Machine learned surrogates for scaling solutions to dynamical systems through Fourier spectroscopy of inexactness

Ryan Pyle, Nikola Jovanović, Adam Duracz,
Krishna V. Palem, Devika Subramanian, Ankit B. Patel

Abstract

As machine learning systems become increasingly demanding in terms of computational resources, energy (or power) and running time are serious hurdles to scaling at a reasonable cost. *Inexact computing* (also referred to as *approximate computing*) has proven to be a very promising approach to overcoming these hurdles. This principle advocates trading a small amount of the quality of the solution for significant savings in resources. Historically, using commercial-off-the-shelf (COTS) hardware, inexactness regimes were realized using the size of the machine word or precision, quantified as the number of bits, However, characterizations of the relationship between degrees of inexactness or precision as it relates to quality have been ad-hoc. In this paper, our primary contribution is a principled approach, referred to as *spectroscopy*, for characterizing this relationship by using the Fourier spectrum of the data in the context of widely used machine learning systems for *predictive modeling* of time-series data. To this end, we choose the widely used Lorenz 96 system as our benchmark, which embodies chaotic behavior in that small changes to its input can cause large changes to the output (13). Our methods are experimental and include a detailed analysis of echo state networks (10), a method based on SINDy which we refer to as D2R2 (2) as well as LSTM networks (8). We demonstrate conclusively that by analyzing the energy spectrum of a dynamical system derived from its time series, the impact of inexactness on solution quality can be quantitatively characterized.

I. INTRODUCTION AND RELATED WORK

Machine learning has lately been explored extensively in the context of providing support in domains of physics and engineering. In these domains, most of the work has been based on computational models which involve solutions to systems of partial and ordinary differential equations (respectively PDEs and ODEs). In this context, the goals have been two-fold: The first goal has been to try and replace expensive solvers for differential equations with lower-cost machine learned *surrogates*. Lately, this approach has found favor in the weather prediction and climate modeling communities (30; 23; 6; 3) wherein the need to find low-cost solutions that scale poses a major hurdle to improving prediction horizons. A second goal is in contexts where models based on differential equations are not known and it is hoped that machine learning will help "discover" underlying structure and models.

In this paper, we are concerned with the first of these two goals, namely that of enabling scaling of solutions to differential equation based models through machine learned surrogates. Briefly, as deep network based solutions get increasingly complex, the infrastructure and computational resources needed to support them have grown dramatically and are increasingly a barrier to their continued scaling (27). So, to be viable as surrogates, effort has to be expended in lowering their cost which spans both the compute time and increasingly the *energy* and *power* costs. To this end, and returning to the theme of this paper, we focus on *inexact computing* which has proved to be a counter-intuitive approach to lowering energy consumption and running time while trading-off negligible to tolerable amounts of solution quality (16; 17; 11; 18).

Historically, inexactness has been acknowledged as an intriguing approach to problems in weather-prediction (4; 19; 5) where precision quantified as number of bits used to represent a number was varied using standard 64, 32 and upon occasion 16 bit formats. Reducing precision was shown to lower energy- and time-costs, often without compromising prediction quality. We note in passing that this approach can be thought of as a harbinger of *quantization* in machine learning and we will return to a discussion of this connection in Section I A. Weather prediction tends to be particularly challenging since it embodies chaotic dynamical systems. In such systems, extremely small changes to the initial conditions or inputs can cause dramatically large changes to the output. This property is appealing since the sensitivity of chaotic dynamical systems to precision in input representation can truly stress our approaches to understanding how to construct inexact surrogates. In keeping with accepted practice in the field, we will use the three tiered Lorenz96 (28) as our benchmark dynamical system throughout this paper.

We will now briefly discuss the particular forms of machine learning architectures we will be exploring in the context of the Lorenz96 system. The first architecture is based on LSTM networks (8), which have shown great results across many time series problems. The second is *echo state networks* (10) (ESNs) – both LSTMs and ESNs have been studied in the context of models for weather forecasting in general, and Lorenz96, in particular (3; 29). Lastly, we use a domain-driven regularized regression approach (D2R2), a generalization of the well-known SINDy algorithm (2).

With this as a backdrop, the main contributions of our paper are:

1. A detailed comparative analysis of the predictive effectiveness of the LSTM, ESN and D2R2 learning approaches in the context of inexactness. We will consider the impact on the quality of prediction or prediction horizon as we lower precision from 64 down to 3 bits.

