dynamic mode decomposition and its variants are regression-like procedures used to infer the action of the Koopman operator on this subspace from data snapshot pairs $\{(x_t, x_{t+1})\}_{t=1}^m$. The chosen dictionary must be sufficiently rich to capture the dynamics of the system and contain the full state observable in its span.

Ideally, we would like the find a Koopman invariant subspace so that successive Koopman operations do not quickly leave the subspace. It is therefore natural to look for eigenfunctions of the Koopman operator. Unless the eigenfunctions are known ahead of time, there is no way of ensuring that they may be found in $\mathcal{F}_\mathcal{D}$. If $\varphi(x)$ is a Koopman eigenfunction with eigenvalue λ then

$$\mathcal{K}\varphi = \varphi \circ f = \lambda\varphi \quad \Longrightarrow \quad \varphi(x_{t+\tau}) = \lambda^\tau \varphi(x_t) \tag{4}$$

If by some miracle, the full state observable is in the span of the Koopman eigenfunctions, then we may write the state as a linear combination

$$g(x) = \sum_{j=1}^{\infty} \xi_j \varphi_j(x) \tag{5}$$

The vectors $\xi_j \in \mathbb{R}^n$ are called the Koopman modes. Applying the Koopman operator τ times allows us to reconstruct the dynamics.

$$f^\tau(x) = \mathcal{K}g(x) = \sum_{j=1}^{\infty} \xi_j \lambda^\tau \varphi_j(x) \tag{6}$$

Of course, the assumption that the full state observable is in the linear span of the Koopman eigenfunctions is very strong. In this paper, we will relax this assumption and search for

a small Koopman invariant subspace with sufficent information to reconstruct the full state observable with a nonlinear transformation. A deep neural network is used to learn a small set of nonlinear observables $\Phi(x) = [\phi_1(x), \phi_2(x), ..., \phi_q(x)]$ spanning a Koopman invariant subspace $\mathcal{F}_\Phi$. The action of the Koopman operator is therefore given by

$$\mathcal{K}\Phi(x) = \Phi(x)A \tag{7}$$

Hence, for any observable $\phi = \Phi(x)u \in \mathcal{F}_\Phi$ the action of the Koopman operator is given by $\mathcal{K}\phi = \Phi(x)Au$. Furthermore, the subspace of observables which we aim to learn is sufficiently rich to recover the full state through a learned nonlinear decoding transformation $g = \tilde{\Phi} \circ \Phi$.

If $A = V\Lambda V^{-1}$ is diagonalizable with right eigenvectors $V = [v_1, ..., V_q]$ and left eigenvectors $W = [w_1, ..., w_q]$, $W^* = V^{-1}$ then it is easily shown than $\varphi_j(x) = \Phi(x)v_j$ is a Koopman eigenfunction. The dynamics can be reconstructed as follows

$$f^\tau(x) = \tilde{\Phi}\left(\Phi A^\tau\right) = \tilde{\Phi}\left(\Phi V \Lambda^\tau W^*\right) \tag{8}$$

The left eigenvectors are the analogues of Koopman modes for dynamics in the encoded variables $z = \Phi(x)$, $\tilde{g}(z) = [z(1), ..., z(q)]^T$.

$$\tilde{g}(z) = \sum_{j=1}^{q} \bar{w}_j(zv_j) \Longrightarrow \mathcal{K}^\tau \tilde{g}(z) = \sum_{j=1}^{q} \bar{w}_j \lambda_j^\tau (zv_j) \tag{9}$$

III. Description of the Data

An easy way to create high-dimensional embeddings of low-dimensional states is to use images. A small linear system with known transition matrix is simulated. The resulting states are then used to scale and shift different patterns in an image. Pure scaling of a pattern produces a linear embedding of the state since the image magnitudes combine linearly with linear combinations of the state. One way to create a nonlinear embedding is to scale a pattern by a nonlinear function of the state. A nonlinear embedding can also be created by translating the pattern in space according to the underlying state. For the case of translations, if the underlying state has a periodic evolution, then the embedding can be described by a different linear dynamical system whose dimension is equal to the periodicity. This is because the images produced will have a finite number of configuratons less than or equal to the periodicity of the underlying system. These distinct configurations could be taken as patterns themselves whose contributions to the image are scaled according to a different underlying system. If this system is taken to be a cyclic permutation, then the dynamics are recovered. However, if the underlying system is chosen to have irrational frequency, then it cannot be periodic. While the embedding could be considered linear in the above sense, it would be infinite-dimensional to account for each distinct configuration. We would prefer for the autoencoder to recover the low-dimensional system associated with the translation.

