Learning physics based models of Lithium-ion Batteries

Maricela Best Mckay * **Brian Wetton** *
R. Bhushan Gopaluni **

* Department of Mathematics, University of British Columbia (e-mail: maricela@math.ubc.ca).
** Department of Chemical and Biological Engineering, University of British Columbia.

Abstract: Lithium-ion (Li-ion) batteries are increasingly pervasive and important in daily life. We present a surrogate modeling approach that uses synthetic data generated by an electrochemical model to approximate Li-ion battery dynamics using a Deep Neural Network. Elechtrochemical models are needed to describe high current operation but are computationally costly. As an initial study, we prototype our approach for the Single Particle Model. We use a battery use-cycle and observations of concentrations and voltages to predict future battery behavior. Given only the use cycle and knowledge that the battery is fully charged, the surrogate model can accurately forecast observations of Li-ion concentrations and voltages for the entire use cycle, as well as give a window of time for battery depletion.

1. INTRODUCTION

Lithium-ion (Li-ion) batteries are increasingly important as their use permeates daily life. Due to their high energy densities, long lifetimes, and low cost, they are employed across a wide array of applications. With recent moves by governments and private citizens to curb green-house gas emissions through the use of electric vehicles and energy storage, the need for Li-ion batteries will only increase (Dunn et al., 2011).

Many modeling techniques exist to study Li-ion batteries. Equivalent circuit models are commonly used to study and optimize battery performance, but they are not accurate under a wide range of C-rates (the ratio of the discharge current to the current draw which delivers the rated capacity in one hour). Physics based models of Li-ion batteries are more accurate but are too complex for online optimization and control. To address this, various model reduction or reformulation strategies have been proposed (Forman et al., 2011; Speltino et al., 2009; Dao et al., 2012; Zou et al., 2016). Reduced order models are useful and effective tools, but involve reductions in model fidelity, or are suited only to certain use cases (Mishra et al., 2016).

Other approaches for studying Li-ion batteries involve data-fitting techniques, or recently, data-driven techniques that make use of the growing wealth of machine learning advances and applications. Data-driven approaches show great promise but rely on good quality data, which can be costly and difficult to acquire and is limited by experimental conditions or what is practical/possible to measure. Additionally, machine learning techniques often lack the interpretability or rigor of first principles models.

This has led to recent efforts to combine machine learning techniques with mathematical modeling. These efforts include using neural networks as approximators in numerical schemes for PDEs (Bar-Sinai et al., 2019; Hamilton et al., 2019), the development of physics-informed neural networks, which utilize physical information in the construction of neural networks (Raissi et al., 2019), and the use of neural networks as surrogate models (Zhu and Zabaras, 2018; Tripathy and Bilionis, 2018).

Motivated by these ideas we use machine learning techniques to develop a surrogate model to replace a high-fidelity electrochemical model (ECM). Rather than using experimental data to create a surrogate model, our approach involves approximating the ECM with a Deep Neural Network (NN). We use the the ECM to generate a large data-set and train an NN to approximate the battery dynamics via the synthetic data set.

While still a type of data-driven approach, the use of synthetic data provides several benefits. Because the data is synthetic it is not limited to measurable quantities, any quantity that can be modeled can be approximated. Parameters used as inputs in the ECM can be changed, as well to create data representative of a variety of chemistries. When using data-driven approaches on experimental data, it may not always be clear how to describe the underlying relationships being captured. With synthetic data generated by an ECM, the DNN is in essence an approximation of the underlying mathematical structure of the model, and relationships are more readily apparent.

In this paper, we present an initial study prototyping the surrogate modeling approach outlined above. We are

able to build a highly accurate surrogate model towards the goal of replacing an ECM. We use a battery use-cycle and observations of concentrations and voltages to predict future battery behavior. Given only the use cycle and knowledge that the battery is fully charged, the surrogate model can accurately forecast observations of Li-ion concentrations and voltages for the entire use cycle, as well as give a window of time for battery depletion.

