**NORBERT’S REPORT**

# Initial situation

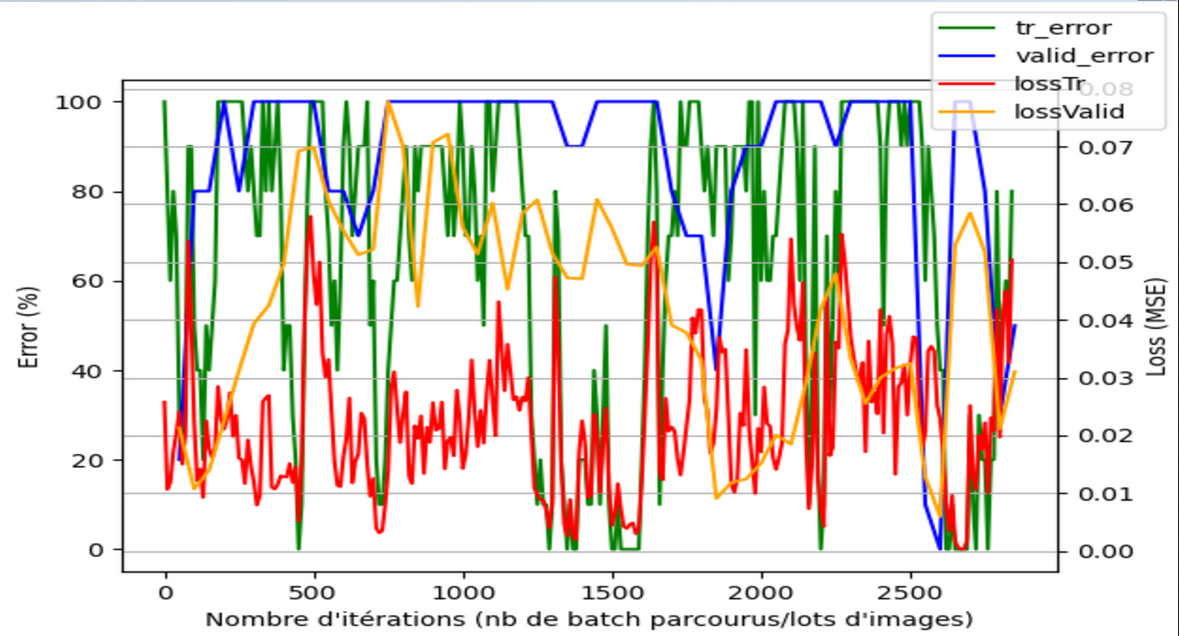
First, we decided to realise a convolutional model based on traditionnal Convolutional Network architecture. We used the article *Autonomous Driving System based on Deep Q Learning* andthe resources of the article *A method to recognize the moving human activity about posture and velocity* ***(c’était pr quoi déjà ?)****.* ***(si c’est ce qui suit, ce n’est pas la seule piste qui a menée à keras, je ne suis pas vraiment sûr que ce soit le moment d’indiquer cette source.)***Indeed, **(Du coup j’aurais coupé le indeed et soit mentionné en intro du projet qu’on a choisit d’utiliser / centraliser tout avec keras soit dit directement «We used ....)** we used a deep learning API written in Python, running on top of the machine learning platform [TensorFlow](https://github.com/tensorflow/tensorflow) : Keras.

Xception based network is an image classification model. We trained it from scratch on the Nuscene dataset. **Par ex. Peut-être ajouter une nouvelle phrase pour expliquer pourquoi : redémarrer depuis 0 peut permettre de mieux comprendre ce qui a le plus d’influence sur les performances du modèle)**. It starts from JPEG image files on disk, without leveraging pre-trained weights or a pre-made Keras Application model.Its goal is to classify images among 23 available classes such as trucks or pedestrians**.**