2. A novel approach to characterizing the impact of inexactness through the (Fourier) *spectrum* of the predicted time-series and explaining conditions under which machine learning schemes do well through this spectral lens. *Specifically, the ability of a particular machine learning approach to do well at a particular level of inexactness or precision is characterized by its ability to capture the dominant frequency modes in the spectrum of the "ground truth" time series*, which in our case is the three-tier Lorenz 96 time series data.

All of our experiments are based on averaging over 100 randomly chosen configurations for the input conditions to the three-tier Lorenz 96 model. The accepted metric for *quality* used for such systems is the *normalized relative error* or error for short and its variance, expressed as a function of the *prediction horizon* which indicates how far into the future the surrogate is predicting. Here, error is averaged over the 100 input runs. Also, prediction horizons are expressed using *model time units* (MTUs). By convention, an average error value of 0.3 is considered to be the tolerable limit and prediction horizons exceeding this value are considered unusable. We will use the format $A(B)$ with A denoting the error and B denoting the standard deviation. Also, for purposes of this paper, *cost* is the number of bits used where all of the numbers are in a floating point format represented as $X|Y$ with X and Y respectively denoting the number of bits in the mantissa and exponent. The sign bit remains untouched. We note in passing that the IEEE standard (1) uses 5 bits for the exponent at 16 bits of precision.

In comparing our three schemes, we found it intriguing that both ESN and D2R2 schemes outperform LSTMs if we pay enough through precision. For example, with a 23 bit mantissa (single precision), the acceptable horizon value is $0.798(\pm 0.379)$ MTUs. In comparison, for ESNs, this value is 0.965 and 1.2 MTUs with just 12 bit mantissas. However, as we lower the mantissa size to 10 bits supported at single precision: LSTM networks had a predictive horizon of 0.565 whereas D2R2 methods proved more robust with a horizon of 1.1 MTUs while ESNs degraded the most down to 0.455. As a result, we focused our efforts on a more detailed analysis of the schemes that yield a performance envelope for trading quality for cost, namely ESN and D2R2.

In performing the deeper analysis, in addition to reducing precision using standard IEEE compliant formats, we also worked with reducing precision synthetically by masking out bits or equivalently reducing the dynamic range of the input parameters which are floating point numbers (Section IV B). First, we started with the standard exponent size of 5 bits which the IEEE standard supports at 16 bits of precision, and lowered it to 4 bits. Both methods failed in a noticeable manner, and thus, we only varied the mantissa in further studies. To highlight one interesting outcome, we note that in the case of ESN, at an error threshold of 0.3, ESN performance degraded to 0.245 and 0.12 MTUs with 9 and 8 bit exponents respectively. In comparison, D2R2 schemes proved more robust where the respective prediction horizons were a comparatively 1.09 (9 bits) and 0.645 MTUs (8 bits).

Why do the methods respond so differently to changes in precision? To provide intuition into the impact of inexactness on model performance, we compare the energy of the Fourier modes (14) of a model computed from the time series of its predictions, against that of the true spectrum, derived from the training data. We can explain loss in predictive power in terms of the inability to capture the dominant energy modes of the underlying system. ESN's drastic drop in predictive performance mirrors the drop in correlation between its Fourier spectrum and the ground truth spectrum below mantissas of 9 bits, while D2R2 fails to track the spectrum of the underlying three-tier Lorenz96 system below 7 bits. Our paper is the first to use Fourier modes to explain the impact of inexactness on prediction horizons of machine-learned surrogate models of chaotic dynamical systems. In addition, we offer a practical methodology for selecting machine precision thresholds to achieve specific bounds on predictive performance on any chaotic dynamical system and for any forecasting model.

A. Related work

Recently, efforts aimed at lowering cost through lowering the number of bits used through *quantization* have been explored (7; 31; 24; 26; 25; 9; 32; 21). Quantization has been used as a post-processing step to reduce memory and compute requirements for large trained neural networks with billions of parameters represented at full precision (7). It has also been integrated into the backpropagation training process, so that quantized, rather than full-precision weights are learned (9; 24; 31; 32). Quantization involves precision reduction for reducing memory footprint and number of operations, similar to inexactness. However, the original goals of inexact computing were focused on energy savings using a range of hardware architectures including customized designs for neural networks (33) as well as hardware designs to take advantage of lower precision needs (4; 17; 20). In this context, in the area of machine learning, *tensor processing units* (15) have embraced this philosophy and approach.