2. SYNTHETIC DATA GENERATION

Our goal is to train a NN to approximate battery dynamics for use as a predictive surrogate model to replace a physics-based model of a Li-ion battery. For this initial study we elect to use the Single Particle Model (SPM) to simulate the battery dynamics. The SPM is a reduced order model for the Doyle-Fuller-Newman Model (Doyle et al., 1993). The SPM model is derived under the assumption that the transport limitations due to scaled intercalated Lithium transport in the particles dominate over the limitations of ion transport in the electrolyte. Under this assumption, all particles can be assumed to see the same electrolyte conditions and so it is sufficient to consider only one representative particle in each electrode.

As a brief description, the SPM keeps track of the inter-calated lithium concentration $c_s(r, t)$ in a representative spherical particle in each electrode. Here, r is the depth into the particle. Also considered are simple models of electrolyte and solid conductivity. For this study, we consider the isothermal case. Some representative $c_s(r, t)$ plots at fixed t are shown in Fig.6. The SPM dynamics add important detail to $c_s(r, t)$, that is not captured by simpler models (which for the most part consider $c_s(r, t)$ constant in r, or implicit inequivalent circuit models.)

We generate data for both training and testing the NN model with Python Battery Mathematical Modelling (Py-BaMM), an open source Python library for mathematical battery modeling (Sulzer et al., 2020). PyBaMM was selected for its robustness, ease of use, and because of its functionality in both specifying input parameters and use-cycles for battery simulations. A full description and derivation of the SPM as implemented in the PyBaMM library can be found in (Marquis et al., 2019).

2.1 Synthetic use cycles

Many applications of interest for Li-ion batteries involve high energy demands and fast C-rates. For example, Electric Vehicles (EVs) are an application area of great interest where high C-rates arise. It is also in the parameter space of high C-rates that a more descriptive electrochemical model is needed as opposed to an equivalent circuit model. Motivated by these considerations, we generate synthetic use cycles that capture rapid fluctuation conditions and contain high-discharge rates.

We select a 100-second window for prediction/sampling. Starting from a fully charged battery with zero current being drawn and uniform Li-ion concentration in particles and electrolyte, the use cycle is given by a piece-wise linear continuous current function $I(t)$ (see fig.1).

Each line segment in $I(t)$ is defined on a 100 second time window and is described by its two end points (t_j, I_j) and
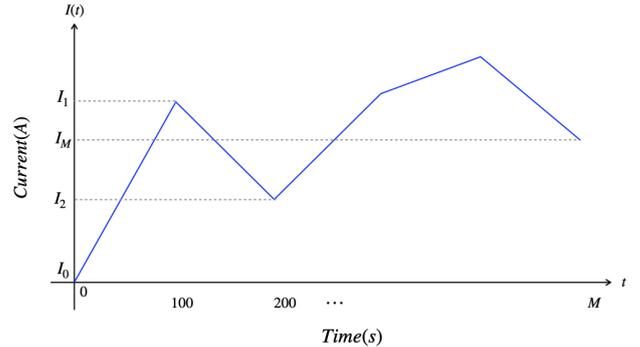

Fig. 1. A diagram illustrating the piece-wise linear use cycle $I(t)$. Each $I(t_j) = I_j$ for $j = 0, 100, \ldots, M \in 100\mathbb{N}$ is drawn from the triangle distribution with the density function shown in Fig. 2.

