Herman Franclin Tesso Tassang

1 Question 1: Method we can explore to train unbalanced data set of green turtles.

EasyEnsemble.

A common problem in real life applications of deep learning based classifiers is that some classes have a significantly higher number of examples in the training set than other classes like it is the case for green turtles behaviours' distribution where the "sleeping behaviour" is highly predominant over the rest, in particular over the "feeding behaviour' which is underrepresented.This difference is referred to as class imbalance and has been comprehensively studied in deep learning. A way for addressing this issue is to operate on training set[1] and change its class distribution, and one of the most straightforward and common approach is "***undersampling***, which consists to randomly remove examples from majority classes until all classes have the same number of examples.It appears to be an efficient strategy to deal with unbalanced dataset.However, the drawback of this method is that it throws away many potentially useful data.In the following, I propose to explore a method named ***EasyEnsemble*** introduced in paper "*Explanatory undersampling[2] for class-imbalance learning"* by *Xu-Ying Liu et al.* to explore the majority class examples ignored by *"undersampling method.*
In a few word, *EasyEnsemble*[4] is some kind of ensemble learning technique designed to address class imbalanced in classification tasks, particularly in scenarios where the minority class is heavily underrepresented (e.g "feeding behaviour" in our case). It works by creating multiple balanced subsets of the original dataset, training a base classifier on each subset, and then aggregating their predictions to make the final prediction.Below is a simplify workflow we can follow for our particular problem.

Step 1. subset creation: randomly select a subset of examples from majority class ("resting behaviour") while maintaining the balance across all classes .

Step 2. Classifier training: train a 1D CNN on each balanced subset created in *step 1.*

step 3. used each trained 1D CNN to predict the class labels for all examples (intances) in the dataset.

Step 4. aggregating : aggregate the predictions of all trained classifiers to make the final prediction for each examples(we can use majority voting).

EasyEnsemble can effectively address unbalanced dataset of green turtles and improve the performance the behaviour classification task.The evaluation can be done using standard evaluation metrics for multi-class classification (Accuracy, precision,...etc).

2 Question 2 : Loss function for classification and unbalanced dataset

2.1 a) Loss function explain.

Let's say we are working on any problem and you have trained a machine learning model on the dataset and are ready to use it. But how can we be sure that this model will give the optimum result?; Is there a metric that will help evaluate our model on the dataset? Yes, here *Loss function* come into play in ML or Deep learning.In the following paragraph, we provide a brief explanation of what loss function is , in Deep learning.
In mathematical optimization and decision theory, a loss function is a mathematical function that quantifies the difference between predicted and actual values in a machine learning model.It measures the model's performance and guides how well it fits the data.In simple terms, The *loss function* is a method of evaluating how well your algorithm is modeling your dataset.

In multi-class classifation task, the commonly used loss function is the *Categorical Cross Entropy loss*

$$L(y, p) = -\sum_{t=1}^{N} y_t \log(p_t)$$

2.2 b) One Loss function we may use to address the unbalanced dataset of green turtles :

<u>"Focal Loss"</u>

Unbalanced datasets are common problem in classification tasks, where number of instances in one class is significantly smaller than number of intances in another class.This will lead to biased models that perform poorly on minority class.A common method for addressing class imbalance is to introduce a weighting factor $\alpha \in [0, 1]$ in a standard *Cross Entropy Loss* by assigning different weights to different classes based on their importance or frequency in the dataset to mitigate the effects of imbalance by increasing the weight of minority classes.
However, the ***Focal Loss***[3] introduced in the paper *"Focal loss for Dense object detection"* by *Tsung-Yi Lin et al. (2017)* has been

shown to be another effective loss function for addressing class imbalance in classification tasks.Widely adopted in various computer vision tasks.They proposed to add a modulating factor $(1-p_t)^\gamma$ to the *Weighted Cross Entropy Loss* with tunable focusing parameter $\gamma \geq 0$ .Let's breakdown its mathematical formulation for multi-class classification step by step:

Given

- y: the ground truth labels
- p_t: predicted probabilities from the model for each class
- γ: the focusing parameter
- α : the balancing parameter for each class
- N : the number of class.

the *Focal loss* is defined as:

$$FL(y, p_t) = -\sum_{t=1}^{N} \alpha_t (1-p_t)^\gamma \log(p_t)$$

It might be a good choice to address unbalanced dataset of the green turtle as it dynamically adjusts the contribution of each example to the overall loss based on its predicted probability and predefined parameter γ and $(\alpha_t)_t$.

References

[1] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[2] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239, 2017.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[4] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.