One difficulty with using a direct image embedding of a system's state is the non-uniqueness of the next image given complete information about the current one. Consider a movie of a swinging pendulum. If you are given a snapshot when the pendulum is at the bottom of its swing, there is no way to tell whether it will go left or right in the next frame. This problem motivates us to use time-delayed copies of the generated images rather than individual snapshots. The Takens embedding theorem [1] shows that a smooth attractor of the original dynamical system can be fully reconstructed up to differomorphism using sufficiently many time-delayed observations. In other words, it is theoretically possible to recover the underlying dynamics if sufficiently many time-delays are used. In this report, each example consists of an image at time t and several time-delayed copies $t-1, t-2, ..., t-\tau$.

A. Simple Nonlinear Embedding

The first test case consists of a simple oscillatory dynamical system with period $T = 20$. We construct a single complex conjugate pair of eigenvuales of the unit circle $\lambda_{1,2} = \exp(\pm i\Delta\theta)$ with $\Delta\theta = 2\pi/T = \pi/10$. The hidden state dynamics are evolved according to the discrete time system

$$h_{t+1} = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) \\ -\sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} h_t \tag{10}$$

A new initial state was chosen randomly every 50 time steps to generate the data. At every time step an 11×11 snapshot s_t was formed by centering a Gaussian "blip" at h_t in the image

$$s_t(i,j) = 5 * \exp\left(\frac{1}{4}\|(i-6, j-6) - 3h_t\|^2\right) \tag{11}$$

The image snapshots were vectorized to form row vectors $x'_t \in \mathbb{R}^{121}$ and then time delayed to form the data row vectors $x_t = \left[x'_{t-2}, x'_{t-1}, x'_t\right] \in \mathbb{R}^{363}$. The training data consists of $m_{\text{train}} = 49997$ time-delayed snapshot pairs $\{(x_t, x_{t+1})\}_{t=1}^{m}$. The testing data consists of $m_{\text{test}} = 1997$ pairs. Figure 1 shows an example time-delayed image sequence from the training set.

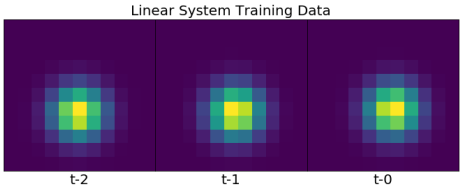

Fig. 1. Simple nonlinear time-delayed image embedding example

B. Challenging Nonlinear Embedding

The second test case was made from time-delayed images in the same way as the first test case. This case was made more difficult by introducing a richer underlying dynamical system with two frequencies having eigenvalues

$\lambda_{1,2} = 0.95\exp(\pm i\pi/10)$ and $\lambda_{3,4} = \exp(\pm i\pi/(5\varphi))$ where $\varphi = (1+\sqrt{5})/2$ is the golden ratio. The golden ratio is the most irrational number since it is poorly approximated by fractions with all finite bases. The existence of an irrational frequency means that the underlying system has a $1D$ attractor which is topologically mixing. This is known to be a very difficult case for variants of the DMD algorithm. The hidden state evolves according to

$$h_{t+1}^{(1)} = 0.95 \begin{bmatrix} \cos(\frac{\pi}{10}) & \sin(\frac{\pi}{10}) \\ -\sin(\frac{\pi}{10}) & \cos(\frac{\pi}{10}) \end{bmatrix} h_t^{(1)} \tag{12}$$

$$h_{t+1}^{(2)} = \begin{bmatrix} \cos(\frac{\pi}{5\varphi}) & \sin(\frac{\pi}{5\varphi}) \\ -\sin(\frac{\pi}{5\varphi}) & \cos(\frac{\pi}{5\varphi}) \end{bmatrix} h_t^{(2)} \tag{13}$$

The image snapshots were formed by composing two two parts which are nonlinear functions of the state

$$s_t(i,j) = s_t^{(1)}(i,j) + s_t^{(2)}(i,j) \tag{14}$$

$$s_t^{(1)}(i,j) = 5\exp\left(\frac{1}{4}\|(i-6, j-6) - 3h_t^{(1)}\|^2\right) \tag{15}$$

$$s_t^{(2)}(i,j) = \left[h_t^{(2)}(1)\right]^3 \sin\left(\frac{2\pi}{4}j\right) \tag{16}$$

The training and testing data were created in precisely the same way as the easier case with a time delay embedding with three snapshots. Figure 2 shows an example time-delayed image sequence from the training set.