**Initial model:** Every five passages of the train mode, we do the valid mode.

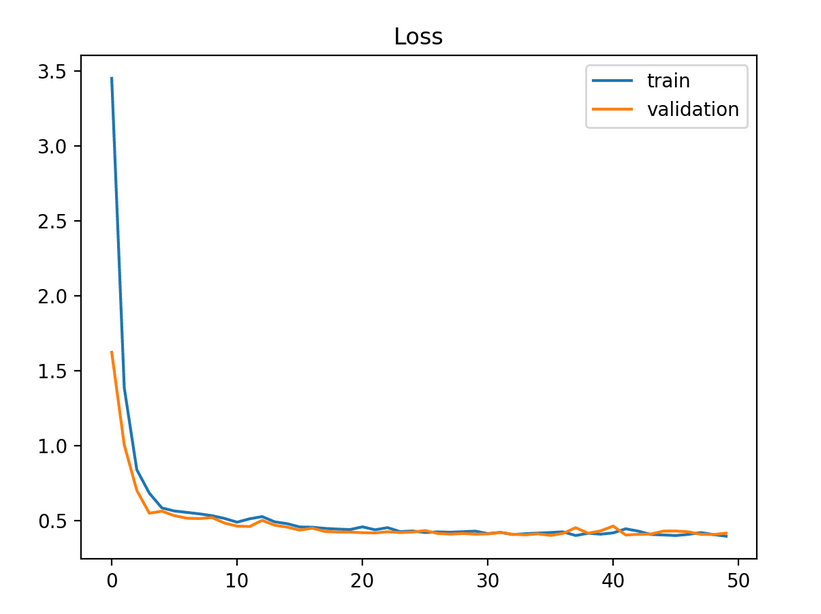
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Image size** | **Batch size** | **Adam Optimizer** | | | **Loss** | **Activation function of the last layer** | **Number of modules** |
| **Learning rate** | | **Adam epsilon** |
| 1600x900 | 10 | 10^-3 | 10^-7 | | Mean squared error | softmax |  |

**(Tu entendais quoi par nb de filtres ? Au final justement je ne l’ai pas exploré, j’ai trouvé plus pertinent de modifier le nombre de modules, pattern répété dans le réseau pour théoriquement augmenter ou réduire ses capacités d’apprentissage )**



Number of iterations (number of crossed batch)

This graph shows the error rate is totally uncertain unstable and the loss curve is not monotonic: there is no training. Indeed the perfect curve would look like figure 2.



# Tests to improve the training

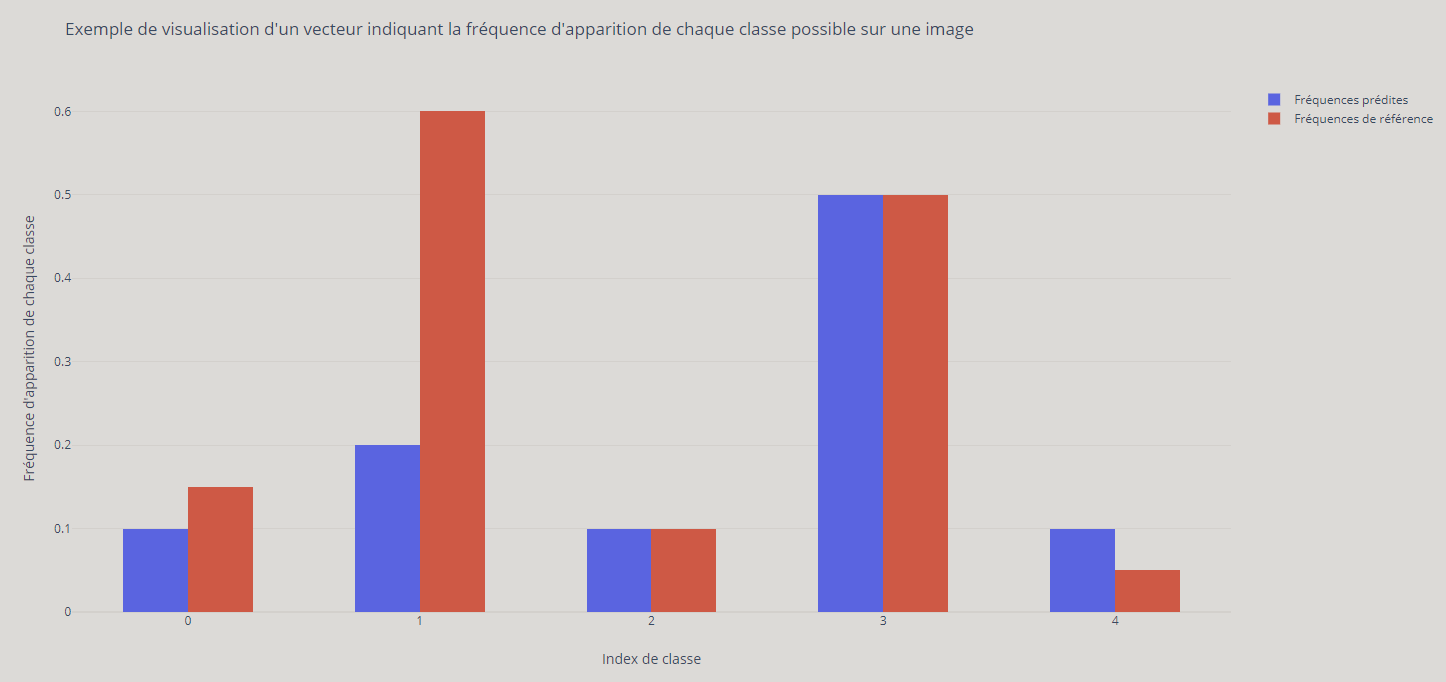
1. **Hypothesis:** our model is correct. We adjust the training parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tests** | **Goal** | **Oberved impacts** | **Result** | | |
| Increase the Adam epsilon and or decrease the learning rate value | Slow the learning process to improve the generalization | No great improvements. All the graphs are like the first one shown to the right | Learning rate  1e-3 | Adam epsilon  1e-7 |  |
|  | | |
| **Regarding the result we decided to choose a learning rate of 1e-3 and an Adam epsilon of 1e-7** | | | | | |
| Changing the batch size | Give bigger samples in order they represent better the global dataset | No improvements whatever the batch size | Souci de memoire car trop de place |  |  |
| **Regarding the result we decided to keep a batch size of 10 and an image size of 400\*225** | | | | | |
| Changing the loss: we use the categorical\_cross\_entropy[[1]](#footnote-1) | More efficient to the learning process to guide the model | Regression task, not prediction |  |  |  |
| Modify the architecture |  |  |  |  | ANNEXE |

