Machine Learning Engineer Nanodegree

Capstone Project

Thi Nguyen

March 28th, 2019

Predicting Physiological Traits from Hyperspectral Reflectance Measurements using Advanced Machine Learning Models

I. Definition

Project Overview

As global population is estimated to reach 9.7 billions by 2050 [1], the projected demand for cereal grain is far exceeding the current agricultural output [2]. In order to meet the projected global food demand, the world-wide crop production is required to be double [3]. The efficient use of physiological traits to raise wheat yield potential is the major target for agricultural researchers.

It is desirable to determine the amount of yield for wheat plants early on during the growth of the plants, instead of waiting for the end. A way of predicting the potential yield of plants is to look at the current biochemical and physiological traits of the plants. Measuring Photosynthesis-related traits, such as nitrogen per unit leaf area (Narea) and leaf dry mass per area (LMA), require laborious, destructive, laboratory-based methods, while physiological traits underpinning photosynthetic capacity, such as maximum Rubisco activity normalized to 25 °C (Vcmax25) and electron transport rate (J), require time-consuming gas exchange measurements.

The project aims to replace the traditional time-consuming laboratory-based methods by fast and high-throughput machine learning models by using leaf-level hyperspectral reflectance parameters to predict the physiological traits.

Problem Statement

The aim of this project is take simple leaf reflectance measurements, which is data is much easier to collect, and then predicting the biochemical and physiological traits. We aim to using deep learning and machine learning models to assess whether hyperspectral reflectance (350–2500 nm) can be used to rapidly estimate these trait values on intact wheat leaves. The proposed models are using gas exchange and hyperspectral reflectance data from 76 genotypes grown in glasshouses with different nitrogen levels and/or in the field under yield potential conditions.

Metrics

I will use the coefficient of determination (R^2) to evaluate the performance of my models to the benchmark [4].

The R^2, the model bias is defined as:

$$\text{Bias (\%)} = 100 \times (\bar{\hat{y}} - \bar{y}) / \bar{y} \qquad (1)$$

to represent the percentage of the difference between the mean of the predicted trait, $\bar{\hat{y}}$, and the mean of the observed trait, $\bar{y}$.

II. Analysis

Data Exploration

The plant dataset consists 1185 total number of records supplied by the Center of Excellence in Plant Energy Biology at the Australian National University (ANU). The data is collected by Aus 1, Aus 2, Aus 3 and Mex 1 experiments from two different geographical locations (Mexico and Australia).

Input data:
Hyperspectral reflectance data is for each leaf image. At the raw level there is hyperspectral reflectance curve for each pixel. The data is captured by a FieldSpec®3 (Analytical Spectral Devices, Boulder, CO, USA). Basically, the intensity of the reflected

light at different wavelengths. The range of wavelengths measured is between 350 and 2500 nm.

Targets: There are 10 biochemical and physiological trait observations to be predicted: LMA, Narea, SPAD, Nmass, Pmass, Vcmax, Vcmax25, J (ETR), Photo, and Cond. The basic statistics of the trait observations are shown as bellows:

	count	mean	std	min	25%	50%	75%	max
LMA_O	525.0	0.004716	0.965413	-2.544563	-0.633425	0.075898	0.598243	2.702531
Narea_O	525.0	0.002739	1.012551	-2.919862	-0.476220	0.156314	0.665426	2.343069
SPAD_O	614.0	-0.005757	0.966404	-3.711706	-0.286792	0.139862	0.566515	1.899076
Nmass_O	615.0	-0.019518	0.995458	-3.978608	-0.301191	0.090976	0.604582	1.840075
Pmass_O	435.0	-0.010429	1.061748	-5.587649	-0.743357	-0.349462	0.604679	3.980818
Vcmax	488.0	0.032144	0.972976	-2.128350	-0.580143	-0.133495	0.638442	3.775657
Vcmax25	488.0	0.041765	0.972382	-3.002729	-0.454763	0.114690	0.642235	2.484561
J	488.0	-0.032477	1.011349	-3.350207	-0.513753	0.050951	0.694887	1.923221
Photo_O	488.0	-0.010209	1.037001	-3.417375	-0.525545	0.247456	0.742518	1.829005
Cond_O	488.0	-0.016069	0.990810	-1.834683	-0.749132	-0.125851	0.571300	3.671617

Table 1: Summary of statistics of the trait observations

Data cleaning: all missing and duplicate values are removed from each traits. At the end, LMA and Narea both have 525 records, SPAD and Nmass both have 615 records, Pmass both has 435 records, Vcmax, Vcmax25, J, Photo and Cond both have 488 records.

