# PROJECT 1

## PHYTOPLANKTON NEURAL NET PREDICTION

### Preprocessing

Before feeding the data to a neural network, it is important that the data is clean and void of null values. Also, the data needs to be numerical for the algorithm to process it. Hence, the data needs to go through preprocessing.

Step 1:

The raw data is in excel format, so we can make use of the **read_excel** method in the **pandas** library to read the data to make it suitable for our goal. In my case, I was able to read the specific columns that we know are important to predict if values are above the threshold. Also, I made use of slicing to pick the specific rows that are important as well as skipping the unimportant heading and some data that was written at the bottom of the worksheet.

Step 2:

The next step was to clean the data to eliminate unwanted text in the dataset as well as converting the integer values that are represented as strings. To do this, I made use of the pandas replace method (to replace text like "Unable to analyse, too much sediment" with -1 and "ND" with 0) and the fillna method to replace all null values with -1.

Step 3:

The next step was to create a threshold column which acts as the class/label column for this binary classification task. To do this, I made use of numpy's where method to compare the thresholds in each column to create the new column where a value of 1 is given if any of the columns has a value above a threshold specified for that column and 0 if none of the columns that intersect that row have a value above a threshold.

Step 4:

At this point, the data is ready to be processed in the feed forward neural net. However, to get an optimal result, it is essential to normalize all data (in this case, I made use of the standard scaler from sklearn). After this is done, the data is normalized to have values from 0 to 1.

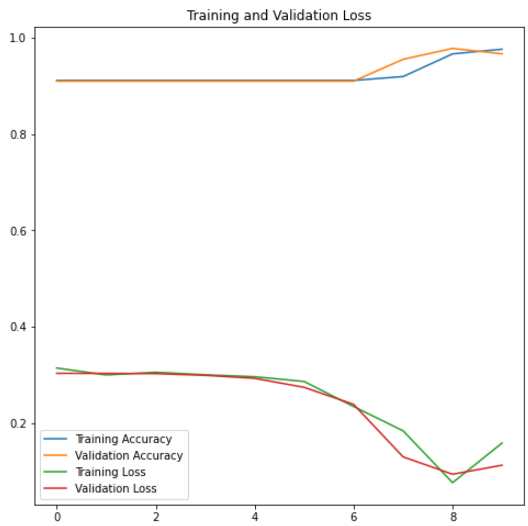
Before, architecting the neural network, I made use of a baseline algorithm that uses the most frequent occurring class to carry out classification. The implementation was done with sklearn's dummy classifier and setting the strategy to "most frequent". I was able to establish a baseline of 91.04%

**Neural Network:**

I made use of keras library to carry out the binary classification task.

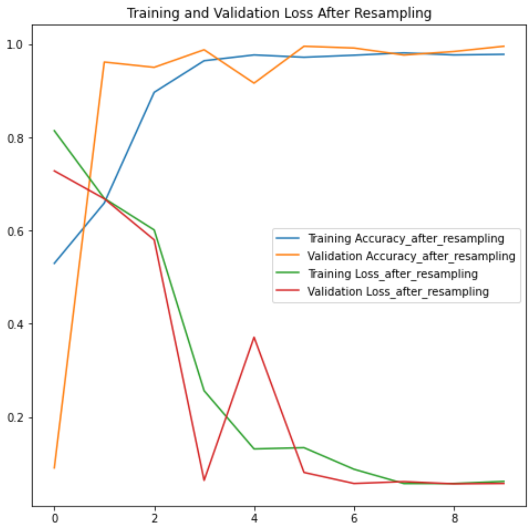
Three architectural modifications:

1. 1 hidden layer(100 units); relu activation function in both the hidden layer and the output layer; adam optimizer ; no dropout; loss function as binary_crossentropy; uniform kernel initialiser Accuracy: 90.6%
2. 2 hidden layers(250 and 15 units respectively); tanh activation function in the hidden layers and the output layer; SGD optimizer ; no dropout; loss function as binary_crossentropy; uniform kernel initialiser
   Accuracy: 94.2%
3. 4 hidden layers(100; 15; 8; and 4 units respectively);sigmoid activation functions in the first two hidden layers and tanh activation function in the next two hidden layers and relu in the output layer; adam optimizer ; 25% dropout after the second layer and 20% dropout after the fourth; loss function as binary_crossentropy; uniform kernel initialiser
   Accuracy: 96.6%




Above is the plot of training loss over 10 epochs using the 3rd architecture above.

After this, I made use of resampling from sklearn's SMOT to oversample the data since it seemed to be favoring one class over the other during training. Thus yielded an accuracy of 99.6%. Below is the result of the training plot after the resampling had been done.




- Specify the accuracy you achieved across 3 architectural modifications (e.g. different numbers of layers, different hyperparameters, etc.)
- Why do you think your accuracy is not higher / lower?

