1 Executive summary

We discovered a new method of approximating partial differential equations using deep neural networks. These methods are able to predict turbulent flow as effectively as previous methods without relying on complex memory gates instead using an approximate form of partial differential equations (PDEs). Additionally, by nature of approximating an arbitrary PDE, solutions fit a powerful class of equations capable of modeling many physically based systems.

1.1 Formulating Approximation of Partial Differential Equations

1.1.1 What was to be tested? What was the expected outcome prior to testing? Previous results used sequence-to-sequence models to learn an embedding of the current state, as well as a model of the dynamics using recurrent networks such as Long-Short Term Memory units (LSTMs). This worked well for prediction, however recurrent neural networks represent a much more complex function approximation scheme. While RNNs have been shown to approximate dynamical systems [Sonoda & Murata (2017)], complex and non-linear mechanics such as forget-gates do not map well to the discovery of physical laws.

1.1.2 High-level summary of main results. Explicitly modeling the dynamics as a PDE allows for interpolation by adjusting $\triangle t$ in the discretized time derivative. Additionally, because many physically based systems are easily modeled using PDEs, the class of problems is well understood and forms a basis for many natural processes such as diffusion, electrostatics, or fluid dynamics.

1.1.3 Discuss relevance to project and DARPA concerns. This formulation moves previous models towards more explainable and better understood formulations simultaneously reducing the number of learned parameters in the model. Solutions that offer good representation using this scheme can be more easily reasoned about and require fewer calculations to infer.

1.2 Evaluating Predictions of Approximated Partial Differential Equations

1.2.1 What was to be tested? What was the expected outcome prior to testing? We evaluate new approximate PDE based architecture against the previous recurrent model over different sequence lengths comparing against the best LSTM based architecture found previously.

1.2.2 High-level summary of main results. Despite a reduction in complexity as well as strong spacial assumptions, the newly discovered architecture performs almost as well as the previous optimized model. Preliminary experiments already achieve a MSE of 0.0023 compared to an optimized MSE of 0.0016 from the best performing LSTM model over 20 time steps. Further results suggest that the assumption of spacial uniform dynamics is significantly impacting prediction, with a short, 5 step model achieving an excellent MSE of 0.000069, an improvement of over 95% compared to the best LSTM model.

1.2.3 Discuss relevance to project and DARPA concerns. These results show the new architecture is capable of predicting turbulent flow effectively, even when given fewer frames of history. Moreover, there is strong potential for further model improvements. Thus we maintain previous successes with predicting turbulent flow while simplifying the network architecture.

2 Achievements

2.1 Scientific Breakthroughs

None

2.2 Technology developments

None

2.3 Application results

More work is needed to determine if the discovered formulation is well suited for applications in sequence interpolation.

2.4 Transitions achieved

None

3 Lessons Learned

Problems encountered/risks that occurred, and corresponding solutions/mitigations
Open Issues

4 Next Steps

Incorporating local features in the dynamics model is necessary to correctly approximate partial derivatives, however, as the dynamics model is uniform across the entire domain, strong edge effects distort the dynamics. Developing a model capable of overcoming edge effects will allow the dynamics to more robustly rely on local features and could dramatically improve performance.
Current scheme uses only a single time derivative to approximate the dynamics. By incorporating higher order approximations, more information can be captured in the learned PDE.
The current formulation learns a discretized time derivative of the dynamics. If model is capable of interpolating between intermediate samples, this would be invaluable for both data compression and super-sampling sequences.
Previous formulations were robust to noise and missing samples. Following this work, we will compare noise resistance of the new fully convolutional formulation against the previous recurrent LSTM based model.

5 Technical details

5.1 Formulating Approximation of Partial Differential Equations

Consider learning a PDE of the form $u_t = f(u)$. The simplest approximation of $u_t = f(u)$ is to discretize the time derivative u_t by $\frac{u_{n+1}-u_n}{\triangle t}$ and approximate the right hand side by $f(u_n)$. This leads to the forward Euler scheme

$$u_{n+1} = u_n + \triangle t f(u_n)$$

where $f(u_n)$ is learned by a deep neural network. Instead of learning the dynamics of the system directly, we use a learned feature representation u_n composed of linear convolutions of the input history. This avoids the need for approximating the local partial derivatives of the system and instead allows the network to learn the optimal feature representation.

Let x represent h patches of history $x = [x_{t-h}, ..., x_{t-1}]$ and y represent the target sequence $y = [x_t, x_{t+1}, ..., x_{t+l-1}]$. We learn an encoder, $\phi(x) = u_0$, a decoder $\theta(u_n) = \hat{x}_n$ as well as a dynamics model $f(u_n)$ from equation 5.1.

We utilize a feature space of dimension m = 16, therefore $u_n \in \mathbb{R}^{16}$. Additionally, to ensure spatial continuity, we expand the domain of $f(u_n)$ to include the local 5x5 region centered about each pixel of a given patch, thus $f(u_n) \in \mathbb{R}^{16\times5\times5} \to \mathbb{R}^{16}$. Similarity for the encoder, we introduce local features by convolving 16 filters of shape 5x5x5 giving us $\phi(x) \in \mathbb{R}^{5*5*5} \to \mathbb{R}^{16}$. Finally, for the decoder $\theta(u_n)$ we simply apply a reduction over the feature space, giving $\theta(u_n) \in \mathbb{R}^{16} \to \mathbb{R}$

5.2 Evaluating Predictions of Approximated Partial Differential Equations

As shown in figure 4, despite the simplified PDE architecture, the performance of the new models is comparable, being more accurate than the LSTM based model in the near-term samples and slightly less accurate in the long term. As illustrated by figures 2 and 3 the fully convolutional composition causes edge effects that reduce performance in long horizons as any assumptions made by local features valid in the center of the patch are violated when evaluated near the edges of the patch.