Exploratory Visualization

In this section, I plotted the statistics of the hyperspectral reflectance parameters (inputs) for all traits and their observation values (predictors).

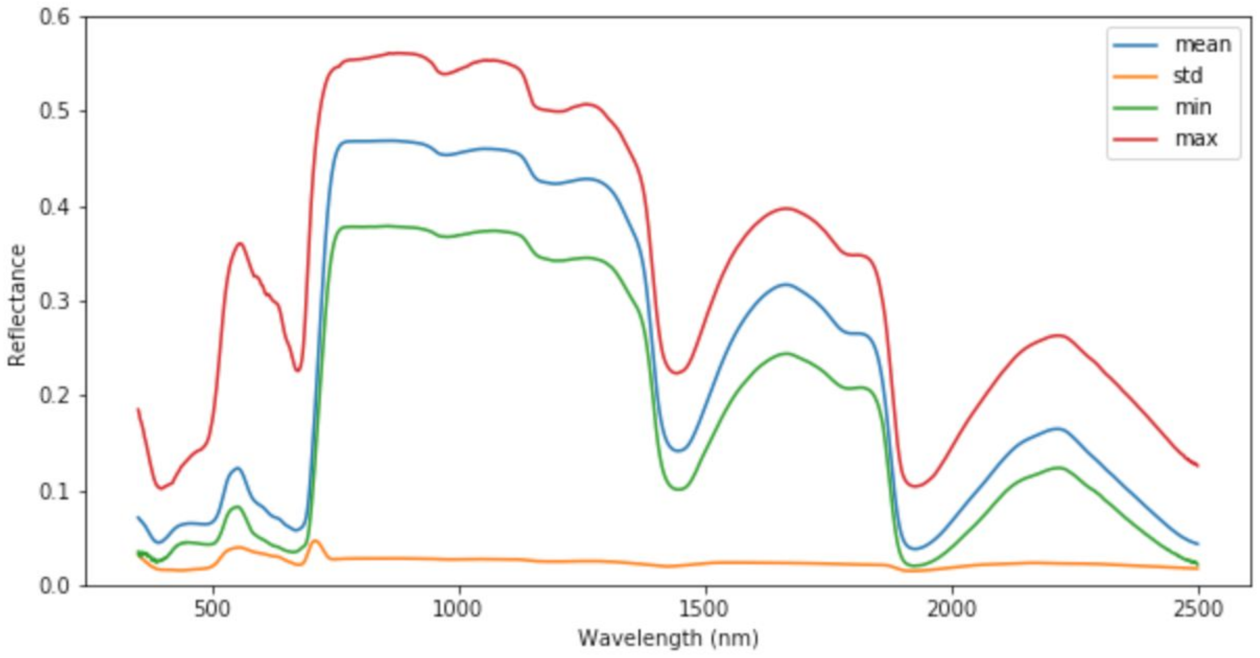

Fig. 1: The hyperspectral reflectance parameters for all traits

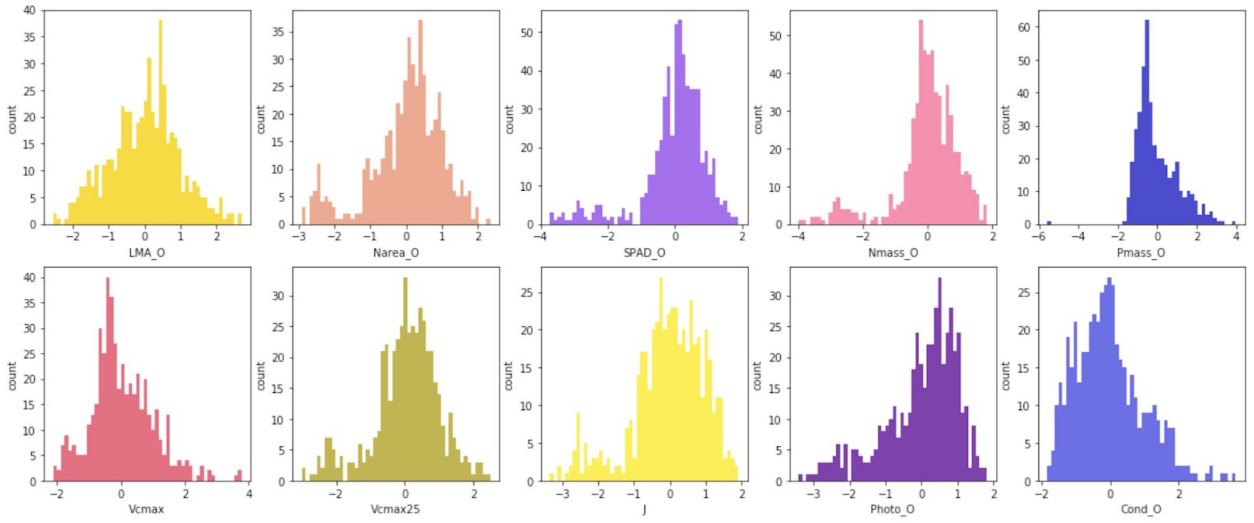

Fig. 2: The histograms of all traits observations

It is observed that the distribution of traits observations are highly skewed. While most the observations fall into -1 to 1 ranges. There are possible problems of unbalanced dataset as there are not enough data to train and predict minority observations.

Algorithms and Techniques

The problem is classified as a regression problem which can be be solved using 1D convolutional neural networks and as well as advanced machine learning frameworks like XGBoosting and LightGBM with the following steps:

1. Split the data into training, validation and test sets.
   I splitted the data set into three separate sets called: training set (70 %), validation set (10%) and testing set (20%). Training and validation sets are used for train the models and test set is used to validate the performance at the final stage.
2. Constructing the convolution neural network (CNN) model. CNN model is the current the state-of-the-art for image classification and detection. I would like to investigate whether CNN is suitable for solving 1 dimensional regression data like the plant dataset. As the dataset is having relatively small number of records, a simple CNN architecture would be are desirable to avoid overfitting. Multiple CNN architectures will be constructed to find the best fit for the problem. Some of the architecture parameters are considered including:
   a. number hidden units (e.i. layers)
   b. filter size: filter size of 3 and 5 will be used for the model.
   c. stride: strides of 3 and 5 are used in the model.
