Deep Learning Application for Plant Classification on Unbalanced Training Set

Rafael S. Pereira[1], **Fabio Porto**[1]

[1]Laboratório Nacional de Computação Cientifica (LNCC)
CEP: 22651-075 – Petrópolis – RJ – Brazil

{rpereira,fporto}@lncc.br

Abstract. *Deep learning models expect a reasonable amount of training instances to improve prediction quality. Moreover, in classification problems, the occurrence of an unbalanced distribution may lead to a biased model. In this paper, we investigate the problem of species classification from plant images, where some species have very few image samples. We explore reduced versions of imagenet Neural Network winners architecture to filter the space of candidate matches, under a target accuracy level. We show through experimental results using real unbalanced plant image datasets that our approach can lead to classifications within the 5 best positions with high probability.*

1. Introduction

Across the world there are many herbariums which host many different species of plants. For example, we could cite the Botanic Garden of Rio de Janeiro. The JBRJ collects thousands of different images of plants. They need to classify them matching against known species characteristics and, eventually, registering new ones. Classifying samples against thousands of known species is a daunting task that may take long and be prone to errors.

Given the recent results in the field of Deep Learning for Image Classification, we propose to study some deep neural network architectures that could help to solve this problem.

In this work we were faced with some challenges given the conditions of the data we expect to process. This is mainly due to the fact that Deep Learning algorithms can be data hungry and require many examples to actually learn how to classify a given instance. Unfortunately, herbarium data may not provide us sufficient sample data for most species.

Moreover, we were faced with the challenge of a specialist versus generalist specie, since we could choose that the network must learn how to classify the plant by its leaf, or its fruit, tree trunk, flowers as well, if one chooses the specialist route then the network can yield higher accuracy at the cost of needing very specific information from the plant to function. The other challenge is the imbalance of the dataset, where the distribution of the data is far from uniform, in our case its a fast decaying curve, which can look like an exponential curve or a power law curve. This is a typical problem in Deep Learning as a statistical solution would bend towards the class with more sample instances.

Our contributions in this paper are the study of *reduced* deep neural networks and the analysis of how training can reduce the effect of the unbalanced data. The term *reduced* neural network here is used because we consider the architechure of models that have won the imagenet competition [1]. We investigate how these wining models behave when the core of the network structure is mantained but the number of network layers is reduced (i.e. less deep). We also present an approach to measure model efficiency that verifies whether it is still improving when the accuracy metric converges on the training process. Finally, we analyse the importance of pre-processing your data for maximum efficiency as well as choosing the kind of data you should work with.

The remaining of this paper is structured as follows. In section 2, we discuss previous works related to classification on data imbalance condition and approaches to plant classification. Next, in section 3, we discuss the methodology applied to solve the problem discussed in this paper. Section 5 presents the dataset used in our experiments and a discussion on the experimental results. Finally, section 6 concludes.

2. Related Work

2.1. Imbalanced Data

Much has been discussed both in the problems of dealing with unbalanced data distributions for Machine Learning and Deep Learning. [H. Han and Mao 2015, G. E. Batista and Monard 2004] use techniques based on data distribution for undersampling or oversampling. Other works develop on the algorithms themselves. For instance, [Shoujin Wang 2016] proposes loss functions that would not be minimized when predicting all as the most frequent class yielding a high accuracy. [Lin et al. 2019] formulates the classification problem as a sequential decision making process and designs a framework using reinforcement learning to deal with imbalanced data.

2.2. Plant classification

Plant image classification proves to be a non trivial task. Many features might be considered to correctly predict species since different species can look mostly the same to the untrained eye. Many different techniques were explored over the years, for example [Zalikha and Mohtar 2011] compare the effectiveness of Zernike Moment Invariant, Legendre Moment Invariant and Tchebichef Moment Invariant techniques. [Sandeep 2012] used manual features to identify Indian medicinal plants, [Jyotismita 2011] used moment invariant and centroid approaches to extract features, while [Chomtip and Chutpong 2011] developed Thai Herb Leaf Image Recognition System using a K Nearest Neighboors algorithm for classification.

