Angus Fursdon-Welsh
DTSC-691 Capstone Project
12/14/24

Project Overview

Surfactant Recognition Machine Learning Model

Project Objective and Scope

Surfactants are an important group of molecules which are used across a wide variety of industries; including oilfield, agriculture and personal care. Identifying molecules which have surface activity is important as researchers seek to make advances in these industries. This model is trained on a set of known surfactant structures, and is able to identify other surfactant molecules which are not part of the training set, purely based on their structure. This means that rather than spending a great deal of time and resources on testing a large number of molecules in the lab, the researcher can prioritise the molecules most likely to succeed based on the recommendations of this model. This project will focus on surfactants since these molecules have these specific structural properties which I believe the model will be able to recognise. However, it is possible that other molecule types could be tried if the scope of this project was extended.

Data Acquisition

PubChem 'is the world's largest collection of freely accessible chemical information'[1] and has details and properties for almost 19 million chemical compounds. I used this remarkable chemical data trove to obtain datasets for surfactant and non surfactant molecules, including molecular structures, which I then used to train the model.

Exploratory Data Analysis

I used correlation matrices and plots to explore if there were any variables which correlated well with which molecules were surfactants and which weren't. Since none of the variables showed a correlation greater than 0.5, this reinforced that my model would be valuable since it would make use of the molecular structure of the molecule to determine whether it was likely to have surface active properties or not. This analysis can be found in my code file named 'Exploratory Data Analysis' in my 'Project Completion folder on Google Drive.

[1] https://pubchem.ncbi.nlm.nih.gov/

Data Preparation and Cleaning

I had to merge data files and remove duplicates to create my surfactant and non surfactant datasets. I then combined them and added a Boolean column to denote whether each row was a surfactant or a non surfactant. Once I realised that having the class of non surfactants be three orders of magnitude greater than the class for surfactants was going to make my model untenable, I randomly selected rows from the non surfactant dataset so that it was equal in size to the surfactant dataset. These datasets then allowed me to return to PubChem and use the list of unique CID (Compound Identifier) values to download molecular structure images for both classes; surfactant and non surfactant. This data preparation and cleaning can be found in my code file named 'Data Preparation and Cleaning' in my 'Project Completion' folder on Google Drive.

Model Training

The intent was to create a model which can differentiate between a surfactant and non surfactant. I chose to select a Keras Sequential model, a type of Convolution Neural Network (CNN), to construct my machine learning model. My rationale for selecting this was that "Convolutional Neural Networks are the most widely used and effective algorithms for image recognition."[2]

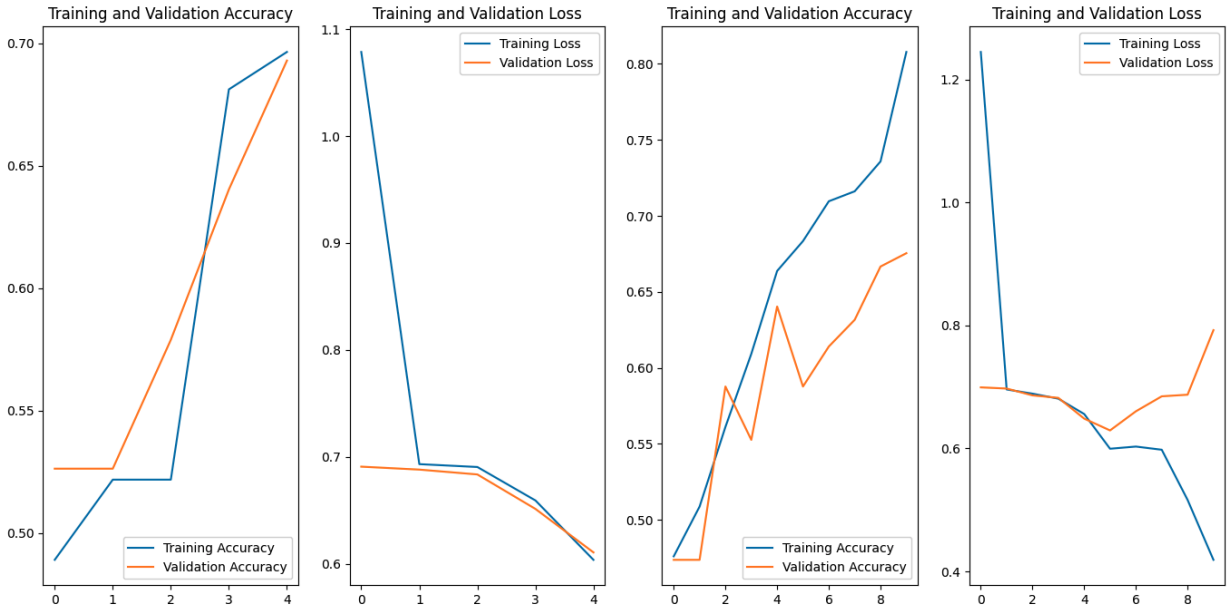
Once I had my prepped and cleaned the datasets (see previous section), I first defined the batch size, and image dimensions for the images I would be using for the model. Next, I split the dataset; 80% training, 20% validation. As outlined above, the data was split into two classes; surfactants and non surfactants. The dataset was then configured for performance and standardised, before being passed to the model. Finally, the Keras Sequential model was created, compiled and then trained using the training dataset. The accuracy and loss of both the training and validation data sets was used to evaluate and optimise the model (see the following section). The model training can be found 'UI and Model'>'models'>'Code' in my 'Project Completion' folder on Google Drive.