II. THE LORENZ96 SYSTEM

For our analysis in this paper, we select a standard benchmark for evaluating surrogate models in weather forecasting – the 3-tier Lorenz96 system, introduced in (28), a multi-scale chaotic dynamical system, which is an idealized model of processes in the atmosphere. This system is an extension of the one-dimensional atmospheric model described in (12). In Lorenz's original (12) model, there are K slowly-varying, large-scale variables $X_1, \dots, X_K$, arranged on a latitude circle, representing a macroscopic atmospheric property like temperature or pressure. Each X_k variable is coupled to J fast-varying, smaller scale variables, $Y_{j,k}$, that represent finer-grained atmospheric phenomena like gravity waves or moist convection. In the 3-tier Lorenz96 system, there is a third spatial scale introduced, in which each of the $Y_{j,k}$ variables is coupled with I even smaller scale and faster varying quantities $Z_{i,j,k}$, representing phenomena at the scale of individual clouds. The governing equations are a system of coupled non-linear ODEs involving the three variable sets, with $1 \leq i, j, k \leq K$, where $K = 8$.

$$
\begin{aligned}
\frac{dX_k}{dt} &= X_{k-1}(X_{k+1} - X_{k-2}) + F - \frac{hc}{b}\sum_j Y_{j,k} \\
\frac{dY_{j,k}}{dt} &= -cbY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{hc}{b}X_k - \frac{he}{d}\sum_i Z + i, j, k \\
\frac{dZ_{i,j,k}}{dt} &= edZ_{i-1,j,k}(Z_{i+1,j,k} - Z_{i-2,j,k}) - geZ_{i,j,k} + \frac{he}{d}Y_{j,k}
\end{aligned}
$$

In these equations, we set the forcing term F to be large ($F = 20$), which yields a very turbulent, hard to forecast dynamical system. The parameters b, c, d, e, g are set to 10, and h is set to 1; these are tuned to produce realistic spatio-temporal variability in the $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ variables. We train models using three different machine learning paradigms, to predict $\mathbf{X}_t$ from a history of past realizations of the $\mathbf{X_s}, s < t$ variables alone. We do not observe the $\mathbf{Y}$ and $\mathbf{Z}$ variables, so the prediction models have to learn latent representations of the $\mathbf{Y}$ and $\mathbf{Z}$ variables to capture the coupling expressed in the equations above.

III. MACHINE LEARNING ARCHITECTURES USED AND INEXACTNESS THROUGH PRECISION REDUCTION

A. The Long Short Term Memory (LSTM) model

LSTMs have been widely used for solving in time series prediction problems in language modeling, finance, and in speech recognition. LSTMs maintain additional structure in the form of a cell state, which is maintained and updated by a series of parameters that are learned during a training phase by backpropagation through time. LSTMs overcome the vanishing and exploding gradient problems associated with simpler recurrent neural net architectures (22). Our LSTM's architecture is derived from (30) – it has 50 hidden units and uses a time-delay embedding of the $\mathbf{X}$ vector with a lookback of three. That is, the model predicts $\mathbf{X}_t$ from knowledge of $\mathbf{X}_{t-1}, \mathbf{X}_{t-2}, \mathbf{X}_{t-3}$.

B. The Echo State Network (ESN) model

The ESN is a special case of a recurrent neural network, composed of a randomly generated mapping W_{in} projecting an input vector $\mathbf{X}$ into $\mathbb{R}^N$, and then passing that projected vector through a reservoir $\mathbf{r}$ composed of N units governed by a fixed, sparse $N \times N$ recurrence matrix A. The ESN's update and output predictions are given by

$$\mathbf{r}_{t+1} = \sigma(A\mathbf{r}_t + W_{in}\mathbf{X}_t) \in \mathbb{R}^N, \qquad \mathbf{r_0} = \mathbf{r_{t=0}} = 0 \tag{1}$$

$$\widetilde{\mathbf{r}}_t = \psi(\mathbf{r}_t) \in \mathbb{R}^F, \tag{2}$$

$$\widehat{\mathbf{X}}_{t+1} = W_{out}\widetilde{\mathbf{r}}_t \in \mathbb{R}^D, \tag{3}$$

where $\sigma(u) := \tanh(u)$ is the reservoir recurrent update function and $\widetilde{\mathbf{r}} = \psi(\mathbf{r})$ is a simple per element nonlinearity, chosen typically to increase the span of the nonlinear features $\mathbf{r}$. $W_{out} \in \mathbb{R}^{D\times F}$ is a matrix of parameters that linearly combines the elements of $\widetilde{\mathbf{r}}_n$ to generate an output prediction $\widehat{\mathbf{X}}_{t+1}$ of the state of the target dynamical system at the next time-step $\mathbf{X}_{t+1}$.

