Mechanical Engineering Department — ME249, Spring 2021
UC Berkeley — Prof. V. Carey

Project 3. Solar PV Power **Due date: April 8, 2021**

You may team up with a partner for this project. Do not share information or results with other groups.

Files to be used:

<u>Part 1</u>
DS3.1.1Lowflux
P3pcaExample
P3pcaPlot1
DS3.1.2Hiflux
CodeP3.1.2

<u>Part 2</u>
DS3.2.1maxMode
CodeP3.1.2
DS3.2.2multiModePerf

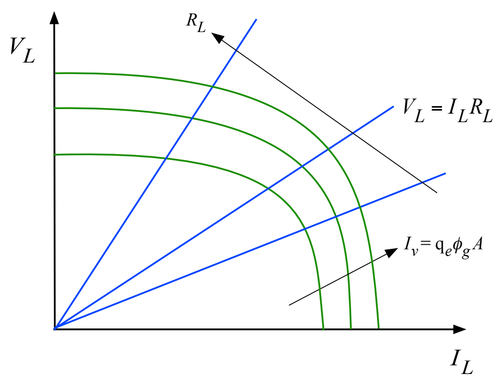

Figure 1. Solar PV panel design (units in mm).

Part 1.
Introduction
Part 1 of this project considers the performance of the solar panel design show above. The panel contains 72 solar cells connected in series, each with an area of 173 cm^2. Performance testing data for this type of unit is provided as a dataset that includes the following performance parameters:

<u>Specified operating parameters:</u>
Outside air temperature, T_{air} (°C)

Incident direct normal solar radiation intensity, I_D (W/m^2)

Load resistance, R_L (Ohms)

Performance (output) parameters:
Panel output voltage to load V_L (V)
Panel power output $\dot{W}$ (W)

Data set DS3.1.1Lowflux (with input data [T_{air}, I_D, R_L]) provided for this project is a collection of data at solar radiation intensities up to the peak incident levels of solar radiation usually possible on the surface of the collector panel (maximum of about 1300 W/m^2). This type of panel is now proposed for use in a system that will have an array of tracking mirrors to reflect additional solar radiation onto the panels. In this system, the panels will receive incident radiation that is as much as twice the direct incident radiation maximum of 1300 W/m^2. Some limited performance data for flux levels above 1300W/m^2 are available, and are provided in Data set DS3.1.2Hiflux. The overall goal is to develop a machine-learning-based model of the performance of the panel based mainly on data at flux levels below 1300 W/m^2, and validate it against data at higher flux levels. The intent is to then use the model to predict performance of the solar PV panel at higher flux levels that should shift the *V*-*I* performance curve to higher current and/or voltage levels.

Task 1.1
The file P3pcaExample contains the example code discussed in an earlier lecture that uses Principal Component Analysis to evaluate the relative importance of input parameters in a data set for a system to be modeled. File P3pcaPlot1 provides code that allows you to look at trends in the data in a 3D scatter plot.

(a) In this task, as a first step, compute the mean and standard deviation for the three input variables [T_{air}, I_D, R_L] involved in data set DS3.1.1Lowflux. Then subtract the mean and divide by the standard deviation to standardize the data.
(b) Next, you are to modify the P3pcaExample code to remove its section where it subtracts the mean from its data, and replace its data set with the standardized data set created from DS3.1.1Lowflux. Also change the names of the variables to suitable choices for the parameters in your data [T_{air}, I_D, R_L].
(c) Then run the program through to the point where the eigenvalues are determined. You do not have to run the portions that create the transformation and transform the data to another space of reduced order.
(d) Summarize the eigenvalues in a table and include it and the 3D P3pcaPlot1 scatter plot of the standardized data created by the code in your summary report. Based on them, provide in your report a discussion of the relative importance of these three variables (are they all important, is one most important, is one of lesser importance than the other two, etc.?)

Task 1.2
CodeP3.1.2 provided with this project is identical to the CodeP2.4r2 file provided earlier (containing updates to cells 2 and 3). Consider this to be a starting-point skeleton code for the activities in this task.

(a) For the original data set DS3.1.1Lowflux, determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the original data and separate it randomly into two data sets: a training set with 2/3rds of the data and a second validation set with 1/3rd of the data.

(c) Substitute the normalized training data into the skeleton code and convert it to a neural network model that can be trained using the training data set. For this first model, use a `keras.sequential` network with having these specs:

- specify a `RandomUniform` initializer (see skeleton code)
- an inlet layer having 6 neurons with `activation=K.elu, input_shape=[3]`
- 3 hidden layers with 8, 16, and 8 neurons
- an outlet layer with 2 neurons with no activation function

Set `activation=K.elu` for all the neurons except the outlet layer, and use the `RMSprop` optimizer, as configured in the skeleton program. Using the `model.fit` routine as configured in the skeleton program is recommended.

