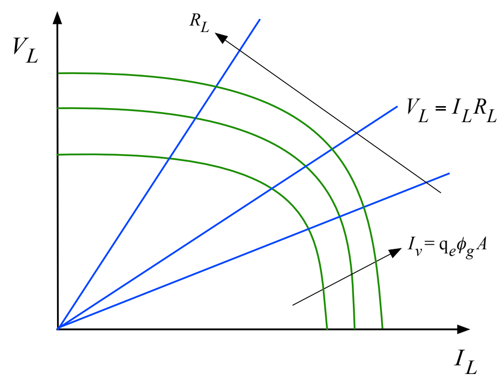
Project 3, Part 1. Solar PV Power **Due date: November 17, 2022**

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

Code Files to be used for Part1:

DS3.1.1LowfluxF22
P3pcaExampleF22
P3pcaPlot1F22
DS3.1.2HifluxF22
CodeP3.1.2F22



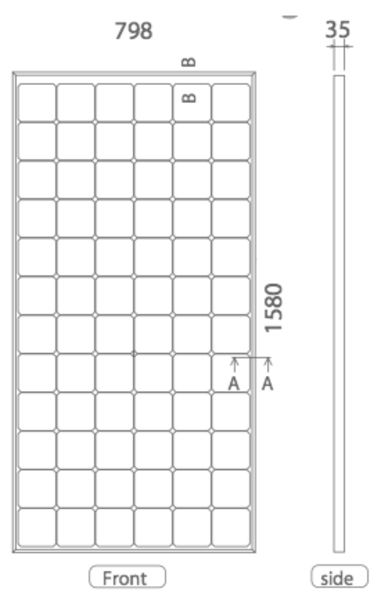


Figure 1. Solar PV panel design (units in mm).

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 test data for this type of unit is provided as a dataset that includes the following performance parameters:

Specified operating parameters:

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 DS3.1.1LowfluxF22 (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 50% more than the direct incident radiation maximum of 1300 W/m^2. Some limited performance data for flux levels above 1300 W/m^2 are available, and are provided in data set DS3.1.2HifluxF22. 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 P3pcaExampleF22 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 P3pcaPlot1F22 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 DS3.1.1LowfluxF22. Then subtract the mean and divide by the standard deviation to standardize the data set.
(b) Next, you are to modify the P3pcaExampleF22 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 DS3.1.1LowfluxF22. 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 P3pcaPlot1F22 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
CodeP3.1.2F22 provided with this project is similar to the CodeP2.4F22 file provided for Project 2 earlier. Consider this to be a starting-point skeleton code for the activities in this task.

(a) For the original data set DS3.1.1LowfluxF22, determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.
(b) Take the normalized original data created in part (a) 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 set 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 DS3.1.2HifluxF22 in the same way as the DS3.1.1LowfluxF22 data. 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 changes 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) in Task 1.2, 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.

Project 3, Part 1 tasks to be divided between coworkers:

(1) Data prepand program modifications for Part 1
(2) PCA - analysis of eigenvalues
(3) Training process and computations for comparisons
(4) Plotting and interpretations of results for Part 1
(5) Summary 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) Summary of PCA eigenvalue findings and interpretation

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

(4) Plots requested in Part 1 (should not be in the Appendx, be sure to label axes with units)

(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 project report due: **November 17, 2022**