Note that the only ESN parameter being trained is W_{out}, over the training samples from dataset $\mathcal{D}_{tr} := \{(t_n, \mathbf{X}_{t_n})\}_{n=1}^{S_{tr}}$. Note also that the optimization amounts to a simple ridge (linear) regression, where the trade-off between goodness-of-fit and parsimony is controlled by the regularization strength α. The size of the reservoir N is usually taken to be a few hundred to a few thousand units, depending on the complexity of the target system. After training on some dataset D_{tr}, we often want to test the performance of our ESN on some testing set D_{te}.

In the testing task the input $\mathbf{X}_t$ is replaced with $\widehat{\mathbf{X}}_t$, the predicted output from the previous timestep. This yields an evaluation loss of the form

$$\ell_{te}(\theta_{ESN}; \mathcal{D}_{te}) := \sum_{t=1}^{S_{te}} \|\widehat{\mathbf{X}}_t - \mathbf{X}_t\|_2^2,$$

where the reservoir state r is generated according to

$$\mathbf{r}_{t+1} = \sigma(A\mathbf{r}_t + W_{in}\widehat{\mathbf{X}}_t), \qquad r_t \in \mathbb{R}^N,$$

and where the model's prediction $\widehat{x}$ is produced according to

$$\widehat{\mathbf{X}}_t = W_{out}\psi(\mathbf{r}_t).$$

Thus, the prediction task only begins with the true testing data in its first time step, and must then correctly predict future time steps using the model's own previous predictions. Hence in the prediction phase, errors can *accumulate over time.*

C. The Domain-Driven Regularized Regression (D2R2) model

For the domain-driven regularized regression (D2R2) model, we directly solve for W_{out} as

$$W_{out} = \underset{W}{\operatorname{argmin}} \|W g(\mathbf{X}_{t-1}) - \mathbf{X}_t\| + \alpha \|W\|_2^2$$

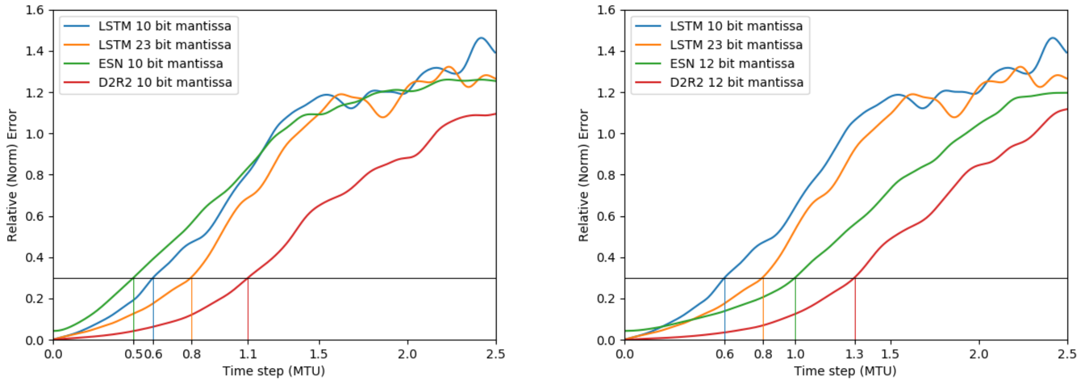
where g is a function that that performs basis function expansion of $\mathbf{X}_{t-1}$. For polynomial regression, $g := g_m$ returns all polynomial terms up to a specified order m e.g. $g_2(\{x, y\}) = \{1, x, y, x^2, y^2, xy\}$. The primary difference between the ESN and the D2R2 model is that we supply a basis function set g to the D2R2 model, while the ESN model uses fixed projections to construct a random basis, and tunes W_{out} so that a linear combination of those random bases minimize a regularized L2 prediction error function.

IV. PRECISION VARIATION MAIN RESULTS

For the purposes of this paper, the Lorenz96 data set has been generated using the Runge–Kutta integration method with a step size of 0.005 time units. The size of the data-set is one million integration timesteps, and only the eight X_k variables being used for model training and evaluation (e.g. only *partial information* is available). Prediction error is calculated as a normalized mean-squared error between the vector $\mathbf{X}$ at a given time t of a prediction time-step and the value of $\mathbf{X}$ in the original Lorenz96 data-set at the same time t. The primary measure for model evaluation is prediction horizon measured in *model time unit* (MTU), which represents the time from the start of the prediction until the prediction error crosses a threshold at which point, the prediction is regarded as unusable. Since the time step for Runge-Kutta integration is 0.005 time units, the prediction horizon is defined as 0.005 times the number of time units till the absolute value of the prediction error exceeds the widely accepted threshold of 0.3.