   d. number of filters: large number of filters will be used for the convolution layers
   e. activation functions: the ReLu is the default activation function which are popular choice for many well known CNN architecture.
   f. dropouts: I used dropout layers for prevent overfitting. The default rate is 0. Later, I will tune this value for some found overfit models.
   g. optimizer: I will use Adam optimizer, the stochastics gradient descent optimizer that has shown itself to work well under different conditions. The R squared cost function is used for optimizing. The learning rate and weight decay also will be tuned during the hyperparameter tuning.
   h. loss function: R squared loss function is the default choose
3. Extreme Gradient Boosting (XGBoost) and LightGBM are both implementations of the gradient boosting frameworks that are highly efficiency, accuracy, and interpretability. I will train both models to compare with the convolution neural network model.
4. Implementing and training Partial least squares regression (PLS regression). This statistical method is used by Viridiana *el al.* [4] to predict physiological traits from leaf reflectance measurements.
5. Perform hyperparameter tuning
6. Compare results and reporting

Benchmark

We test our proposed CNN model and XGBoosting to compare with the existing method [4] used partial least squares regression (PLS regression) to map input to output.

III. Methodology

Data Preprocessing

I performed data cleaning process to remove all N/A and duplicated data from the data set. No further preprocessing is needed due to data was captured by a FieldSpec which is well defined.

Implementation

The convolution neural network (CNN) model was implemented in PyTorch framework. The code used to define the model is available from the Github repository accompanying this report as HyperSpectrumModel.py (class HyperSpectrumModel4Layers). The visualization of the CNN model is shown bellows:

