Estimating California’s Solar and Wind Energy Production using Computer Vision Deep Learning Techniques on Weather Images

S. Bosma  N. Nazari
Energy Resources Engineering, Stanford University
367 Panama St, Stanford, CA 94305
sbo@stanford.edu  nazari@stanford.edu

Abstract

In pursuit of a novel forecasting strategy for the energy market, we propose a ResNet-inspired model [6] which estimates solar and wind energy production using weather images. The model is designed to capture high-frequency details while producing realistically smooth energy production profiles. To this end, we show the value of including multiple weather images from times preceding the estimation time, and demonstrate that the model outperforms traditional deep learning techniques and alternative state-of-the-art computer vision methods. Training and testing are performed on a novel data set that focuses on the state of California and spans the year 2019. The dataset, which is sourced from NOAA [1] and CAISO [9], is a secondary contribution of this work. Finally, multiple topics in line with the motivation are proposed for future work.

1. Introduction and Recent Work

As renewable energy capacity continues to exponentially grow, the market supply of electricity is increasingly unpredictable. Simultaneously, more energy intensive industries and a larger workforce are creating sizable swings in energy demand. This fluctuating supply-demand mismatch has led to volatile electricity market prices that are hard to forecast. As both consumers and energy providers can substantially benefit from improved production, demand and price forecasting, this domain has recently seen a spike in interest.

As market complexity increases, the forecasting problem becomes intractable for traditional methods. Recent data driven approaches have tried to offer better alternatives. Most notably, efforts have focused on (i) predicting wind or solar plant power using high resolution local weather data [3, 4, 22, 19] or (ii) predicting energy spot prices using market data and trends [13, 11]. Although successful, these approaches are restricted. The first approach requires large high-resolution weather data sets that are globally unavailable and lead to computationally expensive solution strategies that scale poorly. The second approach fails to directly connect important market drivers like renewable supply to market prices. To our knowledge, few works have attempted to marry mesoscale weather information and system-scale production forecasting [14, 10]. We propose an improved strategy where we leverage existing accurate weather forecasting and computer vision deep learning techniques.

In this work, we step towards a novel framework in which macro-level energy variables (renewable production in this case) are estimated using weather forecast maps. To this end, we first developed an estimator for renewable production using historical surface weather maps. Future work can then apply the developed model to weather forecasts to obtain a renewable production forecast. This step-wise approach in the development stage is chosen such that we can prove the viability of such an estimation model without suffering from weather forecast inaccuracies.

The model was developed using a data set that focused on California – an interesting test case considering the large renewable market share. The renewable energy production data was obtained from the California Independent System Operator (CAISO) [9], and weather maps are sourced from the National Oceanic and Atmospheric Administration (NOAA) via Google Earth Engine (GEE) [1]. The generated data set and extraction code routines are a secondary contribution of this work.

Considering the nonlinear and spatially-connected nature of weather, Convolutional Neural Networks (CNNs) provide the perfect solution. With CNNs’ ability to find complex relationships in data series while preserving spatial information, they are inherently well suited to outperform traditional algorithms. This realization has recently led to CNN techniques being applied to many weather map related use cases. The most explored application improves or complements numerical weather predictions [2, 21, 18, 17, 15]. To suit our purposes, we follow two approaches: a benchmark linear neural network model and a ResNet [6] inspired...
Both models are tested with a single weather input at the corresponding estimation time, \( t \), and with a stack of the five weather input images leading up to \( t \). In our work, we also explore using DenseNets [8] and Long Short-Term Memory (LSTM) models. Other applicable strategies in literature include Deep Belief Networks [3] and Stacked AutoEncoders [4].

This paper is structured as follows. We introduce the renewable energy estimation problem. Next, the generation of the dataset and its properties are discussed. Subsequently, we explore the various models and training designs. The models’ results are then presented, compared, and analyzed. We conclude with several avenues for future work.