DATA ANALYSIS AND VISUALISATION 1

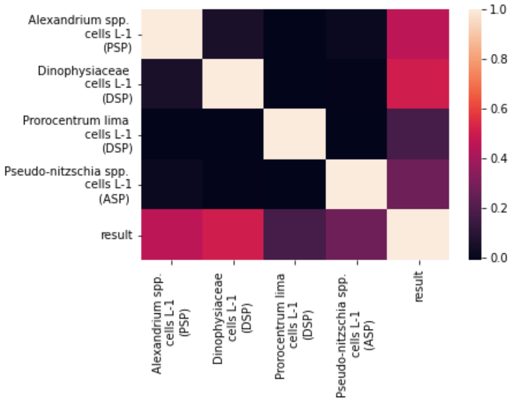
- What effect does the optimisation function have on network performance?
- What happens if you include more than 4 layers?
- What is the effect of the data size on your accuracy?

Solution to Questions above:

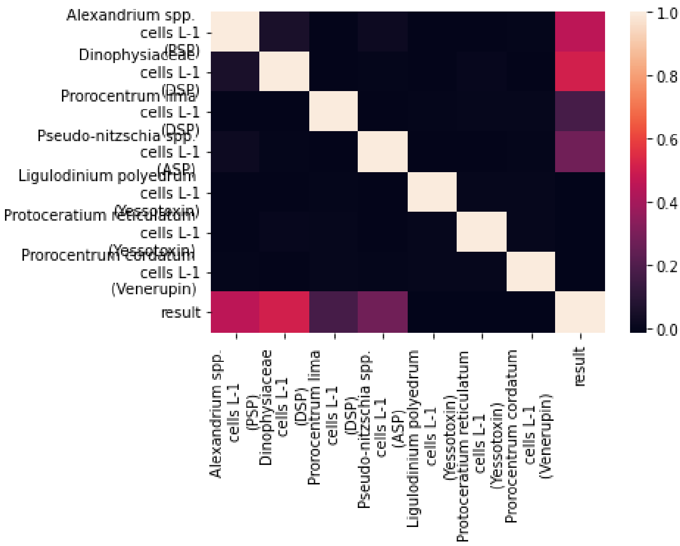
1. Architecture 1: 90.6% Architecture 2: 94.2%
   Architecture 3: 96.6%
2. I think the reason for the very high accuracy is because the features selected were the determinants for the creation of the class column meaning they would have relatively high correlations to the class column.
3. Optimization functions help to reduce loss by adjusting weights and learning rates.
4. When I made use of 5 layers, I got a lower accuracy of about 93%
5. With more data, the neural net will have more patterns to recognize and learn to give a better accuracy as well as not overfit during the learning phase.

More Graphical Plots:

Below are two correlation plots which show that the selected features for this classification task were the most optimal.




In this, we can see that the four features above have a good correlation with the result column.




In this, we can see that the four features selected for the classification task have better correlation with the result column than the other three features added for comparison.

# PROJECT 2

IMAGE CLASSIFICATION FOR VEHICLES.

This classification task was done with Tensorflow and the Keras API.

In mine, I attempted four different architectures, two with data augmentation and two without.

**First Architecture:**

All images were normalized by dividing each pixel by 255. This makes the pixel intensities in the image matrices have a value from 0 to 1. Afterwards, they are passed through a convolution layer with a relu activation function and a pooling layer. In this architecture, I went with the Max pooling which makes use of the highest value in a portion of the intensity pixels to represent the portion.This helps to reduce the size of the colvolved features(pixel values).

Max pooling:

| | | | |
|---|---|---|---|
| 20 | 23 | 88 | 23 |
| 234 | 11 | 111 | 76 |
| 27 | 8 | 2 | 81 |
| 6 | 99 | 244 | 0 |

| | |
|---|---|
| 234 | 111 |
| 99 | 244 |

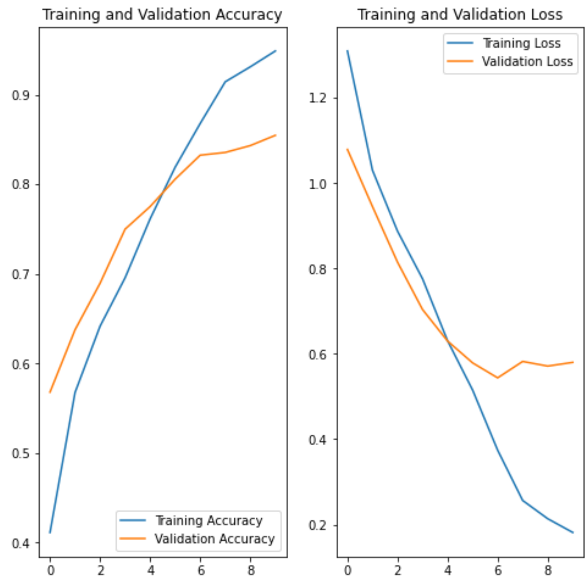
The above image describes how max pooling is done.