3. Methodology

To explore different models for this problem, we are interested in more than just the raw accuracy of these models but we shall look at the following:

- Consider a model trained on a set of n possible classes;
- We try to predict the class of an element of the set;
- The model chooses the correct class as its first option;
- Choosing another element the model tells us there are two more probable classes than the correct one;

[1] http://www.image-net.org/challenges/LSVRC/

- Then for another element the model chooses m more probable classes than the correct one with $(m < n)$;
- We can then repeat the process for many elements and record the relative frequency of every possible value of $m : \{1 : n - 1\}$;
- The frequency distribution of suggested accuracy can be modeled by a probability density function (PDF); expect better models to be modeled by PDFs that converge to zero very fast.

By having this PDF we could then look at the Deep Learning problem not only as a full solution, but as a reducer to the classification problem by instead of the end user having to choose between eighty options, he/she has to choose from only between two for example.

To do so, consider a problem with a set of n classes.
By drawing a distribution $f(x)$ we could say the following:

$$\int_0^n f(x)dx = 1 \tag{1}$$

$$\int_0^m f(x)dx <= 1 \tag{2}$$

Considering $m > 0$.
So if we for example want a confidence of 90 percent that our solution options contain the data we could solve for p the following equation:

$$\int_0^p f(x)dx = 0.9 \tag{3}$$

Which tells us that if we return the first p options to a specialist he/she would have the right option to choose from.

By choosing models that yield PDFs for which this value of p is small, we can then highly reduce the dimension of a classification problem, to one that can be dealt by a specialist in the case we have few data, or in another words instead of the end user needing to classify between n classes using his own knowledge, he/she has to classify with only m classes to choose, where $m << n$.

4. Network Architecture

In this paper we explored how reduced forms of the imagenet architectures behave when trained with datasets that have too few data for a deep learning problem. We shall discuss specially about two reduced architectures: the Reduced VGG and a reduced version of the modified GoogleNet/Inception architecture proposed by [Jose Carranza-Rojas 2017].

The implementation of this study used the one available in keras [Chollet et al. 2015] using the backend Tensorflow[Abadi et al. 2015] and the python language.

4.1. VGG

The VGG architecture was proposed on the imagenet competition by [Zisserman" 2015], In this paper we shall use a reduced version that work as follows:

- Input Layer
- Convnet 32 filters relu activation;
- Batch Normalization;
- Convnet 64 filters relu activation;
- Batch Normalization;
- Convnet 64 filters relu activation;
- Batch Normalization;
- Convnet 128 filters relu activation;
- Batch Normalization;
- Convnet 128 filters relu activation;
- Batch Normalization;
- Dense layer.

The activation function of the dense layer takes into account the compilation method. We can test a binary compilation for multi label classification using a sigmoid activation, or a categorical compilation for a single label across many possible labels using a softmax activation. In this paper we shall discuss the categorical method.

4.2. GoogleNet

The Original GoogleNet was equally present on a later imagenet competition. The one we used in this paper is a modification on the inception module of the original network. For those that are not familiar, the inception module is a set of four sequential convolutional networks that are built parallel to one another with variations on the kernel. The new inception module is built as follows:

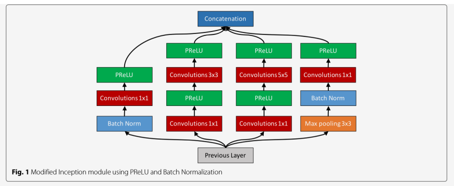

Figure 1. Inception Network [Jose Carranza-Rojas 2017]

Then the reduced version is composed of the following layers:

- Input Layer;
- Convolution with 64 filters kernel 7x7;
- Prelu activation;
- Convolution with 64 filters kernel 3x3;
- Prelu activation;
- Inception 256 filters;
- Prelu activation;
- Inception 256 filters;
- Prelu activation;
- Dense layer linear activation;
- Dense layer softmax activation.