2. The Renewable Energy Estimation Problem

Figure 1 describes the problem workflow. For each time step \( t \), a weather map with six bands (or channels) acts as the input image. More details on the data set are provided in Section 3. The model analyzes input images to estimate solar and wind energy powers. In the first CNN or NN models, one input at time \( t \) leads to one solar and one wind power value for time \( t \). To include temporal effects, we investigate the same models with five stacked sequential weather maps, leading to an input image with 30 channels. In recurrent neural network techniques such as LSTM, the model receives both an input image and information from previous images before time \( t \) through so-called hidden data [7].

Figure 1: General workflow of the problem in a conventional deep learning set-up. Input maps are the six weather channels for each sample. Output power estimates are for wind and solar energy.

As mentioned, the long term goal is to forecast production data in advance, e.g. 1 day or 1 week ahead. Hence, one would need to forecast production at time \( t \) with weather maps that we have available at the time of forecasting. However, our model would then not only need to learn to estimate power outputs but also forecast future weather patterns. The latter is a complex task by itself, however, it is also a well established field. As such, in future work we plan to train and forecast using weather forecasts maps. By doing so, we split up the production forecasting task into two distinct problems, each with an specialized model: (i) a weather forecasting model and (ii) a power estimation model. In this work, we aim to prove the viability of the latter when using computer vision techniques on weather maps. To do so, we use historical weather surface maps and match them to the corresponding production at each hour. Separating the weather forecast problem from the power estimation problem also allows for a more flexible framework. As the model does not need to understand local weather patterns, it requires less alterations to be applied to diverse locations (i.e. transfer learning and only adjusting a subset of the convolutional and linear layers). These topics have potential for future work.

3. California Weather Data Set

A secondary yet significant contribution is the dataset. To train the model, we require input weather maps and output renewable energy production from the same time period.

The weather data is extracted from a National Weather Service (NWS) and NOAA real-time mesoscale analysis [1]. The data has a temporal resolution of 1 hour and spatial resolution of 2500m \( \times \) 2500m. Via the Google Earth Engine, we crop the data to a bounding box around the state of California (Figure 2) and between 01-01-2019 to 31-12-2019. When extracting, we set the coordinate reference system to NAD 83 / California Albers (Code EPSG:3310) to ensure the pixels are exported as a straight-standing rectangle similar to the one selected on the map. Due to the earth’s curvature, we lose a couple of pixels in the corners that will be treated in pre-processing.

Figure 2: A snapshot of the data overlaying a topographic map.

To be able to efficiently use the images, the data is down-scaled and normalized. First, the six (out of 13) most important bands are selected: pressure (Pa), temperature (\( ^\circ \)C), humidity(kg/kg), wind speed(m/s), wind direction(\( ^\circ \)), and cloud cover(%). To manage the size of the data set, we
coarsen the pixels to a spatial dimension of $10\text{km} \times 10\text{km}$. Next, the bands are z-score normalized over the whole data set such that they each have mean $\mu = 0$ and standard deviation $\sigma = 1$. The resulting input images are $6 \times 115 \times 108$ [channels $\times$ pixel height $\times$ pixel width] and are displayed for one time sample in Figure 3. Finally, we set the corner pixels equal to 0.

Figure 3: All input channels for the weather image of a single training sample. The parameters displayed are (a) pressure, (b) temperature, (c) humidity, (d) wind speed, (e) wind direction, and (f) cloud cover. Note that each pixel of each weather parameter only has a single value and that the colors displayed are only for visualization purposes (yellow to blue for high to low).

The renewable power data is sourced from the California ISO website and is specified by source and reported every five minutes. To match the temporal scale of the corresponding weather images, we average the power output over an hour. We only focus on solar and wind power as they are unpredictable and very sensitive to weather. Other renewable sources, such as geothermal or hydro, are generally controllable and are predominantly governed by capacity and demand.

We plan to make both the dataset and code routines publicly available. Moreover, we encourage researchers to further improve the proposed model as well as investigate different regions by generating new weather data images from the NWS data in GEE.