Model Evaluation

I evaluated and optimised the performance of my model by adjusting the number of epochs. As can be seen from the plots below, I ran the model with four different epoch values; 5, 10, 15 and 20. At 5 epochs, the accuracy of the validation data set was still climbing, as can be seen by comparing it with 10 epochs. The 15 and 20 epoch graphs show the validation accuracy is not increasing much beyond 10 epochs, but the validation loss is increasing significantly after 10 epochs, relative to the training loss. This is indicative of overfitting, which I countered by limiting the number of epochs to train the model to 10.

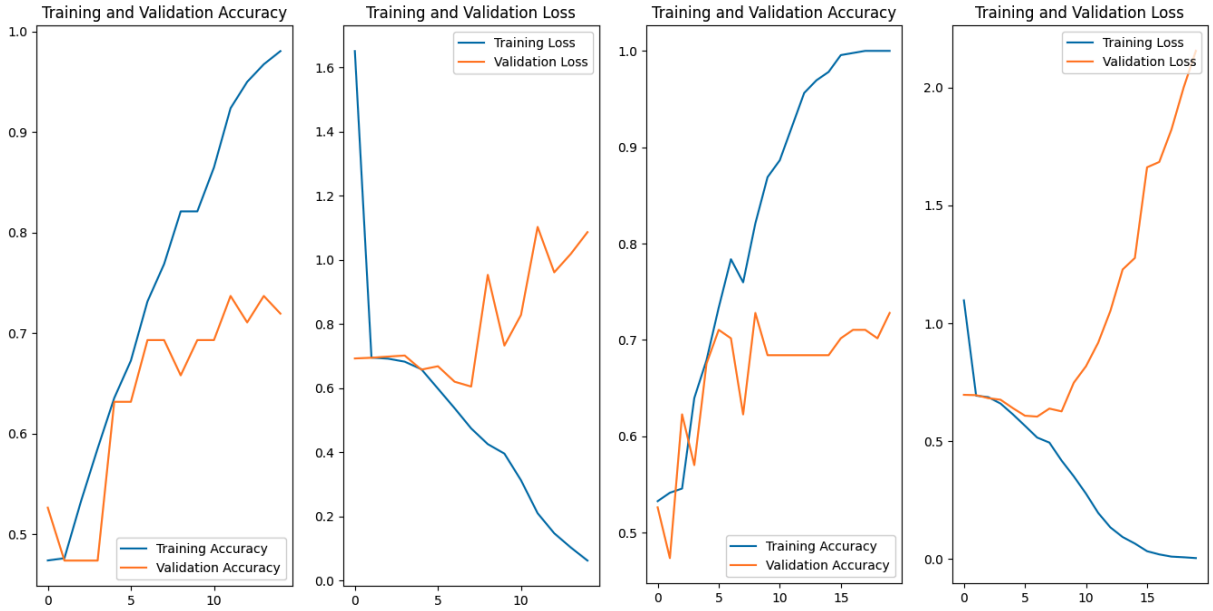
---

[2]https://medium.com/@mansi89mahi/5-best-machine-learning-algorithms-4-image-recognition-ab0eee5e2931



5 epochs

10 epochs



15 epochs

20 epochs

User Interface

I used Flask to create the user interface (UI) to deploy my model. This UI includes a biographical ‘About Me’ page, including a picture file; a Resume page, including a PDF of my resume; and a menu for ‘General Projects’ which includes a specific page for this project. The code for the UI can be found in the ‘UI and Model folder in my ‘Project Completion’ folder on Google Drive.

References

The following resources each played some role in allowing me to complete my project in its current form, even if the subject matter was ultimately not used in the final iteration (i.e. ‘balancing classes’.)

Image Classification:

- https://medium.com/@mansi89mahi/5-best-machine-learning-algorithms-4-image-recognition-ab0eee5e2931
- https://www.tensorflow.org/tutorials/images/classification

Flask:

- https://www.youtube.com/watch?v=dam0GPOAvVI
- https://www.geeksforgeeks.org/deploy-machine-learning-model-using-flask/
- https://www.youtube.com/watch?v=2LqrfEzuIMk
- https://www.youtube.com/watch?v=NtNl97LlpOk
- https://www.youtube.com/watch?v=ksCYBZIGShI
- https://medium.com/@pg2196577/building-an-image-classifier-with-tensorflow-and-flask-a-beginners-journey-8f76ecd90603
- https://geekpython.in/flask-app-for-image-recognition#google_vignette

Balancing classes:

- https://www.tensorflow.org/tutorials/structured_data/imbalanced_data
- https://docs.google.com/document/d/1gSk3l087Ed7woiQZC9o0d1t_esx0Fg_8U95Kw7qNRqg/edit?usp=sharing
- https://stackoverflow.com/questions/66016844/imbalanced-image-dataset-tensorflow2
- https://www.geeksforgeeks.org/how-to-handle-class-imbalance-in-tensorflow/
- https://towardsdatascience.com/dealing-with-imbalanced-data-in-tensorflow-class-weights-60f876911f99
- https://www.youtube.com/watch?v=i3RMlrx4ol4&list=PLTuKYqpidPXYcVqmcV3E6jWtTV_AYA2fH

Training and validation loss:

- https://www.geeksforgeeks.org/training-and-validation-loss-in-deep-learning/

Chrome socket issue:

- https://www.reddit.com/r/flask/comments/ttawkw/access_to_127001_was_denied/?rdt=45008