5. Experiments

5.1. Dataset

In the present work we gathered a dataset of images from the LifeCLEF 2016 plant classification challenge. This challenge contains 93760 images distributed across a thousand species.

This dataset presents the following challenges for the prediction:

- Imbalanced dataset. The distribution of number of species per number of examples follows a fast decaying curve;
- Specialized examples vs Generalist examples;
- Bad quality of images in some species introducing noise in the learning process.

Examples of the dataset are as follows:

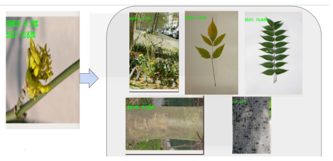

Figure 2. Examples of different species analyzed in this dataset, we can see how the classification process is done in this image as well.

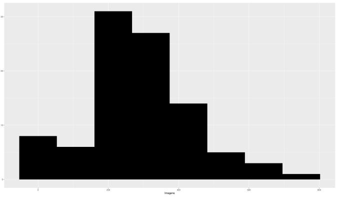

Figure 3. Dataset Distribution, x axis represents the number of images in a given class, and y axis represents how frequent classes with these number of examples are.

Since the dataset has too few data the tests were done on a smaller problem, we choose all species that had at least 200 examples and sampled some species with less examples, one with even 14 examples only, the distribution still is highly unbalanced.

For the train test process we used a 80/20 split meaning that the test size contains 20% of the dataset size.

As we can see in figure 2 this dataset contains many differents views of a plant, for example images of leaves, fruits and even the tree trunk, figure 3 also shows us how this dataset is unbalanced.

5.2. Experiment

For the experiment we used the following technologies:

- Python 3.4.9
- Keras 2.0.5
- Tensorflow 1.2.0-rc2

The computation environment where the code was run is the following:
A Dell PowerEdge R730 server, Processor: 2x Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz, Memory: 768 GB, Storage: 9.6 Terabytes
This is called the fatnode which resides inside the Petrus cluster, located in LNCC.

The Petrus cluster is a cluster designed for the challenges of Big Data and so its hardware was built to deal with the problems of Data Analysis for large volumes of data in a efficient manner, it is used by the Data Extreme Lab (DEXL), the group dedicated to data science at LNCC. The purpose of the following experiments was to measure the following topics:

- Effect of training on distribution;
- Effect of Imbalance as training progresses;
- Comparison between different models.

5.3. Experimental Results

In this section, we shall discuss the experimental results obtained in this work.

5.3.1. Different distributions across epochs

We can analyze how different species behave when a model is trained on more epochs, the purpose is to identify how the training process changes the following graph distribution as well as identifying species that are easier to train, some results as follows:

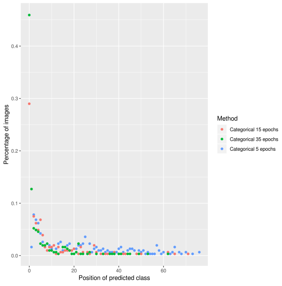

Figure 4. Syringa vulgaris specie

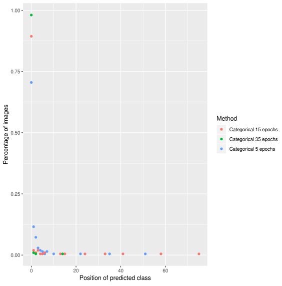

Figure 5. Daphne cneorum specie

We can see from figures 4 and 5 that species whose model yields higher accuracy have a distribution that decays fast behaving like the exponential or power law family of curves. One may also observe that the species in Figure 4 is harder to learn than that in Figure 5, since its curve does not start decaying as fast.