(t_{j+1}, I_{j+1}), where $j = 0, 100, 200 \ldots, M \in 100\mathbb{N}$. Each $I_j \in \{I_j | j > 0\}$ is drawn from a triangle distribution $\mathcal{T}$. The probability density function for $\mathcal{T}$ has a minimum value of zero, a peak at $1/I_{\max}$, and a maximum value of $I_{\max}$, where $I_{\max}$ is the maximum current that can be drawn over a 100 seconds interval from a fully charged cell (see fig.2). The simulated data contains many different trajectories from full state of charge to the minimum voltage cut-off.
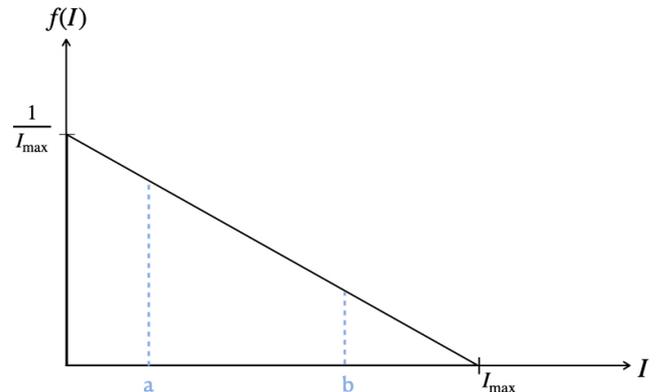

Fig. 2. The probability density function for $\mathcal{T}$. The probability that any sampled I_j will fall in the interval $[a, b]$ is given by $\mathcal{P}(a \leq I_j \leq b) = \int_b^a f(I) dI$. The chance of sampling an I_j close to $I_{\max}$ is quite low, so the likelihood of depleting the battery quickly is also low.

For this choice of $I_{\max}$, the voltage curves depend strongly on changing Li-ion concentration levels. This means that the dynamics of the system cannot be modeled by an equivalent circuit model and a more descriptive model like the SPM is needed (see Fig.6).

3. MACHINE LEARNING FRAMEWORK

We construct and train a feedforward Deep Neural Network (DNN) to approximate the battery dynamics. More specifically, the DNN is a non-linear map between inputs and outputs that approximates the battery dynamics using observations from the simulated data.

Because we would like to capture non-linear dynamics a DNN is a natural choice. Additionally, DNNs are excellent function approximators that have been applied across many disciplines and applications with highly accurate results (Cybenko, 1989).While training a DNN can be computationally costly, offline predictions after training are inexpensive enough to be done on a low-power CPU.

3.1 Approximating Battery Dynamics with a DNN

The NN model is trained to predict future battery behavior over a 100-second horizon, given information on the input current, the terminal voltage, and concentration of lithium in each electrode at the start of the prediction window.

For each 100-second time window $T_j = [t_j, t_{j+1}]$ in the data set, the DNN is trained on the current at time t_j, and time t_{j+1}, the terminal voltage at time t_j, and the concentrations profiles $c_{s_n}(t_j)$ and $c_{s_p}(t_j)$ in the negative and positive electrode. Note that these concentration profiles are not measurable quantities. The only concrete information known is that initially they are constant in r at a fully charged state. The DNN is trained to predict terminal voltage, $V(t)$, at time t_{j+1}, the concentration profiles in the negative and positive electrodes at time t_{j+1}, and whether the battery failed during time window T_j. If the battery failed in the time window being considered, the last values before failure are considered instead.

Concretely, letting t_N denote the last second of battery operation before failure, the inputs to the DNN are

$$\mathbf{x} = [I(t_j), I(t_{j+1}), V(t_j), c_{s_n}(t_j) \, c_{s_n}(t_j)],$$

where $j + 1 \in \{100, \ldots, M\}$ or $j + 1 = N$, and the targets are

$$\mathbf{y} = \begin{cases} \left[1, \, V(t_N), \, c_{s_n}(t_N), \, c_{s_p}(t_N)\right], \text{ if failure} \\ \left[0, \, V(t_{j+1}), \, c_{s_n}(t_j + 1), \, c_{s_p}(t_j + 1)\right], \text{ o/w} \end{cases} \quad . \text{(1)}$$

The desired outputs for the DNN are of mixed type, because the goal is not only to categorize battery failure, but also to predict future observations based on current observations. The former calls for logistic regression, while the latter is trained for prediction accuracy via a Mean Squared Error (MSE) loss. The overall loss function for the DNN is the sum of binary-cross-entropy loss and MSE.