3.1. Training and Testing

The dataset is randomly shuffled and divided (80-10-10) into a training, validation, and test set. Considering the size of the dataset, indices of the samples in each subset are stored and efficient data loaders are designed to extract samples, allowing the use of the same train-val-test indices for both the single input and stacked input set-ups. The only exception is that we exclude the first five samples (first five hours on Jan 1, 2019) when using stacked inputs, because these stacked samples are incomplete. Table 1 specifies the number of samples in each data subset (including first five hours).

| Training Set | Validation Set | Test Set |
|--------------|----------------|---------|
| 7008         | 876            | 875     |

In the current set up, we do not preserve chronological order of training and testing data. However, as temporal features are not included as features, the model can not distinguish subsequent steps. Therefore the violation of chronology is acceptable and allows us to test on all 4 seasons instead of only winter. Future work will look into retraining and retesting with chronologically ordered data as time develops. We note that temporal features are often very valuable for time series forecasting. As such, their omission potentially makes the current forecasting problem significantly harder. We expect to see improved results when including them in future work.

4. Models

In this paper, we present a reference linear model and a ResNet inspired convolutional model. DenseNets and LSTM-CNN models are also briefly discussed, though not displayed, because their current performance was mediocre. To compare the models fairly, we try to keep as many other design choices similar between the models and compare their losses and accuracy metrics. All implementation is done using the Pytorch library.[16]

4.1. Loss Function and Optimization

The final layer in each model outputs a power value for solar and wind energy. As these are continuous outputs, losses for the estimations are computed using root mean squared errors ($RMSE$):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}.$$  \hspace{1cm} (1)

To minimize the loss, the model weights are updated and optimized using backpropagation and the ADAM algorithm.[12] ADAM is chosen because it exhibited the most consistent loss evolution. We recognize that ADAM can converge to a sharp local minima that may be far away from the global minimum. Optimizing with more-sensitive yet less robust optimizers, such as Stochastic Gradient Descent...
Momentum, is the subject of future work. Here, we aim to present respectable results, showing the potential of our proposed methodology.

All models are trained for a maximum of 20 epochs in four stages. At each stage, the learning rate is decreased to allow finer adjustments of the model. L2 regularization is applied to reduce overfitting. After hyperparameter tuning, we choose regularization weights of $\lambda = 0.01$ and $\lambda = 0.001$ for the linear and ResNet models, respectively.

4.2. Accuracy Metric

In addition to the RMSE loss, we design a more meaningful accuracy metric to qualitatively evaluate the performance. To this end, we assess an absolute error relative to the average power output, $y_{\text{train}}$, in the training set, i.e.

$$\text{Acc}_i = 1 - \frac{|\hat{y}_i - y_i|}{y_{\text{train}}},$$

where $\hat{y}_i$ is the predicted power output and $y_i$ is the actual power output for sample $i$. Note this metric is computed separately for solar and wind production because the mean power outputs of the training data are $3.02 \text{ MW}$ and $1.68 \text{ MW}$, respectively.

4.3. Linear Neural Network Model

A linear model is designed to start wide and reduce to the two output classes in four layers—800, 400, 200, and 2 nodes. The input image is flattened and then connected to the first fully connected block. A block consists of a fully connected layer preceded by a dropout layer with a dropout probability equal to 0.2 and followed by a ReLU activation function. Note that we also add an unconventional final ReLU activation layer after the last fully connected layer. This addition ensures that outputs are strictly positive and proved to be essential in enabling the model to capture the zero solar output at night. This final ReLU layer also motivates the need for dropout layers. Note that the final ReLU layer deactivates a large share of the network at initialization, and without dropout would prevent them from activating during training. Hence, the dropout layers are implemented to enable the activation of layers that are initially deactivated.

4.4. ResNet-inspired Convolutional Model

A ResNet-inspired model is designed to take advantage of the spatial correlation in the data. The designed model consists of 38 layers, including 10 “bottle-neck” building blocks [1]. These building blocks and the overall network are presented in Figure 4. Every convolutional layer is followed by a batch normalization and a ReLU activation function. Additionally, average pooling layers are included to reduce the image dimensions. Finally, two linear layers lead to the power estimates. The linear layers and final layer follow the same reasoning as in the linear model already presented. Although this model has significantly more layers, it does not have more parameters than the presented linear model. Furthermore, all convolutional layers are initialized using Kaiming initialization [5], linear layers are initialized with a uniform random distribution, and bias terms are set to 0.