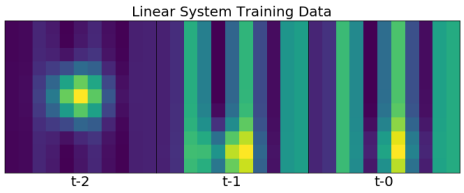

Fig. 2. Challenging nonlinear time-delayed image embedding example

IV. Recurrent Autoencoder Networks

A neural network architecture has been designed which aims to discover the underlying linear system through a nonlinear transformation of the data. This transformation is accomplished using a multiple-layer feed-forward network which we call the encoder $z = \Phi(x)$. The dynamics in the encoded hidden state should be linear. Therefore, we need to learn a matrix A which takes at hidden state z_t to its value at the next time step

$$\Phi(x_{t+1}) = z_{t+1} \approx \hat{z}_{t+1} = z_t A \tag{17}$$

Furthermore, the hidden state must contain sufficient information to reconstruct the state at the current and future time steps. Therefore, we must also learn an approximate inverse transformation $\hat{x} = \tilde{\Phi}(z)$. This transformation from the intrinsic space back into the empirical space is again a multi-layer neural network. Therefore, a prediction τ steps into the found by encoding with Φ, advancing the encoded state with A, and finally mapping back into empirical space with $\tilde{\Phi}$

$$\hat{x}_{t+\tau} = \tilde{\Phi}\left(\Phi(x_t)A^\tau\right) \tag{18}$$

A novel architecture is proposed for simultaneously learning the above transformations and hidden state transition matrix. If we were to naively use a feed-forward network with one or more encoded state updates, we have no way of separating the encoding, decoding, and update processes. In other words, we do not want the encoder and decoder to take part in the update process. We therefore have to simultaneously force the encoder and decoder to reconstruct the data while enabling forward transitions of the encoded state using the linear operator. This suggests an architecure similar to a recurrent neural network. An example is encoded and decoded after each application of the linear update. Minimizing a loss including decoded predictions $0, 1, ..., k$ steps out decouples the update and nonlinear transformation processes.

Minibatch stochastic gradient descent is used with back propagation to learn the weights in the encoder and decoder networks along with the entries in the state transition matrix. The RmsProp algorithm was used for optimization with a decaying learning rate. Experiments have shown that this architecture does not provide sufficient feedback on the hidden state to ensure that it agrees with the encoder after applying the linear state transition operator. Therefore, we also encode the future states include a term in the loss function measuring how much the updated hidden states differ from the true encoded future states. The following loss function was minimized

$$L(x_t) = \sum_{\tau=0}^{k} \alpha^\tau \left[L^{(1)}(x_{t+\tau}, \hat{x}_{t+\tau}) + \beta L^{(2)}(\Phi(x_{t+\tau}), \hat{z}_{t+\tau})\right] \tag{19}$$

$$L^{(1)}(x_{t+\tau}, \hat{x}_{t+\tau}) = \|x_{t+\tau} - \hat{x}_{t+\tau}\|^2 \tag{20}$$

$$L^{(2)}(z_{t+\tau}, \hat{z}_{t+\tau}) = \|z_{t+\tau} - \hat{z}_{t+\tau}\|^2 \tag{21}$$

The parameter $0 < \alpha \leq 1$ is used to prioritize short-term predictions. We set $\beta = n/q$ to be the ratio of the full state dimension to the encoded hidden state dimension. This was done in order to prioritize both error terms equally. The architecture of the unfolded neural network is shown in figure 3

The encoder and decoder are not only responsible for reducing the dimension, but also to map into a space where the dynamics are linear. This transformation may be highly complex and nonlinear. The encoder and decoder used in this report are fully-connected feed-foward networks using $\tanh$ activation functions at each layer. The encoder and decoder were chosen to have the same number of layers and mirrored widths, though this is by no means a requirement. In order to account for scaling, the decoder has a learned scaling factor

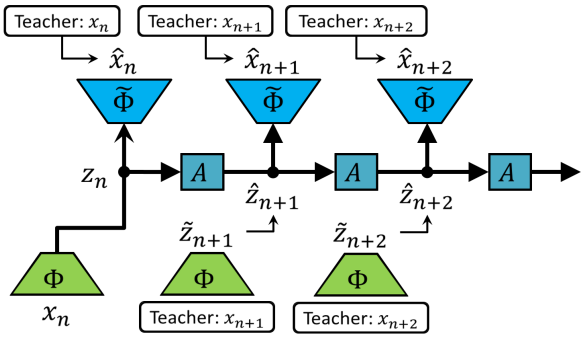

Fig. 3. Unfolded Network Architecture

and bias at each output node. The matrix A was initialized to have evenly distributed eigenvalues around a circle of radius 0.8 in the complex plane. This was done to ensure that the learned eigenvalues would remain stable, while being able to reach all points in the unit circle easily over the course of training.

A. Koopman Analysis using the Network

As was shown in the above section, the recurrent autoencoder network is trained to learn a small invariant subspace of the Koopman operator with sufficiently rich observables to decode the full state through $\tilde{\Phi}$. If the matrix $A = V\Lambda V^{-1} = V\Lambda W^*$ is diagonalizable, then eigenfunctions of the Koopman operator are given by

$$\begin{bmatrix}\varphi_1(x) & \varphi_2(x) & \cdots & \varphi_q(x)\end{bmatrix} = \Phi(x)V \tag{22}$$

with corresponding eigenvalues $\lambda_1, \lambda_2, ..., \lambda_q$.