References

Sho Sonoda and Noboru Murata. Double continuum limit of deep neural networks.ICML WorkshopPrincipled Approaches to Deep Learning, 2017

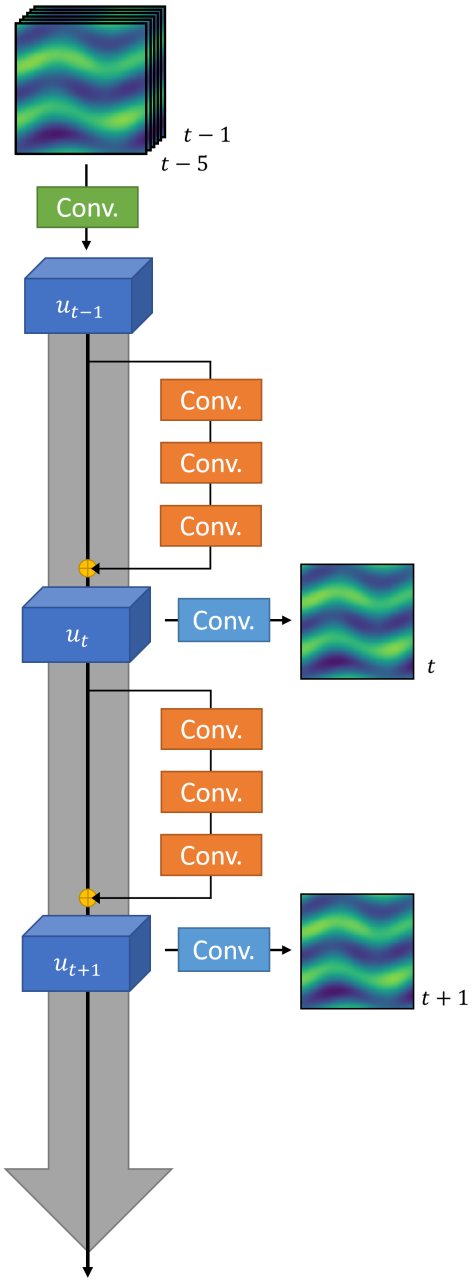

Figure 1: Illustration of the res-net inspired forward Euler architecture. The encoder $\phi(x)$ (shown in green) maps the input into the feature space. Three sequential convolutional layers (shown in orange) are used to approximate $f(u_n)$ and summed with u_n (shown in yellow) to give u_{n+1} as outlines in equation 5.1. At each step, the feature vector u_n is decoded by $\theta(u_n)$ (as shown in light blue) predicting x_n.

Figure 2: Mean absolute error over a single validation batch for PDE architecture. Because of the strong spatially invariant assumptions made by the model, we see strong edge effects near the borders.

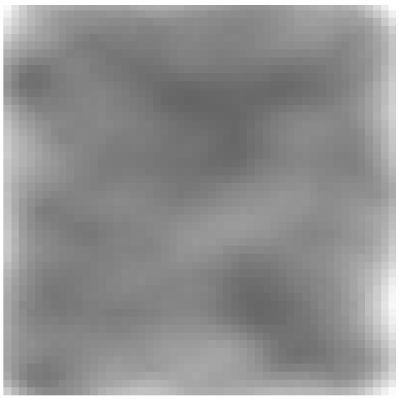

Figure 3: Mean absolute error over a single validation batch for LSTM architecture. Note the error is more uniform through the entire patch.

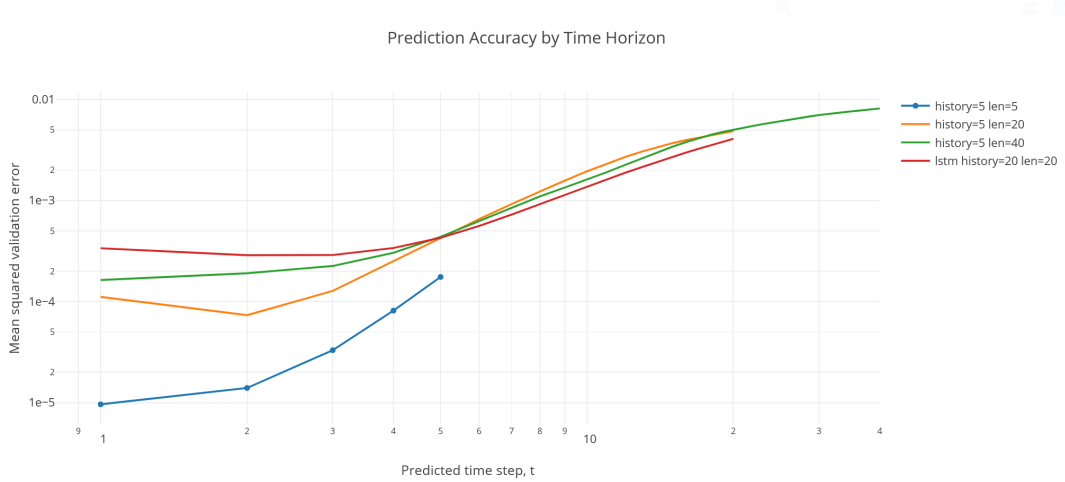

Figure 4: Performance of new PDE architecture for models trained trained using varying sequence lengths. Previous model LSTM model (in red) performs comparably to new model. The high accuracy shown by the short, 5-step prediction experiment suggests that compounding errors from edge effects are likely reducing long term prediction performance.