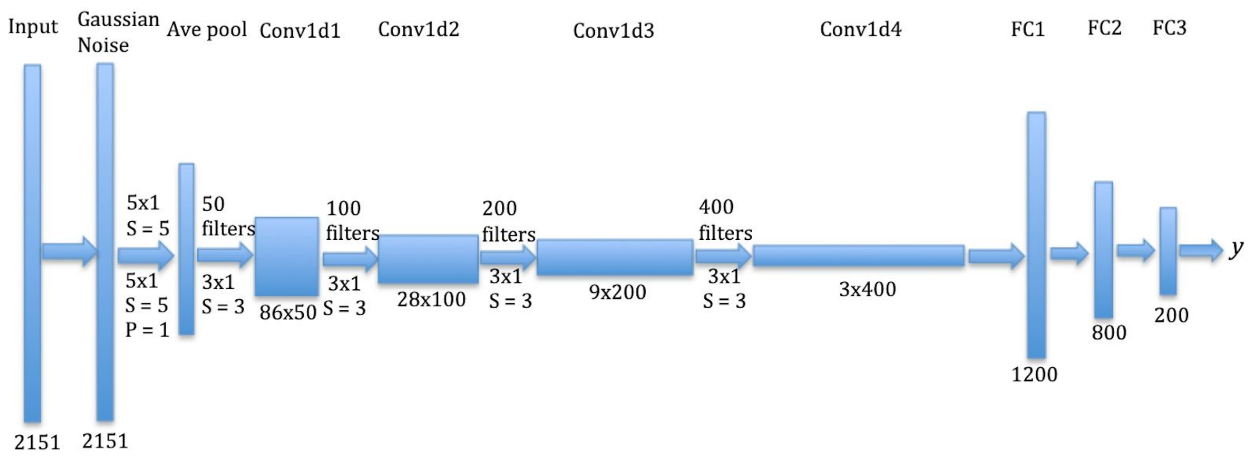

Fig. 3: Convolution neural network (CNN) model for predicting physiological traits

The CNN model consists of four convolution layers for 1 dimensional data (Conv1d) . I added an average pooling layer with the size of 5x1, stride of 5 and padding or 1 and an optional Gaussian Noise layer (see the Refinement section for more information) before the convolution layers. The first convolution layer consists of a 5x1 convolution filter, whereas the remaining layers consist of 3x1 convolution filter. Each convolution layer is following by a batch norm 1D layer and a ReLu activation function. The depths of the filters are 50, 100, 200, and 400, respectively. Finally, the convolution layers are followed by three fully connected layers (FC), each with 1200, 800 and 200 nodes respectively. Each FC is followed by a ReLu activation function and a dropout layer. The model is trained separately for each physiological traits and the output for the model is denoted as *y* which is the predicted value for each physiological trait.

During the training process, after every 10 epochs, the performance of the model will be checked against the validation set and the best model will be recorded. At the end, the model gives the best performance for the validation set is returned. The default parameters are set as the learning rate is 0.01, dropout rate is 0, weight_decay is 0. The number of epochs and the batch size are set to 1000 and 64, respectively. We also implement early stopping policy which is the training is terminated if the model is not improved for the validation set after 200 epochs.

I constructed XGBRegressor model (XGBRegressor.ipynb) which is an implementation of XGBoosting for regression. The XGBRegressor used the default parameters.

Table 2 shows the initial results when using the default parameters of the CNN and XGBRegressor models on the train and validation set. Results show that the CNN model achieved better R^2 scores compared to the XGBRegressor on the validation but lower R^2 scores on train set.

Trait	CNN model		XGBRegressor	
	Train set	Validation set	Train set	Validation set
LMA	0.9399	0.8582	0.9721	0.7757
Narea	0.9482	0.9090	0.9742	0.8804
SPAD	0.9382	0.8582	0.9599	0.8217
Nmass	0.8447	0.6556	0.9404	0.6645
Pmass	0.8628	0.7016	0.9586	0.5775
Vcmax	0.9145	0.7016	0.9681	0.6685
Vcmax25	0.8066	0.6875	0.9485	0.5775

J	0.8992	0.8238	0.8779	0.7328
Photo	0.8422	0.7050	0.9247	0.7772
Cond	0.7636	0.3504	0.9013	0.5073
Average	0.8760	0.7325	0.9425	0.6602

Table 2: Comparison results of the models on the test set using the default parameters

Refinement

For the CNN model, we some hyper-parameters I tried to tune. First, I tried to tune the learning rates from 0.0001, 0.0005, 0.001, 0.005 and 0.01. The results show the 0.0005 is the best learning rate for LMA, Narea, SPAD, Nmass, Pmass, Cond, and Photo. 0.0001 is the best learning rate for Vcmax and J. 0.001 is the best learning rate for Vcmax25. The implementation of the work can be found in PredictingSpectrumTraitsGridSearch.ipynd file.