However, we note that the true full state observable has little chance of being in the linear span of the small invariant subspace $\mathcal{F}_\Phi$ which we learn. This makes it difficult to find the Koopman modes using this method. For the sake of analogy, we may attempt to linearly reconstruct the full state observable as a linear combination of $\{\phi_1(x), \phi_2(x), ..., \phi_q(x)\}$ using regression. If

$$g(x) \approx B^T\Phi(x)^T \tag{23}$$

then the action of the Koopman operator on the full state observable is approximated by

$$f(x) = \mathcal{K}g(x) \approx B^T\bar{W}\Lambda V^T\Phi(x)^T = \sum_{j=1}^{q} B^T\bar{w}_j\lambda_j\varphi_j(x) \tag{24}$$

Therefore, $\xi_j \approx B^T\bar{w}_j$ is an approximation of the jth Koopman mode. We proceed to find the matrix B using regression on the training data. Define the following data matrices from training examples $\{x_j\}_{j=1}^m$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \in \mathbb{R}^{m\times n} \qquad \Phi(X) = \begin{bmatrix} \Phi(x_1) \\ \Phi(x_2) \\ \vdots \\ \Phi(x_m) \end{bmatrix} \in \mathbb{R}^{m\times q} \tag{25}$$

Minimizing $\|X - \Phi(X)B\|_F$ over $B \in \mathbb{R}^{q\times n}$ gives

$$B = \Phi(X)^+X \tag{26}$$

where $\Phi(X)^+$ denotes the Moore-Penrose pseudoinverse. Therefore, the approximate Koopman modes form the columns of

$$\Xi = B^T\bar{W} \tag{27}$$

and the dynamics can be approximately reconstructed as in eqn. 6. This method of reconstruction will be compared to the reconstruction given by the decoder eqn. 8. We expect the decoder to give more accurate reconstructions. However, the approximate Koopman modes may give deeper insight into the system's dynamics. One possibility for generating better approximations of the Koopman modes is to augment the observables $\Phi(x)$ with additional nonlinear observables $\Psi(x)$ creating a larger dictionary which could be used to perform EDMD. This will be a subject for future work.

V. Results

A. Simple Nonlinear Embedding

The network used to analyze the simple nonlinear embedding test case had a hidden state dimension $q = 5$. The encoder and decoder each had three hidden layers with mirrored widths $[100, 60, 20]$ and $[20, 60, 100]$ respectively. The RNN was unfolded for $k = 5$ time steps into the future with a decay factor $\alpha = 0.9$.

A random example was chosen from the testing data. The ground truth state $k = 10$ steps into the future is compared to the prediction made by the trained RNN in Figure 4. The agreement appears to be very close upon visual inspection. The same reshaped data vectors are plotted against each other in figure 5 as well as for $k = 35$ time steps in the future. The reconstruction using equation 6 and approximate Koopman modes computed from eqn. 27 are also shown in this figure. The dots represent the ground truth image, while the red and blue lines are the RNN and Koopman reconstructions respectively. Nonlinear reconstruction using the decoder eqn. 8 from the learned Koopman-invariant subspace reconstructs the state more accurately than the approximate Koopman modes as expected. On this example, the RNN predictions using eqn. 8 remain accurate even over a long time interval. The accuracy of the two reconstruction methods in terms of mean square pixel error is plotted in figure 6 over a range of time intervals. The prediction errors using the RNN reconstruction remain small over time. Therefore, we can conclude that successive

applications of the Koopman operator did not leave the learned subspace $\mathcal{F}_\Phi$, suggesting that it is indeed Koopman-invariant.