The DNN has three dense hidden layers, each with a Rectified Linear Unit (ReLU) activation function. There are two output layers; one performs logistic regression and is thus sigmoid activated, while the other predicts future behavior and has a ReLU activation (see fig.3 for a schematic of the NN model architecture). The Network is trained with Keras, using ADAM as the optimizer with a learning rate of 0.001. The Network is trained with $24,822$ total parameters.

4. MODEL PERFORMANCE

In this section we describe how the NN model's performance was assessed and then describe its performance.

4.1 Training, and testing data

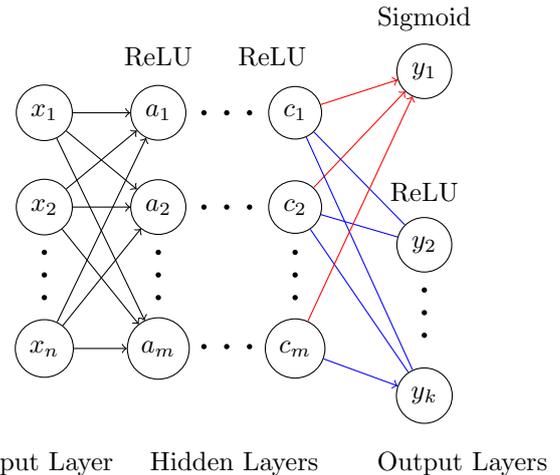We generate a data set of 15,000 use cycles using the procedure outlined in sec. 2.1. We select $M = 15$ so that

Fig. 3. Schematic of the DNN Model architecture.

each use cycle is $1,500$ seconds long. Because each use cycle $I(t)$ in the data set is generated by sampling from the triangle distribution $\mathcal{T}$, for almost all of the use cycles in the data set, the battery does not make it all the way to $15,000$ seconds, but instead is depleted before reaching I_M. To be more precise, there are 165 use cycles in the data set that do not deplete the battery. The distribution of how long the battery lasted before depletion, which we will refer to as run-time, is shown in Fig. 4 for all the simulations in the data set.
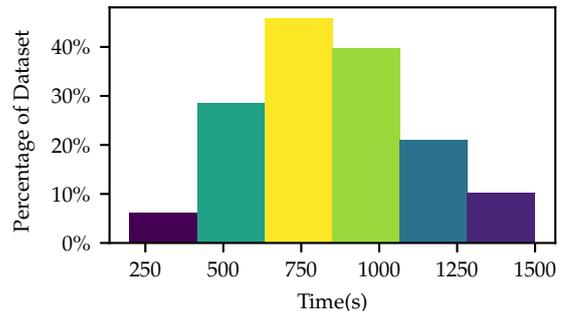

Fig. 4. A histogram of battery run-times for all the battery simulations in the data set.

The simulation data for each battery run is split into 100 second blocks, to generate a set of samples for training and testing the NN model. For a given block B_j in a run, the start and end values coincide with the end points I_j and I_{j+1} of the line segment in the use cycle that was used to generate the simulation data for that block, i.e. the simulation data is split according to the pieces in the piecewise-linear use cycle. If the battery failed in the time window being considered, B_j will only be $(N (\bmod 100) < 100$ long. From each block of data, the first and last values are taken to generate a set of samples for training the NN model.

The data set of $15,000$ runs generates $126,355$ total samples. We split the samples data set into two pieces for cross-validation purposes. The Training data set consists of 65% of the samples, $82,131$ samples in total. The Testing data set, which is held back for exclusive use testing the accuracy of our NN model, is made up of the other

35% of the total samples, $44,223$ total samples. Model performance on these data-sets is presented in the next section.

4.2 Metrics

To evaluate the predictive performance of the model we use several metrics, Mean Squared Error (MSE), Mean absolute percentage error (MAPE) and categorical accuracy.

Mean Squared Error is given by

$$\frac{1}{m} \sum_{i=1}^{m} (y_i^* - y_i)^2 \tag{2}$$

where y_i^* is a predicted value generated by the DNN, y_i is the corresponding target value in the data set, and m is the total number of data samples. (MAPE) is defined as

$$\frac{1}{m} \sum_{i=1}^{m} \frac{|y_i^* - y_i|}{y_i} \times 100. \tag{3}$$

Categorical accuracy measures how often a predicted category is the correct category for a given data sample and is simply the number of correctly categorized samples, divided by the total number of samples. The first two metrics are used to asses how well the model is able to predict $V(t)$ and $c_{s_n}(t)$ at time t_{j+1} or t_N, while the categorical accuracy metric, assesses how well the model is able to predict battery depletion.