1. **Hypothesis**: Actually, our reasoning was not correct. Indeed, we tried to predict how many times is an object present in the image than another object. We tried to obtain a result expressed in percentage. Indeed, the predicted values are added up to one. It returns a probabilities vector. We replace the softmax activation function of the last layer by the sigmoid function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tests** | **Goal** | **Oberved impacts** | **Result** | | |
| Change the softmax activation function by the sigmoid function | The sigmoid function returns a value between 0 and 1 | The network is less imprecise. | Une image contenant texte, instrument d’écriture, stationnaire  Description générée automatiquement | | |
| Adjust the values of the learning rate and the Adam epsilon | Reduce the number of parameters and have a less deep network. Indeed, deeper is a network, more difficult is it o train. | Convergence of the loss but not of the error. The model is not steady. | Learning rate  1e-3 | Adam epsilon  1e-7 | Une image contenant texte, instrument d’écriture, stationnaire, crayon  Description générée automatiquement |
|  | | |

1. **Hypothesis**: Now, let us look at the error metric. The error metric is used to calculate if two objects are similar or mot. It is calculated from the accuracy: error\_metric = 100 – accuracy. However, the accuracy counts the percent of exactly predicted values. So, the error metric does not show how much is a model accurate. We do not do the difference between a big error and a small error. Indeed, we predict normalized frequency of each class (values between 0 and 1). For example, we have an error of 20%. If there are 200 objects in this image, the error will be more important if there would be 20 objects. So, we must predict actual values.



In this graph, the accuracy is 2/5\*100 = 40% so 60% of the error.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ideas** | **Advantages** | **Drawbacks** | **Conclusion** |
| Original Idea | The model shows how much a class is present than another class | -Impossible to know how many objects are detected is an image  -Value useless if objects are well detected in the image or not | Idea useless |
| First Idea | Predicted two values: 1 and 0 | 1 present object -> P(object)=1 ; 1 present object twice -> P(object)=1 | Idea useless because we can not know how many an object present is. |
| Second Idea | -Predicted integer  -Value more useful to understand how many classes and how many objects are in an image |  | Idea more relevant |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tests** | **Goal** | **Oberved impacts** | **Result** |
| Change the activation function of the last layer. | We want to predict positive integers. However, an artificial intelligence works better on floats. We search an activation function f as f : R -> R+ | -No continuous decrease in loss  -Some very important variations of the loss | Une image contenant texte, instrument d’écriture, stationnaire, capture d’écran  Description générée automatiquement  Figure 1: Graph of the linear activation function |
|  |  |  |  |

1. [Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.](https://machinelearningmastery.com/cross-entropy-for-machine-learning/) [↑](#footnote-ref-1)