Second, to reduce the overfitting effect. I added one Gaussian noise layer with standard deviation : $1 / (1 + e^{-x})$ before convolution layers. The *stddev* for the Gaussian noise is to be tuned with 0, 0.01 and 0.05. We also tuned the weight_decay and dropout rate for the Adam optimizer. We performed grid search using the three parameters to train the CNN model for all traits. The grid search space is summarized as follows:

- dropout:
  - Dropout percent used by Adam optimizer
  - Search range: [0., 0.3, 0.5, 0.7]
- weight_decay:
  - L2 regularization used by Adam optimizer
  - Search range: [0., 0.0001, 0.0003, 0.0005]
- stddev :
  - Standard deviation for gaussian noise
  - Search range: [0., 0.01, 0.05]

The table 3 shows the best parameters <learning_rate, dropout, weight_decay, stddev> for the CNN model for each traits and their intermediate results are shown as belows:

Trait	Best parameters	Train set		Validation set	
		Defaults	Optimized	Defaults	Optimized

LMA	0.0005, 0.5, 0.0003, 0.0	0.9399	0.9120	0.8582	0.8803
Narea	0.0005, 0.0, 0.0001, 0.0	0.9482	0.9818	0.9090	0.9184
SPAD	0.0005, 0.0, 0.0001, 0.0	0.9382	0.8509	0.8582	0.8802
Nmass	0.0005, 0.5, 0.0003, 0.0	0.8447	0.9272	0.6556	0.7681
Pmass	0.0005, 0.7, 0.0005, 0.0	0.8628	0.8733	0.7016	0.7314
Vcmax	0.0001, 0.3, 0.0005, 0.0	0.9145	0.9014	0.7757	0.5619
Vcmax25	0.001, 0.3, 0.0003, 0.0	0.8066	0.8617	0.6875	0.7514
J	0.0001, 0.7, 0.0, 0.0	0.8992	0.9004	0.8238	0.8755
Photo	0.0005, 0.5, 0.0003, 0.01	0.8422	0.7741	0.7050	0.6775
Cond	0.0005, 0.7, 0.0003, 0.0]	0.7636	0.7934	0.3504	0.3506
Average		0.8760	0.8776	0.7325	0.7395

Table 3: best parameters and their R^2 scores between using defaults and optimized parameters.

For the XGBRegressor (XGBRegressor.ipynb), I performed GridSearch with parameters defined as bellows:

- learning_rate:
  - Search range: [0.01, 0.05, 0.1]
- max_depth :
  - Maximum depth of a tree
  - Search range: [3, 5, 7,9]
- N_estimators:
  - Number of decision trees
  - Set fix to 1000
- colsample _bytree:
  - Search range: [0.3, 0.5, 0.8]
- Subsample:
  - Set fix to 0.8

I used scikit-learn to perform grid search of the above parameters using 10-fold cross validation, requiring 360 models to be trained (36 configurations x 10 folds) for each spectrum trait. After the grid search, the XGBRegressor achieved average R2 score of 0.6x. The best parameters <learning_rate, max_depth, colsmaple_bytree> for

XGBRegressor each traits and their R^2 scores on the train and validation tests are shown as belows:

Trait	Best parameters	Train set		Validation set	
		Defaults	Optimized	Defaults	Optimized
LMA	0.01, 3, 0.8	0.9721	0.9775	0.7757	0.7794
Narea	0.01, 3, 0.8	0.9742	0.9747	0.8804	0.8782
SPAD	0.05, 3, 0.8	0.9599	0.9959	0.8217	0.8347
Nmass	0.1, 7, 0.5	0.9404	0.9923	0.6645	0.6643
Pmass	0.1, 7, 0.8	0.9586	0.9998	0.5775	0.5241
Vcmax	0.01, 3, 0.8	0.9681	0.9354	0.6685	0.6864
Vcmax25	0.1, 7, 0.8	0.9485	0.9430	0.7328	0.7362
J	0.01, 3, 0.8	0.8779	0.8982	0.7772	0.7880
Photo	0.1, 9, 0.5	0.9247	1.0000	0.5073	0.4411
Cond	0.01, 3, 0.8	0.9013	0.6199	0.1967	0.1710
Average		0.9426	0.9337	0.6602	0.6503

Table 4: best parameters and their R^2 scores between using defaults and optimized parameters.

It is observed that the R2 scores did not improve much on validation set, in worst case, the R^2 scores decreased for Narea, Pmass, Photo and Cond traits. Next section, we will compare the R^2 scores on the test sets.

IV. Results

Model Evaluation and Validation

In this section, we compare non-optimized and optimized models on the validation and test sets.