4.3 Model Accuracy

Model performance for the metrics outlined above is presented in Table 1. We see that the model is able to predict battery failure with close to 100% accuracy for both the training data set and the testing data set. The Mean absolute percentage error for predicted future concentrations and voltages is close to one percent for both the testing and training data. In other words, the model shows very good predictive ability on the testing data set and is able to capture the battery dynamics modeled by the SPM with high accuracy.

Table 1. The Predictive accuracy of the model as measured by the Loss functions and Metrics.

Data	Total Loss	Binary Cross Entropy	MSE	MAPE	Categorical accuracy
Training	0.0070	0.0069	7.6e-05	0.8843	0.999
Testing	0.0072	0.0072	7.18e-05	1.2782	0.998

4.4 Li-ion Concentrations

An advantage of our surrogate modeling approach is that our NN model has the ability to estimate information that cannot be directly measured. One such quantity is the concentration of Li-ions in the negative and positive electrodes. These concentrations hold a wealth of useful information. For example the state of charge (SOC) of the battery can be directly calculated from Li-ion concentrations by integration. In addition, the surface concentration plays a key role in determining cell voltage.

We would like to get a better sense of the model's predictive accuracy for Li-ion concentrations in the negative and positive electrodes. Towards this end, we generate a new smaller testing data set, T_2, created via the same procedures as the original training and testing data sets, but independent of and after model training and validation. This data set is comprised of $1,000$ runs, which when split into blocks come out to $9,319$ samples.

While it is encouraging that the MSE for predicted concentrations is very low, this metric doesn't provide any insight into how well the NN model approximates the shape of the Li-ion distribution in the particles. To get a sense of this we present a heat map of the prediction error as a percentage of the true concentration profile for all samples in T_2 in Fig. 5.
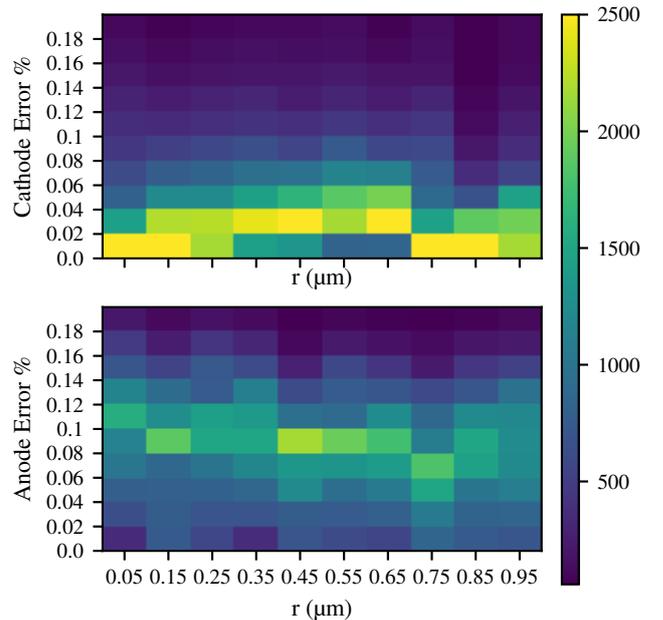

Fig. 5. Heat map of errors in particle concentration predictions as a percentage of the true value, across all samples, at each of the measured distances (r) along the particle radius. The Heat map on the left corresponds to the negative particle, and the one on the right to the positive particle.

We see that the prediction error is low across the entire particle for the positive and negative particles. The Error in the positive electrode is higher overall than the error in the negative electrode. However, both Heat Maps show that error is for the most part evenly distributed across the particle. This demonstrates that predictions provide not just a good approximation on average, but overall shape fidelity.