Considering the above curves, one may normalize them such that they may be described by a probabilistic density function (PDF). Then, we can calculate its expected value $\bar{x}$ given by the formula:

$$\bar{x} = \int_{-\infty}^{\infty} x f(x) dx \tag{4}$$

Then calculating this for all species we are interested in, we can draw the figure 6, which is the expected value of the predicted position per number of examples of the species.
The purpose os this experiment is that we can measure how much can we reduce the original problem, so instead of having to choose between n species, the end user chooses between m, $m << n$

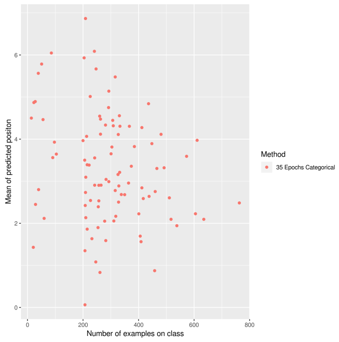

Figure 6. Categorical Method Solution

Another way to look at it is to consider *what-if* we would not only deliver to the end user our network first choice when predicting, but instead the first k best ranked classes. What would it be the probability of the correct choice not falling among the ones delivered?

To answer this we compare both the VGG architecture and the inception in the figure 7:

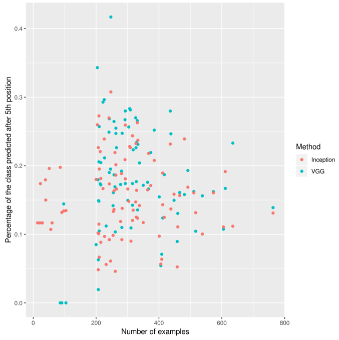

Figure 7. Comparison of Modified googlenet performance vs VGG

In figure 7 we can see how the VGG architecture presents higher accuracy for some of the species with few data, but as the amount of examples increases, the opposite happens. We understand that further experiments are encouraged to better differentiate both approaches.

6. Conclusion

In this paper we analyzed how good models must have a classification distribution that decays fast like the exponential or power law curve, as well as the training process can make the classification process invariant to the data distribution. We also showed that deep learning does not need to have lot of data to actually be useful but that we can look at the technique not only as something to get the actual solution, but as a problem reducer, where instead of hundreds of possible options a specialist must choose from, you can deliver only a very small number of options and be sure the correct option is there for the specialist to choose.

7. Acknowledgements

The authors would like to thank the support of Conselho Nacional de Desenvolvimento Científico e Tecnológico(CNPQ),This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico(CAPES) -Finance Code 001.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Chomtip, P., S. R. P. T. and Chutpong, C. (2011). Thai herbl leaf image recognition system (thlirs). *Kasetsart J.(Nat.Sci.)*, pages 551 –562.

G. E. Batista, R. C. P. and Monard, M. C. (2004.). A study of the behavior of several methods for balancing machine learning training data,. *ACM SIGKDD explorations newsletter,*, 6(1):20–29,.

H. Han, W.-Y. W. and Mao, B.-H. (2015). "borderline-smote: a new over- sampling method in imbalanced data sets learning,". *International Conference on Intelligent Computing Springer*, page 878–887.

Jose Carranza-Rojas, Herve Goeau, P. B. E. M.-M. A. J. (2017). Going deeper in the automated identification of herbarium specimens. *BMC Evolutionary Biology*.

Jyotismita, C., a. R. P. (2011.). Plant leaf recognition using shape-based features and neural network classifiers. *International Journal of Advanced Computer Science and Applications (IJACSA)*, pages 41–47.

Lin, E., Chen, Q., and Qi, X. (2019). Deep reinforcement learning for imbalanced classification.

Sandeep, A., a. P. (2012). Development of a seed analyzer using the techniques of computer vision. *International Journal of Distributed and Parallel Systems (IJDPS)*, pages 149–155.

Shoujin Wang, Wei Liu, J. W. L. C. Q. M. P. J. K. (2016). Training deep neural networks on imbalanced data sets. *International Joint Conference on Neural Networks (IJCNN)*.

Zalikha, Z., P. S. I. S. and Mohtar (2011). Plant identification using moment invariants and general regression neural network. *11th International Conference on Hybrid Intelligent Systems (HIS)*, pages 430–435.

Zisserman", K. S. A. (2015). Very deep convolutional networks for large -scale image recognition.