Let us first consider the impact of inexactness in exponents on performance. Our experiments indicated that for Lorenz 96, there is a precipitous drop in prediction quality in going down from 5 to 4 bits. In particular, for the two systems which are the focus of our deeper study, in the case of ESN, the prediction horizon dropped from a value of 0.965 MTUs to 0.43 MTUs or a drop of close to 50%. Similarly, for D2R2, the prediction horizon plummeted from a robust 1.3 MTUs with 5 bits to hopelessly low value of 0.025 MTUs with a 4 bit exponent! Consequently, in the sequel, we retained 5 bit exponent representations while studying the impact of varying the mantissa.

The relative performance of all three of our machine learning schemes are shown in Figure 1, In Figure 1 (a), we consider the cases of LSTM networks with 23 bit and 10 bit mantissas, respectively supported by double and single precision floating point formats. We note that even with double precision and therefore with a substantially large budget of 23 bit mantissas, LSTM networks cannot predict past 0.78 MTUs. In contrast, with 12 bit mantissa values, both ESN and D2R2 outperform as shown there. Notably, D2R2 has a prediction horizon of 1.3 MTUs, around a 66% improvement with half as many bits in the mantissa!



(a)

(b)

FIG. 1: The error increase rate during the prediction, averaged over 100 randomly chose initial conditions a.) with 12 and 23 bit mantissas for LSTM and 12 bit mantissas for ESN and D2R2 and b.) lowering mantissas to 10 bits for ESN and D2R2.

Both ESN and D2R2 also demonstrate very interesting behavior when the mantissa size is reduced. By going from 12 to 10 bit mantissas (Figure 1), the prediction horizon of ESN systems drops by 55% and has a shorter prediction horizon of 0.455 compared to LSTMs with the same budget for their mantissas. In contrast, for the same reduction in (mantissa) bit budgets for D2R2 systems, their ability to predict decreases from 1.3 to 1.1 MTUs which is a mere 15% degradation. With this as a backdrop, we proceeded to probe the relative behaviors of ESN and D2R2 systems more deeply.

A. Inexactness through non-standard precision levels

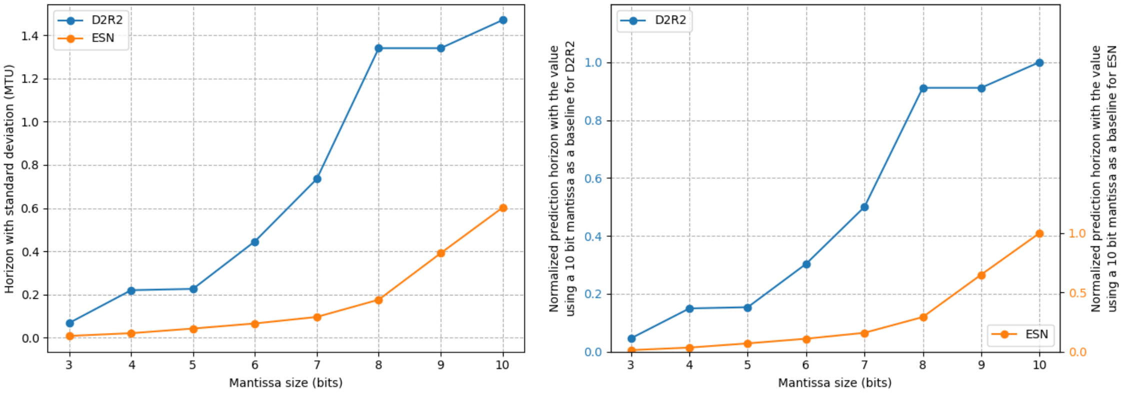
We will now consider the impact of varying precision at bit budgets other than those supported by IEEE standards to explore the space of possibilities.

B. Methodology for precision variation

Standard tools and libraries provided floating point precision settings for standardized 64, 32 and 16-bit floating numbers. In order to emulate more fine grained floating point formats, we used a custom method which involves dividing the number into the mantissa and the exponent. Finding the closest representation of the mantissa using the desired number of bits is then calculated by simple iteration over the fractions of powers of 2 and reducing the number suitably; essentially, we simulate truncation. As for the exponent, we conclude that if the desired value is able to be represented in the dynamic range or whether the result is going to be rounded to a value multiple order(s) of magnitude lower than the desired value. Exponents play a powerful role in that the effects of reducing their size does not having significant effect on the prediction until it is no longer able to represent the smallest weights inside the trained models, at which point the model loses all it's predictive power.