To examine the variance in prediction accuracy, we select three samples from T_2. These are chosen according to the following procedure; Given the distribution of MSE when comparing predicted concentrations to true ones, the samples with error closest to the 25th, 50th, and 27th percentile values of the distribution are chosen. Note that these errors will occur at different samples for the positive and negative electrodes, so all together there are six samples. Concentration profiles for these samples are

shown in Fig.6, along with prediction error as a percentage of the true values. This error is scaled by a factor of three to make it more visible on the graphs. The left column shows the samples chosen from the negative MSE distribution, while the right column shows the samples chosen from the positive distribution.
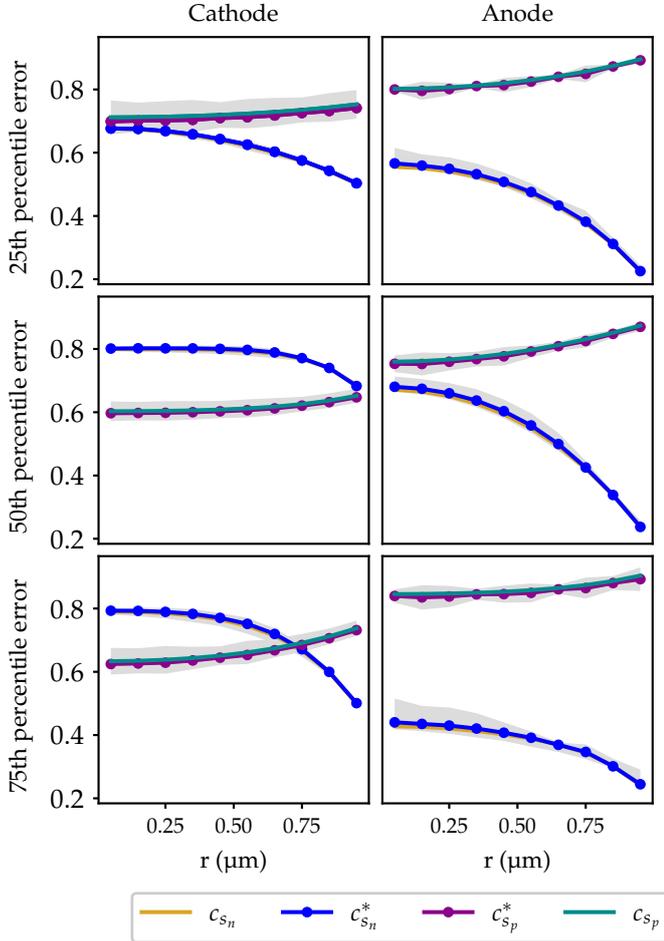

Fig. 6. Negative and Positive particle concentration profiles for samples selected from the upper, middle, and lower quartile of the distributions of MSE for predicted vs true values. The left column of figures shows the samples chosen from the MSE distribution for the Cathode. The right column shows the samples chosen from the MSE distribution for the Anode. Percentage error is shown in grey around the concentration profiles. These errors are scaled up by a factor of three to make them more visible on the graph. True c_{s_n} and c_{s_p} are shown in yellow and teal respectively. Predicted profiles denoted by $c_{s_n}^*$ and $c_{s_p}^*$ are plotted with dot markers and shown in blue and purple respectively.

The six samples in Fig.6 give a representative snapshot of the range of model performance. Shape fidelity and overall accuracy are high for all samples, even the samples corresponding to the 75th percentile MSEs. The topographical shapes of the errors observed in fig.5 align with the shape of the errors along the profiles. For example, errors in the positive particle are lower at the center and surface of the particle and higher than those in the negative particle. These observations agree with error patterns in Fig. 5. This provides further evidence that our six samples are representative of the overall data set and NN model performance.

5. MULTI-STEP PREDICTION

The previous section demonstrated that the NN model is able to accurately forecast 100 seconds into the future. A natural next step is to gauge if error accumulates for a longer forecast horizon. To test this, we recursively feed NN model forecasts, back into the model over k steps.