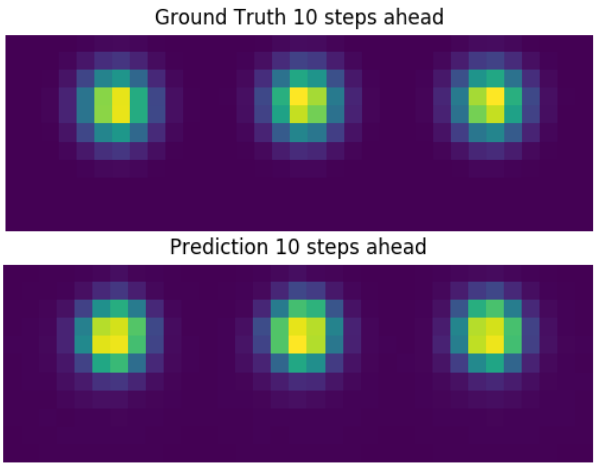

Fig. 4. Ground truth and RNN predicted state

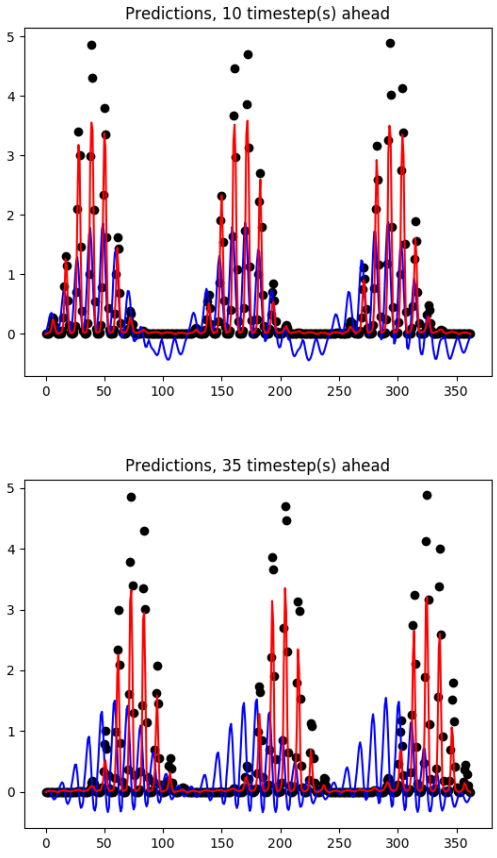

Fig. 5. Ground truth, RNN, and Koopman predicted states

The eigenvalues of the learned A matrix are also approximate eigenvalues of the Koopman operator. The five eigenvalues on the learned invariant subspace are plotted in the complex plane in figure 7. Two of the eigenvalues lie on the unit circle at approximately $\Delta\theta = \pm\pi/10$ indicating that the underlying frequency has been recovered. The corresponding approximate Koopman modes are plotted in figure 8. The first mode is approximately constant in time and is rotationally symmetric. The second mode accurately captures the desired rotation. However, as was observed in the above error analysis, three Koopman modes, though informative, are not sufficient to linearly reconstruct the data. The nonlinear decoding process from the hidden state is much more successful.

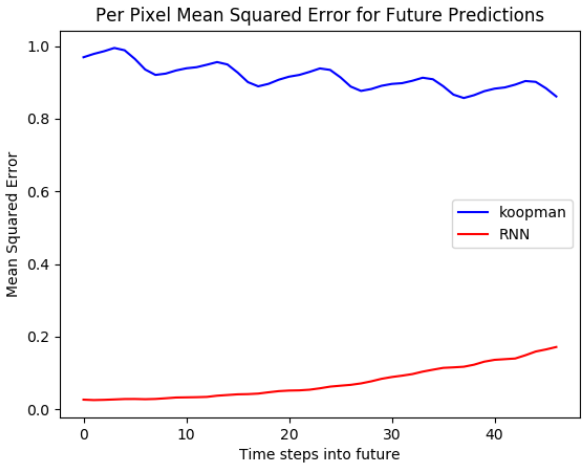

Fig. 6. Mean square prediction error using two methods

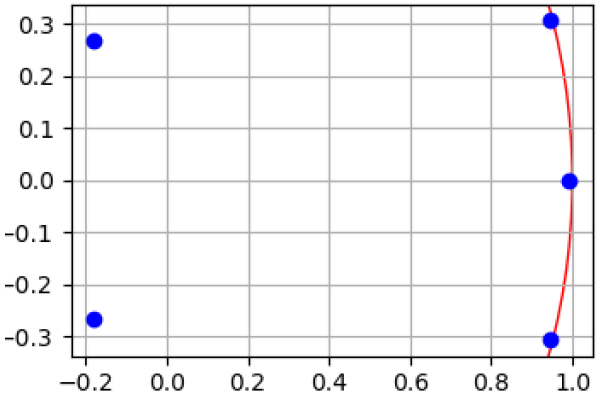

Fig. 7. Eivenvalues of Learned Koopman Operator

B. Challenging Nonlinear Embedding

The network used to analyze the challenging nonlinear embedding test case had a hidden state dimension $q = 10$. The encoder and decoder each had five narrow hidden layers with mirrored widths $[50, 20, 10, 10, 10]$ and $[10, 10, 10, 20, 50]$ respectively. The RNN was unfolded for $k = 5$ time steps into the future with a decay factor $\alpha = 0.8$.

