MACHINE LEARNING

Neuronal Network: Weather Recognition

Guillermo Marion López y Louka Vanhoucke

Introduction

In this project, we will explore the ability of a neural network to recognize different types of weather conditions such as dew, fog, smog, and rain. We chose this dataset because many devices and vehicles can adapt their behavior based on their environment, so being able to accurately identify the weather can be useful in certain situations.

We have selected 7 different classes, each class has more than 120 pictures :

Dew	
Fog Smoke	

Lightning	
Rain	
Rainbow	
Sandstorm	
Snow	

All images used for the training set and for the validation set, have been extracted from this dataset:
https://www.kaggle.com/datasets/jehanbhathena/weather-dataset

Hyperparameters Configuration

We have conducted experiments with different hyperparameters and have created an efficiency table to compare the results. We have also implemented a function that generates rotated, zoomed, and flipped versions of our previously trained images from the dataset by using ImageDataGenerator, img_to_array, and load_img from the tensorflow.keras.preprocessing.image module. In addition, we have tried various optimizers, including SGD, Momentum, NAG, and Adadelta, but found that Adam provided the best results for our dataset and task

Without Data Augmentation

Versión	Layers	Batch Size	Activation Function	DropOut	Validation Accuracy
1	5	16, 32, 64, 128, 7	relu, softmax	0.25, 0.25, 0.5	0.5143
2	7	32, 64, 128, 256, 20, 256, 7	relu, softmax	0.20, 0.25, 0.30, 0.35, 0.5	0.5000
3	7	32, 64, 128, 256, 20, 256, 7	relu, softmax	0.20, 0.25, 0.30, 0.35, 0.5	0.5571

With Data Augmentation

Versión	Layers	Batch Size	Activation Function	DropOut	Validation Accuracy
4	6	64, 64, 64, 64, 256, 7	relu, softmax	0.10, 0.10, 0.10, 0.10	0.7861
5	6	64, 128, 128, 128, 256, 7	relu, softmax	0.25, 0.25, 0.25, 0.30	0.8208
6	6	64, 64, 64, 64, 256, 7	relu, softmax	0.25, 0.25, 0.25, 0.30	0.8266
7	6	128, 64, 128, 64, 256, 7	relu, softmax	0.25, 0.25, 0.50, 0.50	0.8862
8	5	128, 128, 128, 256, 7	relu, softmax	0.50, 0.50, 0.50, 0.50	0.9712

Our best value is the last version **8,** where we used different activation functions and fewer layers than other versions , and thanks to the data augmentation we have increased our result significantly.

Conclusion

The use of data augmentation significantly improved the accuracy of our model. Without data augmentation, our model achieved an accuracy of around **55%-60%**, while using data augmentation, the accuracy increased to **98%-99%** after multiple tests. The loss value also decreased significantly and the training and validation scores reached a high and stable level when data augmentation was used, as shown in the graph.