Pattern Based Multivariable Regression using Deep Learning (PBMR-DP) CVPR Proceedings

Jiztom Kavalakkatt Francis[1], Chandan Kumar[2], Jansel Herrera-Gerena[2], Kundan Kumar[2], Matthew J Darr[3]
jiztom ,chandan, janselh, kkumar, darr@iastate.edu
Computer Engineering[1], Computer Science[2], Agricultural and BioSystems Engineering[3]

Abstract

We propose a deep learning methodology for multivariable regression based on pattern recognition triggering fast learning over sensor data. Our sensors-to-image conversion enables us to take advantage of Computer Vision architectures and training processes. We also explore the use of state-of-the-art architectures to generate regression outputs to predict agricultural crop continuous yield information. Finally, we compare with some of the top models reported in MLCAS2021. We found that using a straightforward training process, we were able to accomplish an MAE of 4.394, RMSE of 5.945, and R^2 of 0.861.

1. Introduction

In the recent years, machine learning algorithms have been improving dramatically in different areas. Unsupervised methods have been incorporated in the deep learning field to solve image-based problems, sound, and text. With neural network architectures changing continuously the training process is also changing. Some works have made changes into the backbone network [13] to achieve better results. But sometimes innovation prevents simpler ideas from being developed. Here, we present our work that combines state-of-the-art image architecture and regression.

Inspired by the data provided in [10], which contains information of multiple sensors with time-stamp. We decided to take a different approach and explored the conversion of this dataset into images (Section 3.1). First, we explored the conversion of sensor data into an accurate image-like data, and then make changes in the neural network architecture as common CV architectures do not tend to give regression as output which was the case for our model. This allows us to perform multivariable regression as in [1] which is pattern-driven instead of data-driven.

In this work, we present two major contributions. First is constructing sensors-to-image conversion in which tabular data can be represented as an image. This facilitates the use of modern CV architectures. Secondly, using these sensors-to-image samples to predict continuous crop yield values.

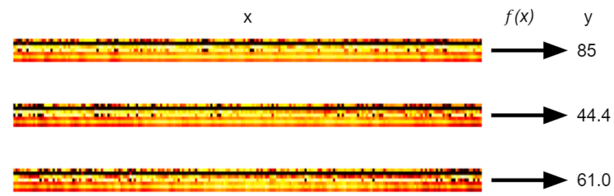

Figure 1. Three samples of tabular input data (x) converted to an image mapped, by our model $f(x)$, to their predicted (y) value.

2. Related Works

Long short-term memory (LSTM) based architectures usually take a lot of resources to perform the training process.We, hence, explored methods performing regression over images leading to image age detector, which affirmed our concerns. [9] deals with the creation of two Convolutional Neural Networks (CNNs), to predict gender and age with a classifier instead of a regressor. [4] works on predicting the rotation angle of given images. A similar idea can be seen in [8], which shows a CNN regression framework for predicting 3D pose estimation. We explore the conversion of sensor data into images such as [15]. Therefore, their conversion was more complex than in this work, but the idea of generating these images is viable. Visualizing sound as images [3, 11] with DNNs improves accuracy and reduces computational requirements from classical methods of event or pattern recognition [7]. Earlier proposed solutions require conversion and generation of custom CNN [2]. Detecting patterns requires much pre-processing with feature engineering. The process is time-consuming and requires extensive study of the correlations.

3. Method

In this section, we explore the input pipeline, architecture design, and our methodology.

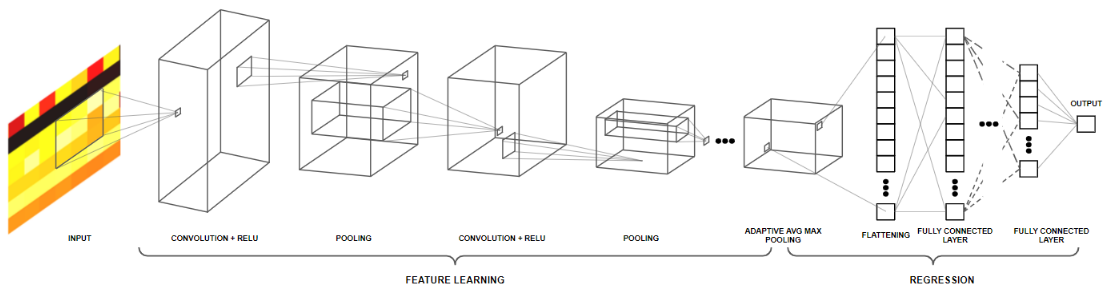

Figure 2. Proposed model architecture. The input is the pre-processed image like 3D-array passed to Convolutional Neural Network (Feature Learning). Using an Adaptive Concat Pooling mechanism and FCN the required single regressor output is generated.