4.5. Other Models

A DenseNet model and a CNN-LSTM model were also investigated. DenseNet did not prove more effective than the ResNet-inspired model and therefore was not included in this report. The initial results of the CNN-LSTM model were mediocre, partly due to data and computational power shortage. Future work will attempt to unlock higher accuracy with these time-series specialized methods using greater resources.

5. Results

The models’ performances on the 2019 California data set are summarized and compared in the following subsections. We evaluate the loss and accuracy to understand the prediction results. Additionally, saliency maps are presented to better the understanding of the model’s vision.

The overall accuracy is presented in Table 2 and Table 3. The ResNet model with five stacked sequential weather maps as input outperforms all the other models in both wind
and solar estimation. The other three models exchange the rest of the ranks depending on energy source and data sub-set. The following subsections explain the reasons for these performances.

Table 2: Model Solar Power Accuracy.

| Model          | Training | Validation | Test  |
|----------------|----------|------------|------|
| Single Linear  | 0.9319   | 0.8673     | 0.8645|
| Single ResNet  | 0.9335   | 0.8350     | 0.8340|
| Stacked Linear | 0.9208   | 0.8736     | 0.8813|
| Stacked ResNet | **0.9414** | **0.8882** | **0.8840** |

Table 3: Model Wind Power Accuracy.

| Model          | Training | Validation | Test  |
|----------------|----------|------------|------|
| Single Linear  | 0.8800   | 0.8469     | 0.8454|
| Single ResNet  | 0.9025   | 0.8426     | 0.8433|
| Stacked Linear | 0.8360   | 0.8103     | 0.8189|
| Stacked ResNet | **0.9037** | **0.8668** | **0.8713** |

5.1. Loss Evolution

Figure 5 displays the RMSE loss evolution on the full training and validation data sets during training. Both ResNet models perform very well on the training data. However, the single input model overfits and has poor validation loss. High regularization weights or increased dropout fails to improve its performance. Similarly, the validation-training gap for the single input linear model is larger than a gap with a stacked input. In this case the validation losses are comparable. The following subsections explain these discrepancies by investigating the solar and wind power predictions.

5.2. Solar Power Estimation Accuracy

We analyze the models’ solar power estimates by plotting the evolution of our accuracy metric, see eq. (2), in Figure 6 and the power estimates for a chosen week as shown in Figure 7.

Figure 6: Solar power estimation accuracy evolution while training for each of the different models.

The solar accuracy evolution shows similar behavior to the RMSE loss. The most notable difference is that the stacked input models exhibit better accuracy for the same loss value. Especially interesting is how well the linear models perform. Since solar intensity is a highly local quantity, the model’s main objective is locating important solar locations (with the exception of forecasting incoming cloud cover). Because spatially correlated information is not required, the linear model is adequately well-suited. Figure 7 reflects this suitability by comparing the true solar output to the model estimates. The plots also show a distinct difference in smoothness where the single input models exaggerate the high frequency variations at high power output.

5.3. Wind Power Estimation Accuracy

Wind power is a more complex energy source and less temporally predictable, making the estimation problem more challenging. As such, Table 3 presents the results that reflect lower scores than obtained for solar power. Additionally, Figure 8 indicates that the linear models are relatively less effective when compared to the convolutional models. This result is consistent with the importance of spatially correlated data for wind power. We also point out that the stacked ResNet model clearly outperforms the other models.

Figure 9 displays the wind power estimates for the first full week of May in 2019. Again, we see that the stacked input models present much smoother predictions. However, in this case, we see that the stacked linear model overdoes...
curves which, although accurate, are not very realistic. The stacked ResNet model takes the best of both worlds and outputs an accurate and realistic representation of the true wind power output.

Figure 8: The wind power estimation accuracy evolution while training for each of the different models.