Recalling that the trained NN model is a nonlinear map between inputs $\mathbf{x}$ and predicted outputs, we adopt the notation $\mathbf{x} \mapsto F(\mathbf{x}) = \mathbf{y}^* \approx \mathbf{y}$ to describe this relationship. Given a synthetic use cycle $I(t)$ from data set T_2, the input data for $t = 0$, $\mathbf{x}_0 = [I(t_0), I(t_1), V(t_0), c_{s_n}(t_0)\, c_{s_n}(t_0)]$, is run through the NN model to get the one-step forecast $\mathbf{y}_1^* = \left[V^*(t_1), c_{s_n}^*(t_1), c_{s_p}^*(t_1) \right] = F(\mathbf{x}_0)$. Here the superscript $*$ is used to indicate that a quantity is an approximation generated by the Neural network. The two-step forecast is $F(\mathbf{x}_1^*) = F\left([I(t_1), I(t_2), \mathbf{y}_1^*]\right) = \mathbf{y}_2^*$. Repeating iteratively, the k-step forecast is

$$\mathbf{y}_k^* = F\left(\left[I(t_k),\, I(t_{k+1}),\, \mathbf{y}_{k-1}^* \right]\right) = F(\mathbf{x}_{k-1}^*). \quad (4)$$

It is important to note that for any use cycle $I(t)$, we start with $I(0) = 0$, and the battery fully charged. This means that $c_{s_n}(0)$ and $c_{s_p}(0)$ have the same constant profiles for any use-cycles, and $V(0)$ also has the same value for all use cycles. So, the only information the NN model has that distinguishes any trajectory that starts at $t = 0$ from another is the future current $I(t_1)$.

We asses model performance for k-forecasting over the testing data set T_2 from the previous section. For each run of the $1,00$ runs in T_2, we perform k iterative forecasting steps. We stop when the NN model predicts battery depletion after the kth forecast.

Table 2. MSE for k-step forecasts vs true observations

k	V	c_{s_n}	c_{s_p}
1	7.24E-04	9.75E-06	3.76E-05
2	3.18E-04	2.48E-05	9.57E-05
3	2.35E-04	4.39E-05	2.05E-04
4	2.52E-04	8.37E-05	4.23E-04
5	2.20E-04	1.26E-04	7.20E-04
6	2.58E-04	1.60E-04	1.08E-03
7	3.54E-04	1.90E-04	1.43E-03
8	6.20E-04	2.11E-04	1.82E-03
9	8.40E-04	2.22E-04	2.13E-03
10	1.73E-03	2.17E-04	2.46E-03
11	1.30E-03	2.36E-04	2.84E-03
12	1.49E-03	2.37E-04	3.30E-03
13	6.82E-03	2.80E-04	3.39E-03

Tables 2 and 3 show the MSE and MAPE for the k-step forecast of terminal Voltage, negative particle concentrations, and positive particle concentrations across all runs in T_2. Due to the distribution of run times (see fig.4) the number of runs used to calculate errors for the k-step forecasts is variable. For example, the 8-step forecast is computed using the most runs.

Table 3. MAPE for k-step forecasts vs true observations

k	V	c_{s_n}	c_{s_p}
1	6.77E-03	3.51E-03	9.91E-03
2	4.09E-03	6.09E-03	1.41E-02
3	3.28E-03	8.98E-03	1.94E-02
4	3.43E-03	1.37E-02	2.69E-02
5	3.27E-03	1.87E-02	3.42E-02
6	3.54E-03	2.34E-02	4.10E-02
7	3.87E-03	2.78E-02	4.59E-02
8	4.49E-03	3.19E-02	5.07E-02
9	5.27E-03	3.51E-02	5.38E-02
10	6.55E-03	3.63E-02	5.71E-02
11	6.24E-03	3.83E-02	6.08E-02
12	8.19E-03	4.08E-02	6.53E-02
13	1.33E-02	4.70E-02	6.53E-02