3.1. Input Data

We use Soybean Crop Yield dataset found in the MLCAS2021 challenge. It is a temporal data computed in real-time and is composed of 93000 samples over 214 days with seven sensor readings, each pointing to a Single Crop Yield (y). There are some additional information such as genotype ID, location ID, and year for each sample, which are also treated like sensor data by us. Sensor readings can be noisy due to different measuring speeds of the dataloggers [6]. The initial assumption is that all the data is measured over the same time-space, corrected, or spread to a fixed tabular form. Sensor data is considered as the ranges for sensors are absolute, ensuring that on normalization stage in pre-processing values are between 0 and 1.

3.2. Pre-processing

Before feeding machine learning models with input data, the original data is pre-processed and statistically analyzed extensively. This process is time-consuming and requires human and computer resources to verify the correlation of the data to the output it is being trained with. Our process converts tabular data into images. The input data is arranged as rows with time along the y-axis and apply a Row Normalization technique. Each row is normalized based on the absolute range of the sensors Eq. (1) to ensure that the final table contains values between 0 and 1.

$$\overrightarrow{x_{ij}} = \frac{x_{ij} - \sigma(s_i)}{\lambda(s_i) - \sigma(s_i)} \tag{1}$$

where $\overrightarrow{\boldsymbol{x}_{ij}} \in [0, 1]$ is the normalized data point at positions i, j. The values in x_{ij} represent the original tabular data in which i represents the row (our sensor), and j the time in our dataset. In addition, $\sigma(s_i)$ and $\lambda(s_i)$ represent absolute minimum and maximum values of sensor $s_i \in S$ where S is the set of all the sensors.

Our data preparation method explained above can be fed directly to CNNs without major modifications to the architecture. The tabular data must be across a common measurement axis, such as time series or measured at the same interval. Missing values are accommodated with the previous value ensuring noise is reduced to a minimum in the input data. Fig. 1 shows how the data can be visualized with patterns.

3.3. Model Input

We directly generate a 3D data array in range 0 and 1. The data is normalized specifically to each row and not batch normalized for the entire slice. Normalization is performed since each row is sensor data over time with absolute ranges. On experimentation using a batch normalization method with unique time-series data, sensors with very small ranges were found to have limited or low impact on the final results. The generated data (Fig. 1) is fed into the models to maximize the learning ability of neural networks instead of solving for best fit.

3.4. Architecture Design

The model relies on the feature learning, heavily used in CNN classification models. We modify layers to convert them into a regression pattern model Fig. 2, which outputs a single regression yield output instead of class probability with softmax. Instead of classification, we introduce an Adaptive Concat Pool layer(combination of Adaptive Average Pool and Adaptive Max Pooling layers) after the feature learning layers to understand regression data. This custom layer allows us to convert the problem into a FCN approach to the regression values. The use of DNNs with different optimizers and fixed hyper tuning allows us to maximize the results. These adjustments that followed the state-of-the-art architectures create a single output for each 3D input.

4. Experiment

We evaluate our approach with different state-of-the-art machine vision models. We conducted our experiment on Crop Yield Regression Task [10]. It is a multivariable regression problem with 7 daily variables measured over a fixed time period of 214 days. We run our models on Intel i9-10900k CPU with 128 GB 2666MHz RAM and

Performance		Models		
		Resnet 50	EfficientNet B0	ResNeXt 50
MAE↓	SGD	4.529	5.535	**4.394**
	ADAM	5.496	5.232	5.371
	LARS	4.644	6.577	5.191
RMSE↓	SGD	5.963	7.312	**5.945**
	ADAM	7.258	6.958	7.118
	LARS	6.266	8.586	6.889
$R^2\uparrow$	SGD	0.849	0.789	**0.861**
	ADAM	0.792	0.809	0.799
	LARS	0.845	0.709	0.812

Table 1. Performance metrics with different standard models using different optimizers. All models run with the learning rate and batch size specified in Section 4.

Competition Teams	Model approaches	Performance		
		MAE↓	RMSE↓	$R^2\uparrow$
QU(exp006)	Statistical Modelling	4.41	**5.89**	**0.87**
CUFE	ensemble Regression	4.42	5.95	0.86
Star	M/4*1D-CNN with Ensemble	4.47	5.95	0.86
Elendil	M/7*1D-CNN with Ensemble	4.47	5.95	0.86
AA2	XgBoost	4.6	6.15	0.85
PBMR-DP	ResNeXt 50 - SGD	**4.39**	5.94	0.86

Table 2. Comparison with the models submitted in MLCAS2021 .