A random example was chosen from the testing data. The ground truth state $k = 9$ steps into the future is compared to the prediction made by the trained RNN in Figure 9. The

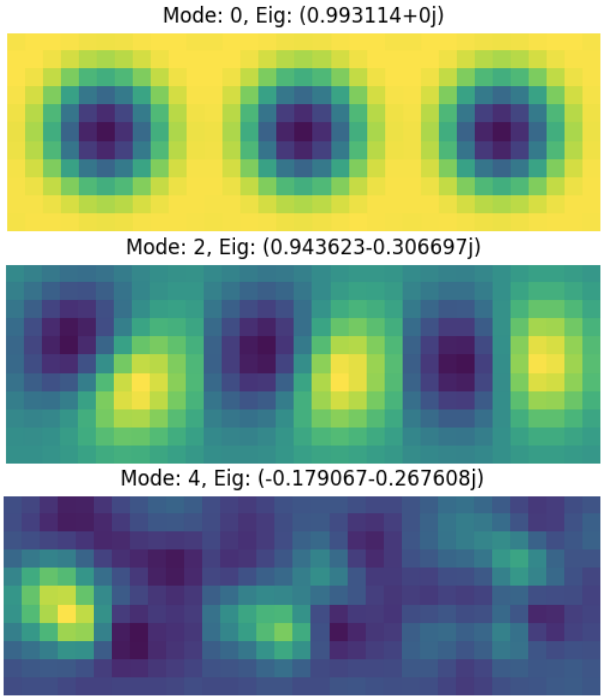

Fig. 8. Approximate Koopman modes

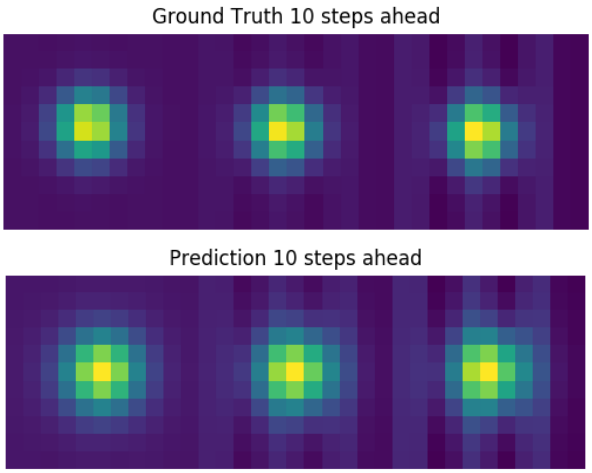

Fig. 9. Ground truth and RNN predicted state

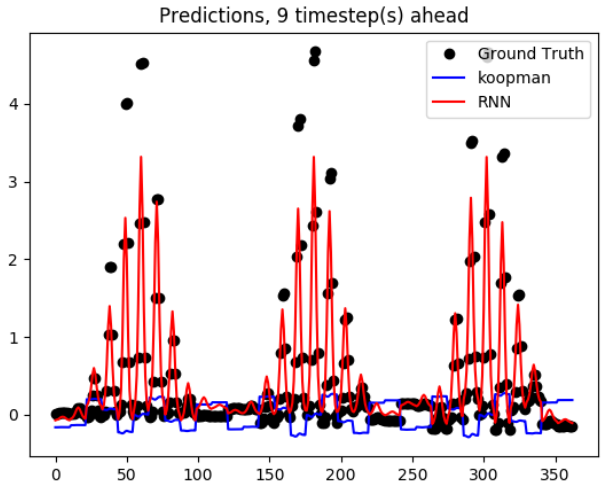

Fig. 10. Ground truth, RNN, and Koopman predicted states

agreement appears to be very close upon visual inspection. The same reshaped data vectors are plotted against each other in figure 10. The reconstruction using equation 6 and approximate Koopman modes computed from eqn. 27 are also shown in this figure. The dots represent the ground truth image, while the red and blue lines are the RNN and Koopman reconstructions respectively. Nonlinear reconstruction using the decoder eqn. 8 from the learned Koopman-invariant subspace reconstructs the state very accurately while the approximate Koopman modes are totally unsuccessful at reconstruction. On this example, the RNN predictions using eqn. 8 remain accurate even over a long time interval. The mean accuracy of the two reconstruction methods is plotted in figure 11 over a range of time intervals. The predictions using the RNN reconstruction remain very accurate over time. Therefore, we can again conclude we have learned a Koopman invariant subspace. The reconstructions using the Koopman modes are poor because there are not enough Koopman modes to linearly reconstruct the state.

The eigenvalues of the learned A matrix are are plotted in figure 12. Two of the eigenvalues lie near the unit circle at approximately $\Delta\theta = \pm\pi/10$ and radius $r \approx 0.95$ indicating that one of the the underlying frequencies and rates of decay has been recovered. Another pair of eigenvalues lies near the unit circle at an angle approximately $\Delta\theta = \pm 3\pi/(5\varphi)$. This seems to agree with the cube of the second pair of eigenvalues used to generate the data, though there is some decay when there should not be. Some of the corresponding approximate Koopman modes are plotted in figure 13. While insufficient in span to linearly reconstruct the state, the Koopman modes do reveal interesting features of the dynamical system. Different modes pick out the striped pattern, the Gaussian blip and the changing behaviors from frame to frame.