I repeated the convolution-pooling layers two more times before flattening the result into a 1 dimensional array for training the model.

Summary:

3 convolution layers (relu activation function) & Max pooling.

This was done over 10 epochs and gave an accuracy of 85.47%




**Second Architecture:**

Summary:

3 convolution layers (relu activation function) & Max pooling.

One dropout layer after the last convolution-pooling pair (0.2 dropout)

This was done over 15 epochs and gave an accuracy of 63.23%

Also, data augmentation was implemented in this architecture.


**Third Architecture:**

Summary:

4 convolution layers (relu activation function) & Average pooling.

Three dropout layers after the last three convolution-pooling pair (0.2 dropouts each)

This was done over 20 epochs and gave an accuracy of 64.9%

Also, data augmentation was implemented in this architecture.

Average Pooling:

In this case, instead of picking the maximum value in each portion, the algorithm finds the average of the pixel intensities instead.

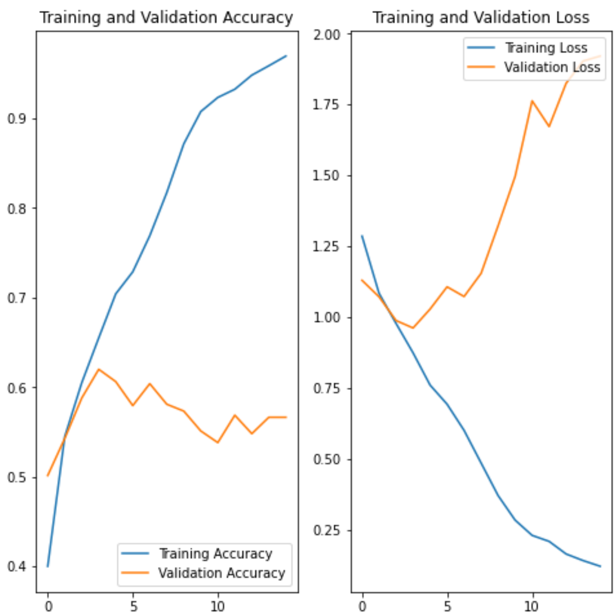

**Fourth Architecture:**

Summary:

3 convolution layers (relu activation function) & Average pooling.

No dropout layer and no data augmentation.

This was done over 15 epochs and gave an accuracy of 56.65%




• How long does the network need to train until reaching an accuracy of 95% (or does it not reach this level at all)?

After trying different architectures, the closest accuracy to 95% I was able to get was 85.47%. However, this was without any data augmentation. It took 29 minutes to train.

• What is the tradeoff between using many layers (i.e. having a "deeper" network) and accuracy? And layers and time?

From my architectures tried, having many extra layers of network might not necessarily improve the performance too much. In my case, when I increased the layers, I got a slightly better accuracy but it trained in a longer amount of time.

• What is the effect of changing the pooling mechanism, e.g. average vs max?

Changing the pooling mechanism from Max to Average did not have too much of an impact as the accuracy with similar architecture and data augmentation gave almost the same accuracy for both Average and Max pooling. But I think this is because of the use case we have mainly bright images. Average pooling normally smooths out images and reduces how well the sharp portions of the image are represented while max pooling displays the brightest portions of the image, and is usually evident with darker images, so i'll say they won't have too much of an effect in this scenario, and this shows in my results because they have very similar accuracies for similar architectures.

As a follow-on part, collect your own dataset of images containing the four object categories above. Make sure that they occur in different context, e.g. close-up, far-away, in a busy visual context, in an isolated image, etc. It is up to you how you collect these images- you can either take photos yourself or collect images from the internet. You should collect 20 images and copy these into your report, so I can see them.

• How well does your network do at classifying these images?

• Does fine-tuning make a difference?

The first model does pretty well in predicting the correct classes. In my case, the model was correct in more than half of the test samples.

Fine tuning and changing hyperparameters did not do so much for improvement of prediction, at least for the ones I tried.





AIR FRANCE








SCHOOL BUS




Extra challenge - integrate explainability methods, such as tf-explain (https://github.com/ sicara/tf-explain ) to visualise how your model makes predictions for a small set of example images.

I integrated the explainability method from the tf-explain method (Grad Cam) which is supposed to give a heat map of what the model focuses on while training. Unfortunately, my model wasn't doing too well in that aspect.

Below is an example of a grad cam heatmap for a bus.

The bus is at the left side of the image but the heatmap is scattered all over the image.