(d) Train the neural network model constructed in part (c) using the training data. Try to get the mean absolute error below 0.025 if possible. You can adjust the initialization and/or the learning parameter a bit to try to improve convergence.

(e) Compare the trained model predictions to the training data set, report the mean absolute error for the fit, and create a log-log plot of predicted power output vs. data value power output for each set of data point operating conditions.

(f) Repeat the steps of part (e), comparing the model predictions this time to the normalized validation data. Report the mean absolution error and include the log-log plot specified in (e)for these data in the summary report.

(g) Normalize the limited data for I_D > 1300 W/m^2 provided in data set DS3.1.2Hiflux. Repeat the steps of part (e), comparing the model predictions, this time to the normalized limited data for I_D > 1300 W/m^2. Report the mean absolution error and include the log-log plot specified in (e) for these data in the summary report.

(h) Taking the air temperature to be fixed at 20°C, use the trained model created in this task to create predictions of the solar power output for 4 Ohms < R_L < 8 Ohms and 500 < I_D < 1800 W/m^2, and create a surface plot of the power delivered ($\dot{W}$) to the load as a function of these two variables.

Task 1.3

Repeat steps (a)-(h) in Task 1.2 to construct and train a neural network model with the same specs as Task 1.2, except for the following change to the network design:

Use 4 hidden layers (instead of 3) having 8, 12, 16, and 8 neurons

With this new model, repeats steps (a)-(h), and do this additional step (i):

(i) Compare the results for this task with those for Task 1.2, and assess whether (1) this model better matches the data, and (2) whether there are any signs of overfitting. Summarize your conclusions in your report.

Part 2.
Introduction

In part two of this project you will consider a solar PV system comprised of 4 solar panels of the type described in Part 1. They will be installed in a close-spaced rectangular array, but will be wired with switches that they can be connected so the four are in parallel (mode 1), 2x2 in series/parallel (mode 2), or with the four in series (mode 3). As shown in Fig. 2, these changes combine the *V-I* characteristics of the individual modules to produce three very different overall system *V-I* characteristics.

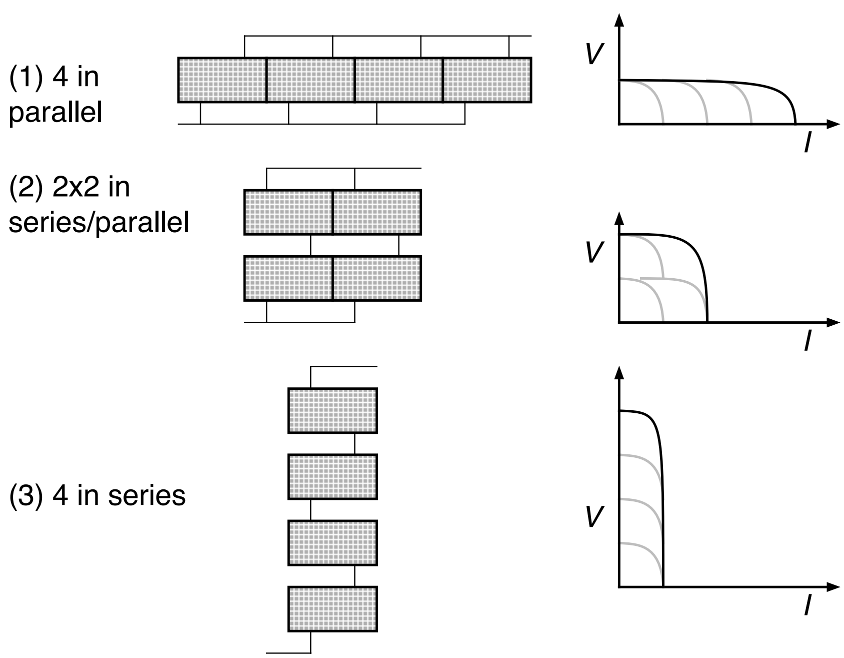

Figure 2. Four PV panel system in different modes.

Performance data for the system is can be obtained at specified values of the following operating parameters:

Outside air temperature, T_{air} (°C)

Incident direct normal solar radiation intensity, I_D (W/m^2)

Load resistance, R_L (Ohms)

The system can tested tested in all three modes to determine which mode (1, 2 or 3) provides the highest power output. The performance data outputs from such a test are:

The mode number M_{max} (1, 2 or 3) providing maximum power
System power output for the maximum mode $\dot{W}_{\max}$ (W)

For this system, the goal is to develop and evaluate a machine-learning based model that can predict which mode will produce the most power for a specified set of operating conditions. Specifically, the objective is to use the model that predicts the mode number M_{max} for maximum performance and output power $\dot{W}_{\max}$ of the solar PV panel for that mode at a given set of operating conditions (T_{air}, I_D, R_L) for model-based control of the PV system. The details of how to set up this model, and a second model to assess its performance, are described in the two tasks below.