1. Results and trends

Our results are summarized in Figure 2. In Figure 2 (a), we show the error growth as the prediction horizon increases with the number of mantissa bits. It is instructive to note that D2R2 consistently outperforms ESN networks, for example by close to a factor of 4 with 7 bit mantissas. It is also interesting to note that in the case of D2R2, the steepest gain is in going from a 7 bit to a 8 bit mantissa which represents a gain of close to 85% from 0.39 to 0.645. In contrast, ESN networks show a more gradual loss in their prediction horizons.



(a) (b)

FIG. 2: Considering non-standard mantissa values from 10 to 3 bits a.) comparing the prediction horizon as a function of mantissa size and b.) relative decrease normalized using a 10 bit mantissa value as the baseline with the D2R2 scale along the ordinate axis on the left and the ESN on the ordinate on the right.

These observations come into stark focus if we consider Figure 2. In this figure, we use the prediction horizon with 10 bit mantissas as a baseline and calculate the relative quality (decrease) as we increase the number of bits in the mantissa. As shown there for the ESN case, the prediction horizon decreases by more than 50% in going from 10 to 8 bits or a two bit decrease whereas for the same change, the loss in prediction horizon is around 12.5% for D2R2 systems.

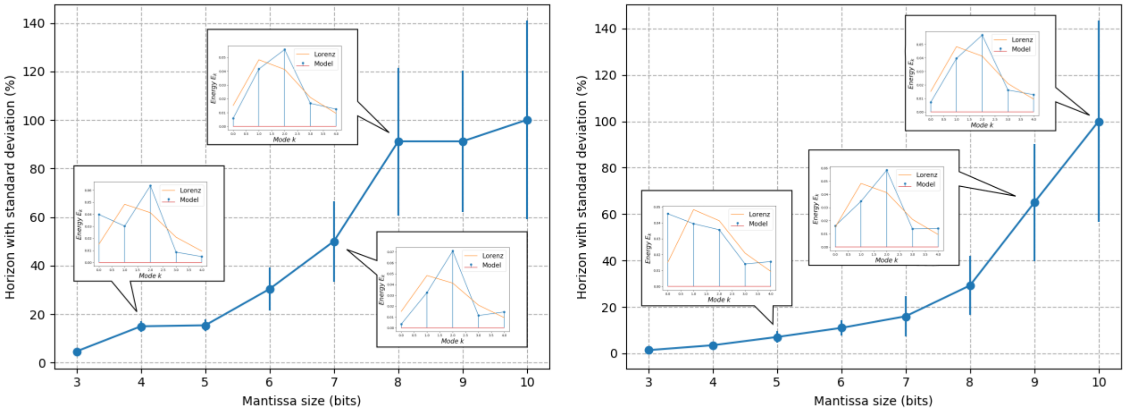
V. USING FOURIER MODES TO EXPLAIN PERFORMANCE VARIATION WITH REDUCTION IN PRECISION

The 3-tier Lorenz 96 system (28) used here has a very high dimensional phase space and an unknown number of attractors. For such systems, short-term prediction (weather forecasting) as well as characterization of long-term statistics (climate forecasting) can be very challenging. While we know the exact structure and parameter values of the coupled non-linear ODEs for this Lorenz96 system, it is difficult to work with them analytically to understand system behavior. Instead, we use a purely data-driven approach for reconstructing a frequency domain models from the ground-truth time series of the 3-tier Lorenz96 state vectors, as well as the predicted time series produced by the ESN and the D2R2 models. Essentially, we map each state vector in a time series (ground-truth or predicted) into the frequency domain using the discrete Fourier transform (14; 30).

First, we shift and scale the $\mathbf{X}$ vectors by their discrete or Dirichlet energy, so that the scaled states $\widetilde{\mathbf{X}}$ have zero mean and unit energy. The transformation equations are

$$\widetilde{X_k} = \frac{X_k - \overline{X}}{\sqrt{E_p}}, \widetilde{dt} = \sqrt{E_p}dt, E_p = \frac{1}{2T}\sum_{k=1}^{K-1}\int_{T_0}^{T_0+T}\left(X_k - \overline{X}\right)^2 dt$$

where the Dirichlet energy of a vector $\mathbf{X}$ is $\frac{1}{2}\sum_{k=0}^{K-1} X_k^2$, E_p is the average energy fluctuation in a time interval T about time step T_0, $\overline{X}$ is the mean over all the X_k at a given time step. This



(a)

(b)

FIG. 3: Varying mantissa precision on Lorenz96, with Fourier modes of the prediction and the ground truth sequences, for interesting mantissa values, for the D2R2 model (left) and the ESN model (right). The y-axis represents predictive accuracy as a percentage of the length of the prediction horizon at mantissa size of 10.

transformation ensures that the scaled energy $\frac{1}{2}\sum_{k=0}^{K-1}\widetilde{X_k}^2 = 1$ and that the scaled variables have zero mean $\widetilde{\overline{\mathbf{X}}} = \frac{1}{K}\sum_{k=0}^{K-1}\widetilde{X_k} = 0$.