# PROJECT 3

**ENERGY AND POLICY CONSIDERATION FOR DEEP LEARNING IN NLP**

**INTRODUCTION**

Artificial intelligence plays a vital role in shaping the universe and its benefits are significant in almost all aspects of our lives. However, AI has drawbacks that hose its impressive credential. Professionals and researchers in AI should be aware of the numerous ethical challenges faced by this promising field. Attempts have been made by AI professionals and academics to derive benefits of AI and as much as possible while reducing ethical issues. One of such attempts is the conversation within the AI community to ensure that AI technologies are utilized for ethically good purposes through the campaign "AI for Good" (Berendt,2019) Additionally, a few researchers have called attention to, examined, and proffered answers for some disturbing moral issues in the field.

In the paper: "Energy and Policy Consideration for Deep Learning in NLP", Strubbel , at al raised relevant issues concerning AI. The paper evaluated the monetary and environmental cost of accomplishing a result in ML. According to them, a transformer discharges fundamentally, more carbon than five times the lifetime emission of an average fuel car. This was done by reviewing the financial cost and carbon emission of four NLP models: Transformers, Elmo, BERT, and GPT-2

The paper also talked about the cost of acquiring computational infrastructure leading to an unfair distribution of computational resources between academics and AI industry experts as most efficient AI resources are too expensive to be afforded by researchers.

Also, the R&D cost which is not the same as the cost of training a model usually duplicates the cost of these models. Research and development expect retraining to assess various designs, variations, and hyperparameters. This cost was evaluated by concentrating on the logs of all training needed to foster the Linguistically Informed Self Attention (LISA) model.

Definitively, the team recommended that Industry and academia ought to advance exploration on efficient models and hardware that require less energy. Again, the software creators should report training time and computational assets used to foster a model as this will empower an immediate correlation across models. They also recommended even-handed access to AI resources by both researchers and industry experts.

While Strubbel et al may have uncovered and addressed the issue of financial cost and energy usage in NPL, other areas of applied AI face similar challenges and there is a need for policy consideration and researchers should be careful. The paper finds application in almost all areas of AI, from computer vision, natural language processing, data center management to machine learning as they come with CO2 price tags. (Flip A et al, 2018).

**UNFAIR AND UNEVEN ACCESS TO COMPUTATIONAL RESOURCES**

The unfair distribution of assets is not found only in the area of acquisition infrastructure but also in human capital. There is a gap between talent demand and supply in AI. Because of this gap, qualified data scientist places an exorbitant cost on their abilities which makes it hard for some research institution to hire them. (Metz, 2018). It has even been reported that tech giants are employing academic staff away from research institutions (Gofman & Jin, 2019)

**DATA CENTER MANAGEMENT**

Another area that the research has found application is in the areas of data center management. It is on record that energy consumed by data centers around the world is more than the energy consumption of some countries ( Andrae, 2015). Data centers owned and run by some tech giants like Amazon, Alphabet, Apple, Facebook, and Microsoft are far bigger than what oil companies like ExxonMobil, Totals, and Shell has. This is expected to increase as the demand for data increases due to advancement in AI and may lead to an energy explosion (Koomey, 2011)

The carbon emission of the overall data centers is estimated at 0.3% while that of the entire AI sector is more than 2%.

Luckily, researchers and specialists who comprehend the impact of such discharge are attempting to decrease the carbon impression by building more proficient foundations. Figuring processes are smoothed out, and more sustainable types of energy are utilized in data centers. Also, tech giants acquire a large portion of the necessary energy from sustainable or carbon credit offset resources, however, the energy needed of these models is  subject of concern since:

I.) Energy is not presently gotten from carbon nonpartisan sources in numerous areas like most places in Africa.

II.) When renewable energy is available, it is restricted to the gear we need to store them.

**CONCLUSION**

As artificial intelligence is gaining momentum in almost all sectors, it is important that policymakers, researchers, and engineers acknowledge the ethical issues related to this development especially as it concerns energy usage and infrastructural cost.

## REFERENCES

Andrea, A. & Edler, T (2015). Challenges 6 , 117-157 ( 2015)

CDEI,(2021) Interim Report: Review into bias in Algorithmic Decision Making , CDEI London

CEM Dilmegani (2021): Bias in AI : What it is, Types & Examples of Bias & Tools to fix it.

Filipp A. et al. (2018): Artificial Intelligence & Human Rights. Opportunities and an risk.
Berkman Klian Center Research Plublication , No 2018-6

Gofman and Jin(2019) : Facebook Friendships Expose Sexual orientation. 2020

Koomey , J G  et al IEE Ann. Hist.comp 33, 46- 54 (2011)

Strubell, E, Ganesh, A and McCallum, A.(2019) Energy and Policy Consideration for Deep Learning in NLP ,ar:1906.02243