VI. Discussion

For both the simple and challenging nonlinear embeddings, it was possible to train the proposed architecture and make accurate predictions many time steps into the future. This agreement is evident in the randomly chosen test data examples plotted in for both cases as well as in the low mean square pixel value errors. The linear Koopman reconstruction was unable to recover the state in both cases. Therefore, the introduction of a nonlinear decoder from the learned low-dimensional intrinsic space enables much more accurate reconstructions using a small encoded state dimension.

In both example systems, we have recovered useful Koopman eigenvalues of the underlying systems. The other spurious eigenvalues were small in magnitude and correspond to fast transient modes. The approximate Koopman modes were not

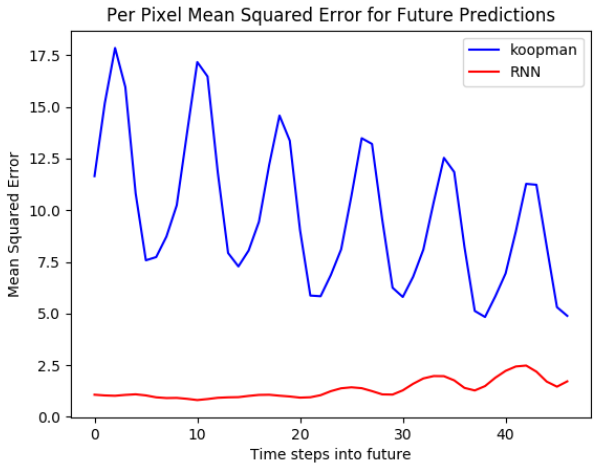

Fig. 11. Mean square prediction error using two methods

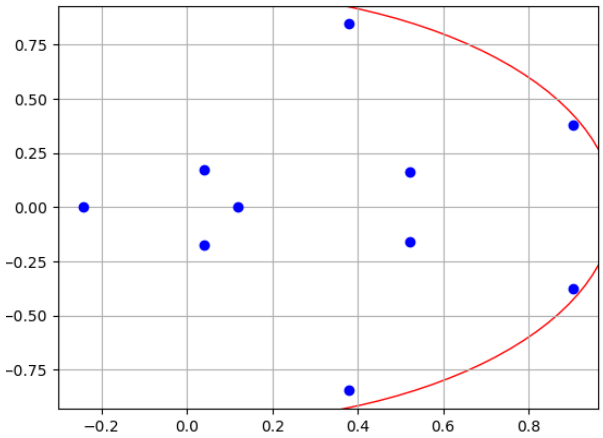

Fig. 12. Eivenvalues of Learned Koopman Operator

useful for reconstructing the predicted dynamics in either case. However, surprisingly, they did illustrate physically relevant dynamics of the system. For example the leading mode in the simple case has the correct eigenvalue and clearly shows a rotation. This rotation agrees with the constructed rotation of the Gaussian blip in the image data. This suggests that we are learning a correct subset of the actual Koopman modes, but not sufficiently many modes to get accurate reconstructions.

It should also be noted that is was difficult to converge on good parameters for training the networks. The results were sensitive to all of the parameters. The most sensitive parameters were the learning rate, learning rate decay, and minibatch size. It was important that the encoder and decoder had sufficient depth to learn the desired nonlinear transformations. However, the widths were less important and were chosen to make training as quick as possible. Significant width may not have been necessary since the data consisted of only a few dominant features. However, depth was needed to provide sufficient nonlinearity.