While the error does accumulate over a battery run, the model is able to forecast the entire battery run with surprisingly high accuracy. This can be seen in Fig. 7, which depicts predicted and actual terminal voltage profiles for selected MAPE distribution percentiles for an 8-step prediction. This demonstrates that given only a use cycle and knowledge that the battery is fully charged, the NN model can predict concentration profiles and voltages for the entire trajectory to discharge.
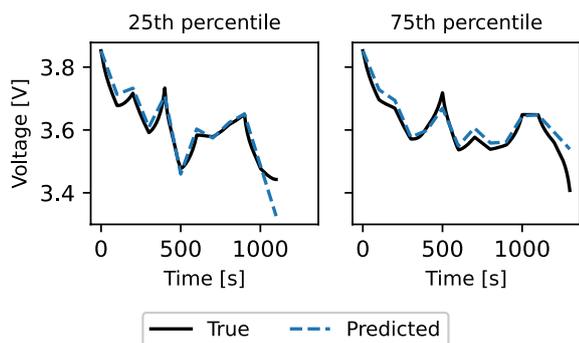

Fig. 7. The figure on the left shows the terminal voltage for a run selected to be close to 25th percentile of the MAPE distribution for T_2 at step k=8, the right shows the 75th percentile.

6. CONCLUSION

We have demonstrated that a NN can replace a Single particle dynamics electrocehmical model of a Li-ion battery and give accurate predictions of voltage, particle concentrations, and failure time through an entire use cycle. This NN has a significantly reduced computational cost which is crucial for control and battery managements strategies.

REFERENCES

Bar-Sinai, Y., Hoyer, S., Hickey, J., and Brenner, M.P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31), 15344–15349.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303–314.

Dao, T.S., Vyasarayani, C.P., and McPhee, J. (2012). Simplification and order reduction of lithium-ion battery model based on porous-electrode theory. *Journal of Power Sources*, 198, 329 – 337. doi: https://doi.org/10.1016/j.jpowsour.2011.09.034.

Doyle, M., Fuller, T.F., and Newman, J. (1993). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical society*, 140(6), 1526.

Dunn, B., Kamath, H., and Tarascon, J.M. (2011). Electrical energy storage for the grid: a battery of choices. *Science*, 334(6058), 928–935.

Forman, J.C., Bashash, S., Stein, J.L., and Fathy, H.K. (2011). Reduction of an electrochemistry-based li-ion battery model via quasi-linearization and pade approximation. *Journal of The Electrochemical Society*, 158(2), A93. doi:10.1149/1.3519059.

Hamilton, A., Tran, T., Mckay, M., Quiring, B., and Vassilevski, P. (2019). Dnn approximation of nonlinear finite element equations. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Marquis, S.G., Sulzer, V., Timms, R., Please, C.P., and Chapman, S.J. (2019). An asymptotic derivation of a single particle model with electrolyte. *Journal of The Electrochemical Society*, 166(15), A3693–A3706.

Mishra, P.P., Garg, M., Mendoza, S., Liu, J., Rahn, C.D., and Fathy, H.K. (2016). How does model reduction affect lithium-ion battery state of charge estimation errors? theory and experiments. *Journal of The Electrochemical Society*, 164(2), A237–A251. doi: 10.1149/2.0751702jes.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.

Speltino, C., Di Domenico, D., Fiengo, G., and Stefanopoulou, A. (2009). Comparison of reduced order lithium-ion battery models for control applications. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 3276–3281. doi: 10.1109/CDC.2009.5400816.

Sulzer, V., Marquis, S.G., Timms, R., Robinson, M., and Chapman, S.J. (2020). Python battery mathematical modelling (pybamm). *ECSarXiv. February*, 7.

Tripathy, R.K. and Bilionis, I. (2018). Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375, 565–588.

Zhu, Y. and Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366, 415–447.

Zou, C., Manzie, C., and Nešić, D. (2016). A framework for simplification of pde-based lithium-ion battery models. *IEEE Transactions on Control Systems Technology*, 24(5), 1594–1609. doi:10.1109/TCST.2015.2502899.