NVIDIA RTX 3090 with 24 GB VRAM. For our experiments, the learning rate was set to $1e^{-03}$ with a batch size of 128 for 1,000 epochs and MSEloss or L1-loss. We follow [5, 12, 14] to construct the Feature learning stage of the models (depth). The pooling layer is modified to a custom Adaptive Concat Layer with Fully connected layers pointed to a single output.

4.1. Performance Metrics

We used the standard metrics such as Mean Average Error (MAE), Root Mean Square Error (RMSE), and R^2 to evaluate the performance. The loss function used in the model is MSEloss or L1loss. K-cross-validation is performed to overcome over-fitting of data. The data was tested and compared with the same test dataset as the MLCAS2021 competition to keep the results and metrics constant and form a common comparison baseline.

Regression Analysis Techniques	Performance		
	MAE↓	RMSE↓	$R^2\uparrow$
Linear Regression	6.100	8.121	0.740
Elastic Net	9.103	11.548	0.471
LASSO	9.987	12.790	0.363
SVR-RBF	5.976	7.875	0.758
Stacked-LSTM	5.484	7.276	0.792
Temporal Attention	5.441	7.239	0.795
PBMR-DP	**4.394**	**5.945**	**0.861**

Table 3. Different performance metrics on the Soybean Crop Yield Data performed using the published ML models.

5. Results and Discussions

Table 1 shows the results gathered when comparing the different networks with different optimizers such as Stochastic Gradient Descent, Adam Optimizer, and LARS with the same parameters and metrics described in 4. ResNeXt50 with SGD optimizer performed the best in the three different metrics used for this experiment. The second and third best models were ResNet50 with SGD and LARS, respectively.

Table 2 shows the performance of different teams from the MCLAS Challenge. Most of the results involved using ensemble techniques to combine weights generated using different models to get the best results. Our approach is much simpler in comparison and outperforms the methods in the competition except for one method. Our model is trained without optimizing any hyperparameters as we wanted our solution as a general method.

Table 3 shows the crop yield prediction dataset results. Our results prove a significant increase in prediction performance. In addition, our approach allows faster data to model regression without the need for analysis of the correlation between the inputs and the output. This table shows a comparison with previous works over same dataset. We see that our model outperforms each of these methods.

6. Conclusion

This work provides a pattern-based approach for multivariable regression. With our sensor-to-image conversion, we bring computer vision and convolutional neural network techniques to regression tasks. Our method of sensor-to-image conversion is completely lossless. Our experiment with multiple models and different optimizers proves the validity of our method. We have outperformed every classical approach and are at par with the best ensemble methods. In addition, we hope to make a significant impact with tabular data and advance the research even further in these areas.

References

[1] Evangelos Alexopoulos. Introduction to multivariate regression analysis. *Hippokratia*, 14:23–8, 12 2010. 1

[2] Roy Assaf and Anika Schumann. Explainable deep neural networks for multivariate time series predictions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6488–6490. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 1

[3] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals, 2019. 1

[4] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Image orientation estimation with convolutional networks. In *GCPR*, 2015. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[6] Jiztom Kavalakkatt Francis. *Cloud-Based Multi-Sensor Remote Data Acquisition System for Precision Agriculture (CSR-DAQ)*. PhD thesis, Iowa State University, 2019. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-22. 2

[7] Zvi Kons and Orith Toledo-Ronen. Audio event classification using deep neural networks. In *INTERSPEECH*, 2013. 1

[8] Siddharth Mahendran, Haider Ali, and Rene Vidal. 3d pose regression using convolutional neural networks, 2017. 1

[9] Abdullah M. Abu Nada, Eman Alajrami, Ahemd A. Al-Saqqa, and Samy S. Abu-Naser. Age and gender prediction and validation through single user images using cnn. In *Semantic Scholar*, 2020. 1

[10] Johnathon Shook, Tryambak Gangopadhyay, Linjiang Wu, Baskar Ganapathysubramanian, Soumik Sarkar, and Asheesh K. Singh. Crop yield prediction integrating genotype and weather variables using deep learning. *PLOS ONE*, 16(6):e0252402, Jun 2021. 1, 2

[11] Sugianto Sugianto and Suyanto Suyanto. Voting-based music genre classification using melspectogram and convolutional neural network. In *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 330–333, 2019. 1

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 3

[13] Subrahmanyam Vaddi, Dongyoun Kim, Chandan Kumar, Shafqat Shad, and Ali Jannesari. Efficient object detection model for real-time uav application, Jan 2021. 1

[14] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017. 3

[15] Chao-Lung Yang, Zhi-Xuan Chen, and Chen-Yi Yang. Sensor classification using convolutional neural network by encoding multivariate time series as two-dimensional colored images. *Sensors*, 20(1), 2020. 1