Then, we apply the discrete Fourier transform to the energy-standardized states $\widetilde{\mathbf{X}}$. The Fourier coefficients $F_j \in C$ and the inverse Fourier transform to recover the original vector are

$$F_j = \frac{1}{K}\sum_{k=0}^{K-1}\widetilde{X_k}e^{-2\pi ijk/K}, \widetilde{X_k} = \sum_{j=0}^{J-1}F_j e^{2\pi ijk/K}$$

The F_j's constitute the Fourier spectrum of the energy-standardized state vector. The spectrum is symmetric and we can uniquely characterize a state by $K/2+1$ coefficients, which translates to five coefficients in our system, since $K = 8$. These coefficients are called the Fourier modes of a state $\widetilde{\mathbf{X}}$. The Fourier energy of each mode is defined as: $E_j = Var(F_j) = E[(F_j(\widetilde{t}) - \overline{F_j})(F_j(\widetilde{t}) - \overline{F_j})^T$. For a time series of scaled states, the energy modes are averaged over the entire sequence. The energy spectrum of the predicted sequence and the corresponding ground-truth series are plotted in Figure 3 as call-outs for specific values of mantissa precision. We can see that the energy spectrum of the predicted time series is distorted relative to the ground-truth spectrum at a mantissa size of 8 bits for the D2R2 model and at 9 bits for the ESN model. The figure also shows the reduction in prediction horizon in both models – D2R2 retains 80% of the prediction horizon at a mantissa size of 8 bits, while ESN drops to about 60% at a mantissa size of 9 bits. We see an explanation for the degradation of predictive performance in the models as a consequence of precision reduction – the predicted models become less and less effective in capturing the true modes of the 3-tier Lorenz system. Another view of the ability to capture the energy modes is in Figure 4. Here we show the correspondence between the energy spectrum of the ground-truth time series and the predicted series using Pearson's correlation coefficients. Note that while D2R2 retains a fairly high correlation with the true spectrum upto a mantissa size of 9 bits, ESN fails to capture the modes of the true spectrum below 11 bits.

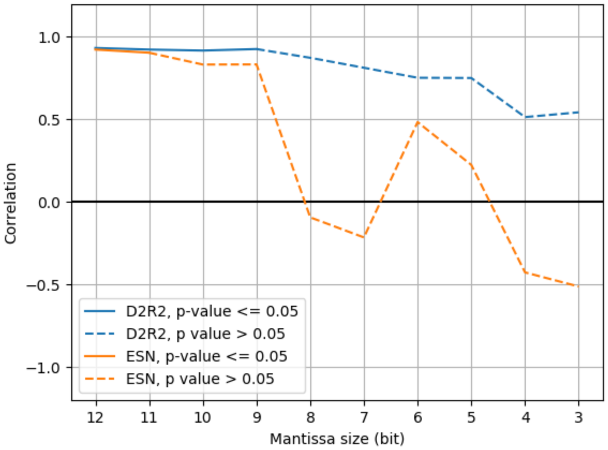


FIG. 4: Degradation of Pearson's correlation coefficients between the prediction and the ground truth energy spectra, with decrease in mantissa size.

VI. REMARKS, CONCLUSION AND NEXT STEPS

In this paper, we proposed an approach for systematizing the use of inexactness to enable scaling of machine learning surrogates for predictive modeling, with an emphasis on the domain of applied mathematics. In this domain, models are canonically represented as ODEs and PDEs whose use is ubiquitous in the physical sciences and engineering disciplines. To this end, we chose the three-tier Lorenz 96 system which is viewed as being representative of phenomena common to weather prediction and climate modeling. This system is very popular since it serves as a good benchmark for *multi-scale* time series models, is amenable to incorporating constraints from physics, and can produce chaotic behavior. To contribute to the systematization of understanding and using inexactness to enable scaling of machine learned surrogates, a central contribution of this paper is the introduction of spectroscopy, wherein the Fourier spectrum is used to represent and evaluate trade-offs in a design space representing the cost of the surrogate and its prediction quality. While our current framework for spectroscopy is based on experimental methods, our intention is to, by leveraging the natural mathematical structure inherent in spectral methods, develop a mathematically robust and useful framework to aid in the understanding of the interplay between inexactness and machine learning. We hope this will serve as a basis for developing a principled methodology for the design of scalable machine learned surrogates.