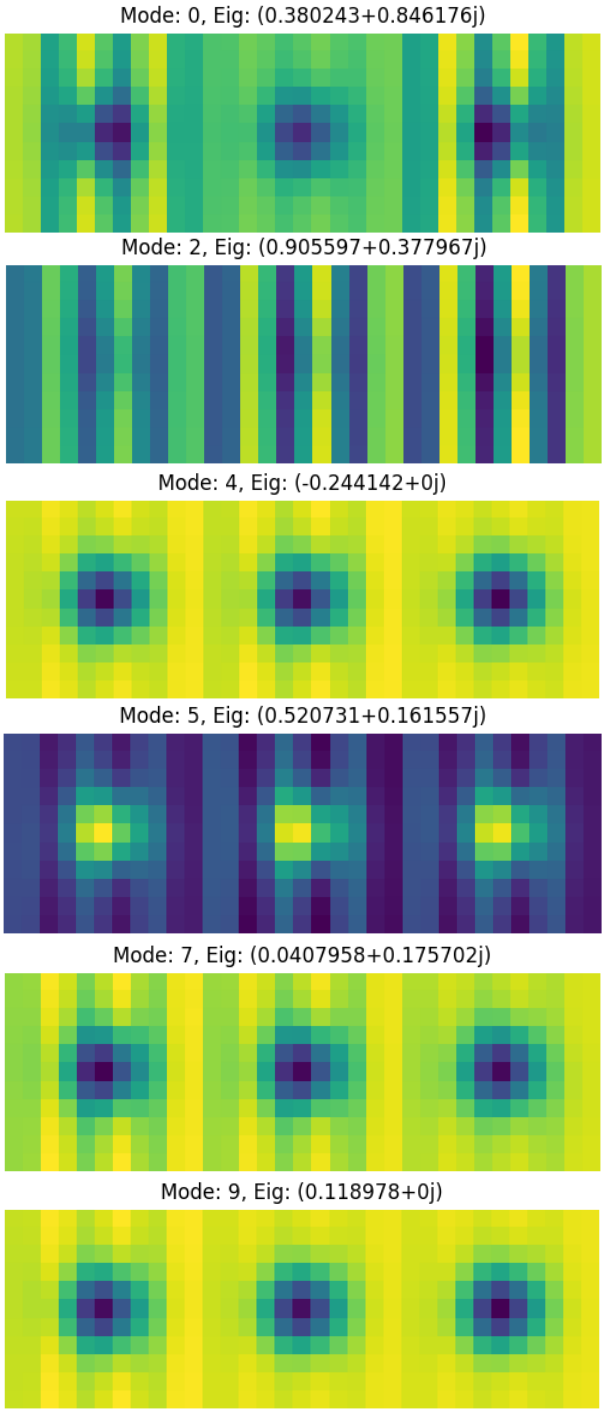

Fig. 13. Approximate Koopman modes

VII. Further Work

Future work certainly includes a more thorough investigation of the parameter space regarding the encoder and decoder layers as well as the training parameters. Deeper networks using dropout and layer-skipping connections will be investigated. Different optimization algorithms will be tried along with ReLu units instead of tanh. Since the data comes

from spatially and temporally contiguous sampling locations, it may be useful to introduce convolutional layers for feature detection and generation in the encoder and decoder. These layers would form the first few layers of the encoder and the last few layers of the decoder. The convolution could also be extended across the time dimension in the case of delay embeddings. This may be used to detect how coherent features are moving in space.

In terms of training methodology, two additional methods will be investigated. The first is to use an autoencoder to perform dimensionality reduction prior to training. This should produce good features for the proposed recurrent network to use right out of the gate. The encoder and decoder would then be responsible for further transforming the feature representations so that they evolve linearly in time. In other words, separating the tasks of learning features and then a nonlinear transformation may boost performance and reduce the time needed to train. A related idea is to employ greedy layer-wise training to achieve more depth in the encoder and decoder.

While there is reason to believe that the approximated Koopman modes are correct, they do not span a sufficiently large space to be able to reconstruct the full-state observable. However, they provide useful physical information about the system. Furthermore, the learned observables provide a very useful invariant subspace of the Koopman operator. The nonlinear functions comprising the encoder may be used to enrich an existing dictionary used for Extended Dynamic Mode Decomposition (EDMD). Together the functions may span a rich enough space to do improved linear reconstruction using the Koopman modes. Finally, the results using this method will be compared to EDMD and Kernel Dynamic Mode Decomposition (KDMD). Though we expect to be able to reconstruct the state from a smaller encoded dimension due to the use of a nonlinear decoder. The methods will also be tested on different data sets coming from real world dynamical systems.

VIII. Conclusion

A novel neural network architecture has been designed and implemented for applications to high-dimensional nonlinear dynamical systems. The architecture is intended to learn an invariant subspace of the Koopman operator consisting of sufficiently rich features to reconstruct the full state through a nonlinear transformation. The method was evaluated on two artificial dynamical systems – each having an underlying low-dimensional linear system. Though difficult to train, the network was able to learn transformations into and out of an intrinsic space of features whose dynamics are linear. The eigenvalues of the linear dynamics learned from data in the encoded state reflected the true eigenvalues used to generate the data sets. The encoding network was able to learn an invariant subspace of the Koopman operator enabling accurate reconstruction of the dynamics many time steps into the future. The method is advantageous since the time-evolution of the modeled system is linear and low-dimensional. Predictions can be made by decoding the linear state only when they are needed. Furthermore, a regression technique was used to approximate a small set of Koopman modes associated with the learned Koopman eigenvalues and eigenfunctions. While these modes did not span a large enough space to linearly reconstruct the state, they did provide useful insights into the dynamical features of the high-dimensional systems.

References

[1] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[2] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[3] Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.