Task 2.1

Use the CodeP3.1.2 provided with this project as the starting point skeleton code for the activities in this task.

(a) Data set DS3.2.1maxMode, contains the arrays for the input data [T_{air}, I_D, R_L] and the output parameters [M_{max}, V_L, $\dot{W}_{\max}$] for the 4 panel system depicted in Fig. 2. Determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the normalized DS3.2.1maxMode data and separate it randomly into two data sets: a training set with 3/4ths of the data and a second validation set with 1/4 of the data.

(c) Substitute the normalized training data into the skeleton code and convert the code to a neural network model that can be trained using the training data set. For this model, [T_{air}, I_D, R_L] should be the inputs, and the model should be trained to match data values of [M_{max}, V_L, $\dot{W}_{\max}$]. Here, use a sequential network and make appropriate choices for the number of inputs, the number of hidden layers, and the number of neurons in each layer (including the output layer). Base your choices on your experience in constructing previous models, and make the network complex enough to accurately fit the data, and avoid making it so complex that convergence takes an extreme number of iterations and/or the model overfits the data. Be sure to clearly document all your network design choices in your final report.

(d) Train the neural network model constructed in part (c) using the training data. Try to get the mean absolute error below 0.025 if possible.

(e) Compare the trained model predictions to the training data set, report the mean absolute error for the fit, and create separate log-log plots of predicted power output vs. data value power output, and predicted M_{max} vs. the corresponding data value for each set of data point operating conditions.

(f) Repeat the steps of part (e), comparing the model predictions, this time to the normalized validation data. Report the mean absolution error and include the log-log plots specified in (e) in the summary report.

Task 2.2

(a) Again start with the skeleton code CodeP3.1.2, but here, use data set DS3.2.2multiModePerf, which contains the arrays for the input data $[M, T_{air}, I_D, R_L]$ (which includes the mode number) and the output parameters are the load voltage and power output $[V_L, \dot{W}]$ for the 4 panel system depicted in Fig. 2. Determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the normalized DS3.2.2multiModePerf data and separate it randomly into two data sets: a training set with 3/4ths of the data and a second validation set with 1/4 of the data.

Then, follow the same steps (c) - (f) in Task 2.1 to set up the data for input in the code, create a neural network, train it, and evaluate it against the training data and the randomly sampled validation data. Document the mean absolute error values and include the indicted plots in your report.

(g) The final element of this task is to compare predictions of the two models you have created in Part 2 of this project to compare how well the first model predicts the mode number M_{max} for maximum power output. To do this:
(i) Use the first model to predict M_{max} for the combinations of operating conditions in the table below:

T_{air} (deg, C)	I_D (W/m^2)	R_L (Ohms)
10.0	200	50.
20.0	200	130.
10.0	500	40.
20.0	500	80.
20.0	700	30.
20.	700	55.
10.0	1000	12.
20.0	1000	25.
20.0	1000	39.

In a table, document the values of M_{max} and load power $\dot{W}_{max}$ predicted by the first neural network model for these conditions.

(ii) Then, input each combination of the variables above together with the corresponding predicted M_{max} value <u>rounded to the nearest integer</u>: $[[M_{max,}]_{rounded}, T_{air}, I_D, R_L]$ into the second neural network model, and determine its predicted power output. With the results generated this way, determine the mean absolute difference between the corresponding $\dot{W}_{max}$ value from the first model and the predicted output power from the second model at those conditions.

(iii) Based on your results from the two tasks in Part 2, in your report, summarize your assessment of whether the first neural network model can accurately control the switch setting in the multi-mode 4 PV panel system described in Fig. 2.

Project 3 Tasks to be divided between coworkers:

(1) Data prep and program modifications for Part 1
(2) Training process and computations for comparisons
(3) Plotting and interpretations of results for Part 1
(4) Data prep for Part 2
(3) Program modifications for neural network modeling in Part 2
(5) Plotting and analysis of the results for Part 2
(6) Write-up of the results and conclusions

Deliverables:

Written final report should include:

(1) Written summary of how the work was divided between coworkers.

(2) Assessment of the results and comparisons for the two different neural network designs considered in Part 1.

(3) Plots requested in Parts 1 and 2

(4) An assessment of viability of the first neural network design considered for system control in Part 2.

(5) Your assessments and conclusions should be clearly written with quantitative information to justify them.

(6) A copy of your programs should be attached to the report as an appendix.

Grade will be based on:

(1) thoroughness of documentation of your analysis , especially the logic behind design choices for neural network
(2) accuracy and clarity of interpretation
(3) thoroughness and the documentation of the reasons for your assessments of results.

Summary report due: Thursday April 8, 2021