

Assignment 3

Due date : Tuesday 3rd May by 11pm.

There are four files for this assignment. Assignment3 - Pokemon.pdf, pokemoninfo.zip, pokemonimages.zip, evalimages.zip. This assignment is worth 20% of your overall.

All files are available on Moodle. Complete all work using Jupyter notebook, typing any answers to questions in a Markdown cell type. You are to build all models using TensorFlow in this assignment, you can use others desired tools from numpy, sklearn, pandas as needed but the actual model building must be using TensorFlow.

Your submission will consist of the Jupyter workbook and a PDF file summarising the system. Do not zip the files together, upload them separately. Ensure when you are uploading your work you only upload your own work and not any dataset. You can complete this in either one or two notebooks. Put your Name and ID at the top of the workbook.

This is **NOT** a group assignment. You must complete the assignment **individually**.

Pokémon

Pokémon also known as Pocket Monsters in Japan, is a Japanese media franchise managed by the Pokémon Company, a company founded and with shares divided between Nintendo, Game Freak, and Creatures.

Pokémon RPG games began with Red and Green (later released as Red/Blue outside Japan) in 1996 and in 2019 the latest versions Sword and Shield were released. The Pokémon RPG games are developed by Game Freak. Each game has added new Pokémon and they are split by Generation. For example, Red and Blue are Generation 1 with 151 total Pokémon in the game. Sword and Shield are known as Generation 8.

1 Which Pokémon are Legendary

Each Pokémon has base stats that determine it's features such as HP, Attack, Defense, etc. In each generation some of the Pokémon are classified as *Legendary*. Legendary Pokémon are a group of incredibly rare and often very powerful Pokémon, generally featured prominently in the legends and myths of the Pokémon world.

The question: Is it possible to build a Machine Learning Classifier that can determine if a Pokémon is Legendary based on its stats.

For this exercise, the file `PokemonDataSet.csv` will be your training data. This is a collection of all the Pokémon in the first 7 generations. The file `PokemonGen8.csv` will be your test data - this file is a collection of the 400 Pokémon that are obtainable in Sword and Shield (there will be some Pokémon in both lists and it only includes the Pokémon before the expansion packs)

1. Only use the stats HP, Attack, Defense, Sp_Atk, Sp_Def, Speed as your features.
2. You are to build a Neural Network to classify if Pokémon are legendary or not (you do not need to do train/test split remember) .
3. When building the model, you may want to use a scaler on the data - this could be build into the model or done separately. Use of a scalar or not is a hyperparameter....
4. Don't forget about validation sets and ordering in a validation set.

5. Try different models and eventually choose one - this should be relatively quick as it is not a very large dataset and only a small number of features so trying multiple hyperparameters should not be an issue.
6. When done print the loss vs val.loss graphs.
7. From your training set - print out the names of which Pokemon were false positive for Legendary and then separately the Pokemon that were false negatives for Legendary. Maybe the false negatives are just weak Legendaries and maybe the false positives are just very strong Pokémon.
8. For the test data
 - Print the score.
 - Print the confusion matrix.
 - Print the names of the Pokemon that were identified as Legendary. Does it correctly predict Zacian, Zamazenta and Eternatus as the only Legendaries in the Gen 8 list?
9. Write a summary of your work and include this in the overall PDF - why did you choose the parameters you did - what is your overall assessment?

2 Who's that Pokémon

You have been given a collection of 6,796 images of different Pokémon from Generation 1. 149 of the first 151 Pokémon are included (The Nidorans are missing but the evolutions are there). Your goal is to build a model that can identify the Pokémon in `evalimages.zip`.

Note: You are to use things I talked about in class or had in workbooks - don't try transfer learning (although this may improve results) or any online tools that help with Machine Learning. If you see anything about VGG, InceptionModules or things I never mentioned - ignore them! Your model may not be very good or it may be great - doing the process is the main thing.

Also: This is going to take longer to run, get coffee - do other things while the model is fitting (study?). You are going to want to build a few models but sitting there staring at the progress bar is not going to do much

They are images, so it should be obvious what type of model to build.

Take 10% of the images (selected randomly) as a test set for evaluation of your model in the usual manner, seeding anything with your id number excluding the s and 0s at the front.

Ensure when loading the images, all file types are included, there are at least jpg, png and jpeg file extensions in the set. You can use whatever method you like to load the images and get the corresponding classes, although https://www.tensorflow.org/api_docs/python/tf/keras/utils/image_dataset_from_directory may help if you wish to investigate how that method works - but it is not a requirement to use this.

You will again need to build multiple models with different hyperparameters and checking against overfitting. Check with the val vs val.loss graphs.

Do a usual evaluation when you've chosen which model to use (confusion, precision/recall etc). What Pokémon is your model good at predicting, what ones is it bad at predicting?

See if it correctly identifies the pictures in `evalimages.zip`. I'm aware one of the pictures is Mimikyu but it would be happy if your model thinks it is actually a Pikachu.

Finally, write a summary of your work and include in the PDF. This summary should be longer than the previous one, but at most one page. Tell me why you made the decisions you made and include the answers to the last two parts in the document.