BROADER IMPACT

This work aims to improve the efficiency with which known machine learning schemes can be used for forecasting multi-scale dynamical systems, such as weather, by replacing expensive numerical integration of large systems of coupled ODEs and PDEs representing domain physics. Thus, we do not introduce new learning schemes or biases. Also, in our case, the ground truth is a mathematically and physically well defined quantity based on measurements of variables. Thus, in the weather prediction domain for example, it involves such parameters as temperature, wind speed, cloud reflectivity and so on. We therefore expect our use cases to be based on such objective physical measurements. In forecasting applications, the metric used is the model's prediction horizon which is compared to historical data, which is also measured. If forecast quality of the model is poor, then machine learned surrogates described here will fail to serve as valid replacements for the expensive, traditional physics based models represented as ODEs and PDEs. We expect any such adoption to follow extremely rigorous and substantial testing. We are unaware of any other positive or negative impact possibilities for our proposed work.

REFERENCES

[1] Standard for Floating-Point Arithmetic. Standard, The Institute of Electrical and Electronics Engineers, June 2019.

[2] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[3] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian. Data-driven prediction of a multi-scale lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ann, and rnn-lstm. *arXiv preprint arXiv:1906.08829*, December 2019.

[4] P. D. Düben, J. Joven, A. Lingamneni, H. McNamara, G. De Micheli, K. V. Palem, and T. Palmer. On the use of inexact, pruned hardware in atmospheric modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2018):20130276, 2014.

[5] P. D. Düben and T. Palmer. Benchmark tests for numerical weather forecasts on inexact hardware. *Monthly Weather Review*, 142(10):3809–3829, 2014.

[6] P. D. Dueben and P. Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018.

[7] H. M. Han, Song and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[9] e. a. Jacob, Benoit. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[10] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[11] A. Lingamneni, C. Enz, K. Palem, and C. Piguet. Synthesizing parsimonious inexact circuits through probabilistic design techniques. *ACM Transactions on Embedded Computing Systems (TECS)*, 12(2s):1–26, 2013.

[12] E. Lorenz. Predictability of weather and climate, 2006.

[13] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[14] A. Majda, R. V. Abramov, and M. J. Grote. *Information theory and stochastics for multiscale nonlinear systems*, volume 25. American Mathematical Soc., 2005.

[15] e. a. Norman P. Jouppi. In-datacenter performance analysis of a tensor processing unit. *SCA '17: Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017.

[16] K. V. Palem. Computational proof as experiment: Probabilistic algorithms from a thermodynamic perspective. In *Verification: Theory and Practice*, pages 524–547. Springer, 2003.

[17] K. V. Palem. Energy aware computing through probabilistic switching: A study of limits. *IEEE Transactions on Computers*, 54(9):1123–1137, 2005.

[18] K. V. Palem. Inexactness and a future of computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2018):20130281, 2014.

[19] T. Palmer, P. Düben, and H. McNamara. Stochastic modelling and energy-efficient computing for weather and climate prediction, 2014.

[20] T. N. Palmer. More reliable forecasts with less precise computations: a fast-track route to cloud-resolved weather and climate simulators? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2018):20130391, 2014.

[21] J. A. Park, Eunhyeok and S. Yoo. Weighted-entropy-based quantization for deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[23] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

[24] R. G. B. G. H. J. Pierre Stock, Armand Joulin. And the bit goes down: Revisiting the quantization of neural networks. *CoRR abs/1907.05686*, 2019.

[25] M. E. M. R. T. T. Rahaf Aljundi, Francesca Babiloni. Memory aware synapses: Learning what (not) to forget. *CoRR abs/1711.09601*, 2017.

[26] S. Seo and J. Kim. Efficient weights quantization of convolutional neural networks using kernel density estimation based non-uniform quantizer. *Applied Sciences 9.12 (2019)*, 2019.

[27] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[28] T. Thornes, P. Duben, and T. Palmer. On the use of scale-dependent precision in earth system modelling. *Quarterly Journal of the Royal Meteorological Society*, 143:897–908, 2017.

[29] P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 2020.

[30] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[31] e. a. Zhe, Wang. Optimizing the bit allocation for compression of weights and activations of deep neural networks. *2019 IEEE International Conference on Image Processing (ICIP). IEEE*, 2019.

[32] e. a. Zhou, Yiren. Adaptive quantization for deep neural network. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] Y. C. K. P. O. T. Zidong Du, Avinash Lingamneni and C. Wu. Leveraging the error resilience of machine-learning applications for designing highly energy efficient accelerators. *19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore*, 2014.