5.4. Saliency Maps

Saliency maps inform us which parts of an input image influence the network output. To compute a saliency map, we compute the absolute value of the model output’s gradient with respect to the model input and then take the maximum value over the channels to represent the pixel intensity.

Figure 10 presents saliency maps of two different time stamps. As expected, we see that the solar saliency map at 22:00 is black because there is no sun and the output is equal to 0. On the other hand, we see a semicircular pattern of high intensity at noon, starting near the north central valley, passing by L.A. and bending towards Las Vegas. Comparing this pattern with the location of solar plants, we see a clear correlation (Figure 11). Also interesting is that both wind and solar saliency maps correctly show low intensity in the ocean areas, which are of little importance. Note that this probably would have been different if we had tried to forecast energy based on old weather maps preceding the target time \( t \) (a much more complicated task).

Figure 7: Solar energy comparison between true power output and models’ estimations. The data reflects the week of May 5 - May 11, 2019. Note the data is displayed chronologically for visualization purposes. Training and testing is performed with a random sampling of data points. As such, the visualization includes training, validation, and testing data samples.

Figure 9: Wind energy comparisons between true power output and models’ estimations. For notes on visualization data, see Figure 7.

Figure 10: Wind power estimation accuracy evolution while training for each of the different models.
Figure 10: Saliency maps for two different samples: May 8 at 12:00 (noon) and 22:00.

The wind saliency maps also show bright spots at corresponding locations to the wind power plants. These bright spots are larger and also cover populated areas. The increased spatial dimension corresponds to the importance of a larger spatial area around a wind power plant. The explanation as to why the intersection with locations of high population density is less straightforward. One possibility is that wind energy is subjected to curtailment depending on demand. Hence, the model might use information about weather conditions of surrounding cities to implicitly estimate energy consumption, which in turn would influence the energy production estimate.

5.5. Discussion & Future Work

As this is an initial exploration of this application, the results could be further improved by adjusting model parameters, overcoming data limitations, and seeking more complex models.

The saliency maps display a reasonable amount of low amplitude noise. We expect a near-perfect model to display the same targeted bright spots without any noise in unimportant regions. Moreover, the maps also indicate the importance of demand and other parameters. To maximize model accuracy, extra data streams, such as temporal features, demand forecasts and sun angle, should be added to allow the model to more accurately focus on the important features in the input weather maps.

Incorrect production data also significantly impacts the models. Although largely pristine, the data set contains two days where the wind production data is (presumably falsely or through human intervention) reported as flat. This misinformation can throw off our model during training and negatively influence the model’s accuracy metrics. Figure 12 exhibits the solar and wind production graphs for October 6th and 7th, 2019. One can opt to exclude these data points, however, we decided to include them because the incorrect data points are few in number and only relate to wind energy. Additionally, by including them, the full dataset remains complete and chronologically intact. A better strategy might be to use a model to generate synthetic data for October 6 and 7. As such, one would avoid obtaining false correlations and still fully take advantage of the dataset.

Figure 11: Location of Electricity Producers in California by source.

Figure 12: Solar and wind production for days with incorrectly reported wind energy.
6. Conclusion & Future Work

This work takes a first step towards a novel approach in forecasting renewable energy production and, in the bigger picture, any large-scale energy market variable. To this end, we propose using state-of-the-art computer vision techniques on weather maps to estimate renewable energy production. Our ResNet-inspired model outperforms other traditional deep learning techniques and obtains accuracies close to 90% for both solar and wind power. We also show the value of using multiple input frames from times leading up to the estimation. The model’s performance is justified through comparison with other models, the nature of the energy to be estimated, and saliency maps indicating the focal points of the model’s "vision". Worth noting, the model independently assigns importance to data at verifiable energy producing locations.

This work is not an end, but rather a beginning of a novel strategy to forecast energy market variables. In future steps, the model will be applied to weather forecasts (rather than historical data) to obtain wind and solar power forecasts. We also emphasize that the results presented in this work are achieved using solely weather data as input. Additional non-image data, such as temporal data or sun incidence angle, should be included as inputs to further improve performance and more accurately leverage weather map data.