Table 5 shows the R^2 scores of the CNN model and XGBRegressor using default and optimized parameters on the test set. Results show the R^2 scores are improved for all traits except for Pmass and Vcmax, however the increase and decreasing are not very significant.

Trait	CNN model		XGBRegressor	
	Defaults	Optimized	Defaults	Optimized
LMA	0.8172	0.8566	0.7264	0.7642
Narea	0.8508	0.8702	0.814	0.8169
SPAD	0.8059	0.8437	0.8183	0.8322
Nmass	0.6495	0.6502	0.6611	0.6658
Pmass	0.5303	0.5176	0.4504	0.4366
Vcmax	0.6754	0.7126	0.5407	0.5688
Vcmax25	0.3995	0.4939	0.4625	0.4727
J	0.7315	0.7695	0.7415	0.7427
Photo	0.6525	0.7047	0.6643	0.6544
Cond	0.4525	0.4166	0.5175	0.4948
Average	0.6565	0.6836	0.6397	0.6439

Table 5: The R^2 scores of the CNN model and XGBRegressor using default and optimized parameters on the test set.

Next, I dropped the input rate from 5%, 10%, 15% to 20% and report the R^2 scores on test data sets. The results are shown in the table 6. Result show that both CNN and XGBRegressor still consistency achieved similar R^2 results across different input dropping rate.

Drop input percent	0%	5%	10%	15%	20%
CNN model	0.6836	0.6815	0.6899	0.6890	0.6740

XGBRegressor	0.6439	0.6336	0.6367	0.6337	0.6388

Table 6: Sensitivity analysis

Justification

In this section, I compare the performance of the CNN model, XGBRegressor and PLS regression (PLSRegression.ipynb) on the test set. The summary of results are shown in the table 7.

In general, all the models achieved good results (R^2 scores > 0.65) for predicting LMA, Narea, SPAP, Nmass, Vcmax (except XGBRegressor), and J. Whereas, All models achieved low R^2 scores for predicting Pmass, Vcmax25 and Cond traits. The XGBRegressor is having the lowest for all traits except the Cond.

The CNN model achieved better R^2 scores for predicting SPAD, Pmass, Vcmax25, Photo than the PLS Regression, whereas, the PLS Regression achieved better R^2 scores for predicting LMA, Narea, Nmass, Vcmax and J than The CNN model. Overall, the CNN model have a better average R^2 score than PLS Regression, however the signification is small e.i., 0.6836 vs 0.6771.

Trait	CNN model	XGBRegressor	PLS Regression [4]
LMA	0.8566	0.7642	**0.8698**
Narea	0.8702	0.8169	**0.8812**
SPAD	**0.8437**	0.8322	0.8286
Nmass	0.6502	0.6658	**0.7596**
Pmass	**0.5176**	0.4366	0.4744
Vcmax	0.7126	0.5688	**0.7207**
Vcmax25	**0.4939**	0.4727	0.4905
J	0.7695	0.7427	**0.8049**
Photo	**0.7047**	0.6544	0.6755

Cond	0.4166	**0.4948**	0.4535
Average	0.6836	0.6439	0.6771

Table 7: Final R^2 scores of the CNN model and XGBRegressor compared to PLS regression.

V. Conclusion

Free-Form Visualization

Figure 4 shows the R^2 scores during the whole training epochs on both train and validation sets for the CNN model. To avoid overfitting, the training process is terminated when R^2 scores on the validation set are not improved after 200 epochs. Results show R^2 scores on the training set are improving over epochs for all traits. R^2 scores on the validation set are improving for LMA, Narea, SPAD, J, Photo traits. For other traits, the validation curves are fluctuated and well under the train curves. It explains why the R^2 scores are low on both validation and test set.

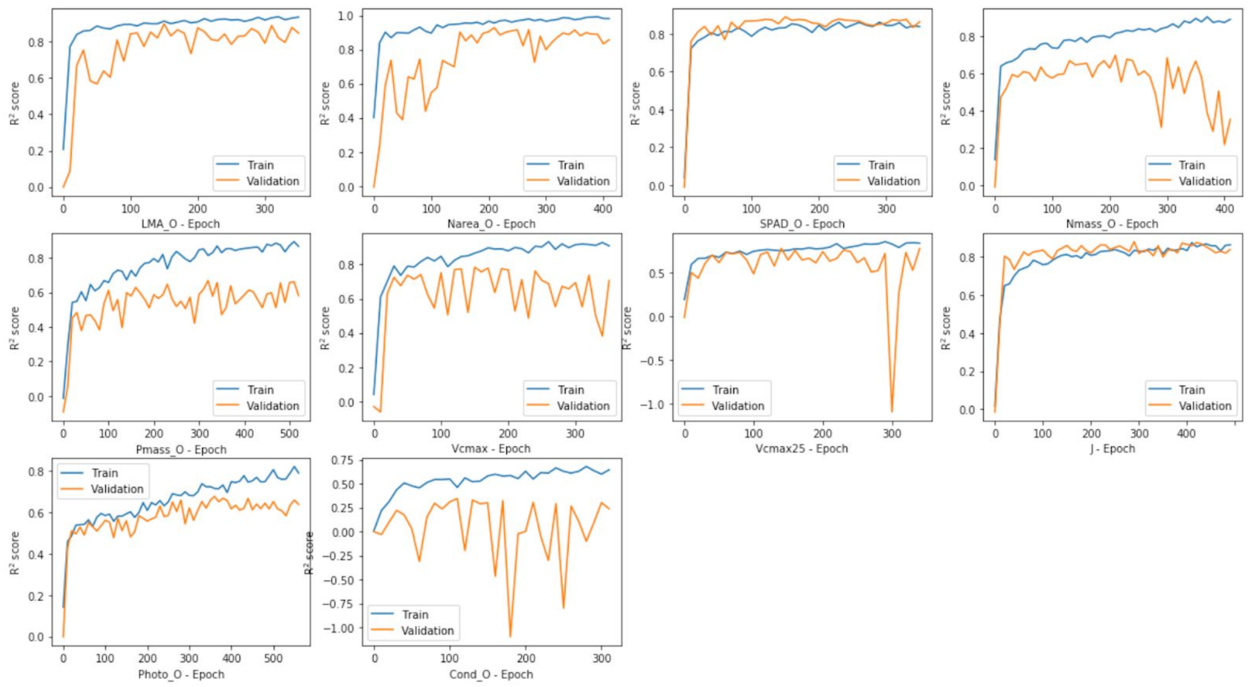

Fig. 4 R^2 scores over epochs for train and validation sets.

Reflection

This project discussed deep learning and machine learning approaches for predicting biochemical and physiological traits using leaf reflectance measurements. I proposed the Convolutional Neural Network (CNN) with for hidden Conv layers, followed by three fully connected layers. I also implemented XGBoosting for regression to solve the problems. I performed hyperparameter tuning using grid search to find the best models for each traits. Lastly, I evaluated the performance of the proposed models against partial least square regression used by [4]. During the project, the most difficult is how the deal with overfitting as the data set is relatively small for CNN model to work well. Different CNN architectures with different configurations are implemented and tested. No models worked well for all the physiological traits. The final model worked well for some physiological traits, but not so good for other traits. The performances of the proposed models are comparable with the existing method. The solutions are not quite fit our expectations for the problem. More investigation needs to be done to improve the models and their performances.

Improvement

There are some aspects of the implementation I would consider as the future work. First, I can used the prediction for good performed traits are the input to to predict under performed traits. For example, I can use the prediction of LMA, Narea, and SPAD as the inputs for predicting Cond trait. Second, I will investigate how to apply multi-task learning to the problem. The multi-task learning is a sub-field of deep learning/machine learning that aims to solve multiple tasks at the same time, by talking the advantage of similarities between different tasks [5].

References

[1] UN Department of Economic and Social Affairs. 2015. World population prospects. The 2015 revision. Key findings and advance tables. New York: United Nations Department of Economic and Social Affairs, Population Division.

[2] Nguyen HT, Lee B-W. 2006. Assessment of rice leaf growth and nitrogen status by hyperspectral canopy reflectance and partial least square regression. European Journal of Agronomy 24, 349–356.

[3] Tilman D, Balzer C, Hill J, Befort BL (2011) Global food demand and the sustainable intensification of agriculture. Proc Natl Acad Sci USA 108: 20260–20264

[4] Viridiana Silva-Perez, Gemma Molero, et. al., Hyperspectral reflectance as a tool to measure biochemical and physiological traits in wheat, Journal of Experimental Botany, 2017

[5] https://www.geeksforgeeks.org/introduction-to-multi-task-learningmtl-for-deep-learning/