The presented methodology can also be applied to other important society scale variables. For example, energy demand is closely correlated to weather because of indoor climate control. In this line of thinking, one could envision a set up similar to state-of-the-art image classification model infrastructure. That is, a framework where sub-models predict lower level variables which are then used as input in the global model to predict the primary variable. In particular, forecasts of complex market variables, such as energy price, might benefit significantly from a multi-level structure.

Another interesting tangential avenue could explore how to use these renewable energy forecasts to improve storage capacity design and energy-intensive process scheduling. If successful, this exploration will lead to industry savings and allow an increase in renewable energy share on the grid.

Finally, future work naturally includes enhancing the presented models. This work serves as a proof-of-concept and we encourage others to extend our research. In this, alternative architectures and models are of interest, e.g. CNN-LSTM or temporal convolutional network (TCN). Additionally, higher resolution input data should be explored and new locations of interest can be evaluated. Transfer learning and conducting training/testing at different locations would also be valuable topics of future work in going towards real-world deployment.

Acknowledgements

The authors thank Jon Braatz and Stace Maples for their support in collecting the weather data. We also thank Robyn Lockwood for her great help in revising the manuscript. Finally, we thank Titiaan Palazzi, Patrick Sheehan and Pieter Verhoeven for the interesting discussions which led to this work.

Contributions

The model and methods were designed by SB and NN. Hyperparameter tuning was performed by SB and NN. Weather data was collected by SB. Power data was collected by NN. SB and NN wrote the paper.

References

[1] National Atmospheric and Oceanic Administration. Rtma: Real-time mesoscale analysis. https://developers.google.com/earth-engine/datasets/catalog/NOAA_NWS_RTMA#description
[2] A. Chattopadhyay, P. Hassanzadeh, and S. Pasha. Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data. Scientific reports, 1317, 2020.
[3] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He. Short-term load forecasting with deep residual networks. IEEE Transactions on Smart Grid, 10(4):3943–3952, 2019.
[4] A. Gensler, J. Henze, B. Sick, and N. Raabe. Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Proceedings:2858–2865, 2016.
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV ’15, page 1026–1034, USA, 2015. IEEE Computer Society.
[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[7] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.
[8] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.
[9] California ISO. Supply and renewables. [http://www.caiso.com/TodaysOutlook/Pages/supply.aspx]
[10] Han Seung Jang, Kuk Yeol Bae, Hongshik Park, and Dan Sung. Solar power prediction based on satellite images and support vector machine. IEEE Transactions on Sustainable Energy, 7:1255–1263, 07 2016.
[11] L. Jiang and G. Hu. Day-ahead price forecasting for electricity market using long-short term memory recurrent neural network. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 949–954, 2018.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv, 2014.

[13] Jesus Lago, Fjo De Ridder, and Bart De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Applied Energy, 221:386–405, 2018.

[14] Jesus Lago, Fjo De Ridder, and Bart De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Applied Energy, 221:386–405, 2018.

[15] Johan Mathe, Nina Miolane, Nicolas Sebastien, and Jeremie Lequeux. Pvenet: A lrcn architecture for spatio-temporal photovoltaic powerforecasting from numerical weather predichion. arXiv, 2019.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raixon, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alch´e Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

[17] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark dataset for data-driven weather forecasting. arXiv, 2020.

[18] S. Scher. Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. Geophysical Research Letters, 45(22):12,616–12,622, 2018.

[19] Navin Sharma, Pranshu Sharma, David Irwin, and Prashant Shenoy. Predicting solar generation from weather forecasts using machine learning. IEEE International Conference on Smart Grid Communications (Smart Grid Comm), Brussels, Belgium, 10 2011.

[20] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013.

[21] Jonathan A. Weyn, Dale R. Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. submitted to Journal of Advances in Modeling Earth Systems, 2020.

[22] A. Zhu, X. Li, Z. Mo, and R. Wu. Wind power prediction based on a convolutional neural network. International Conference on Circuits, Devices and Systems (IC-CDS), Proceedings:131–135, 2017.

Preprint submitted on 3/15/2021