EAI6010 - Module 2

Customize a Pre-trained Model for CV Classification

Submitted to:
Prof. Abhijit Sanyal

Submitted By:
Abhigna Ramamurthy
002982276

Date : 7th June 2022.

Introduction

Computer vision is a branch of artificial intelligence (AI) that allows computers and systems to extract meaningful information from digital photos, videos, and other visual inputs — and then act or make suggestions based on that knowledge. Starting with pre-trained weights, rather than random weights, reduces training time and so is more efficient. Sampling can also increase data usability and save cycle time. Another option is feature engineering, which use machine learning techniques to turn existing data, domain knowledge, and intuition into superior, analytics-ready information. We are using a pre-trained model because, in general, the extracted characteristics for the classification job will be comparable. In the project I used the Oxford-IIIT Pet Dataset by O. M. Parkhi et al., 2012 which features 12 cat breeds and 25 dogs breeds. Our model will need to learn to differentiate between these 37 distinct categories. There are 200 images for each class. The images feature a wide range of scale, pose, and illumination. Breed, head ROI, and pixel level trimap segmentation are all related ground truth annotations on all images (Parkhi et al., 2012).

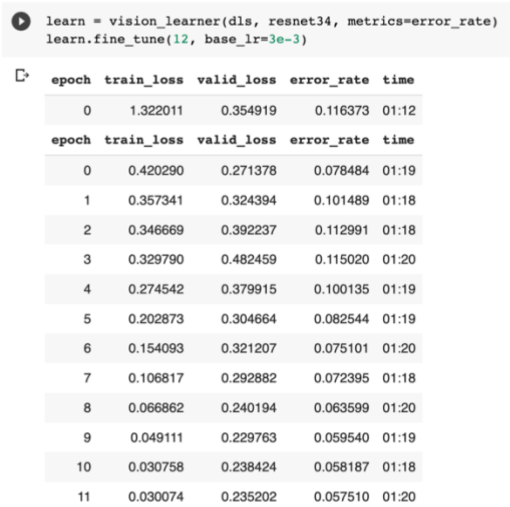
Project Overview

The data of pet images with pet breeds as file names was considered for the analysis. Getting the images ready using data blocks, presizing (helps in getting all the images from the dataset to the same dimensions, which can be further fed into the GPU for analysis), fitting the model, softmax probabilities that help determine the exact class of the pet breed, model interpretation using confusion matrix, finding optimal learning rate using learning rate finder, unfreezing layers and training the data again, are some of the steps taken to achieve pet breed classification model.

Questions

1. Can you create and document a scenario where over training occurred?

If the training takes too many epochs with only last layer of the neural network being trained, then we can see a highly overfit model. We can see that in the below example as the train_loss goes down and valid_loss goes up.



```
learn = vision_learner(dls, resnet34, metrics=error_rate)
learn.fine_tune(12, base_lr=3e-3)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.322011	0.354919	0.116373	01:12

epoch	train_loss	valid_loss	error_rate	time
0	0.420290	0.271378	0.078484	01:19
1	0.357341	0.324394	0.101489	01:18
2	0.346669	0.392237	0.112991	01:18
3	0.329790	0.482459	0.115020	01:20
4	0.274542	0.379915	0.100135	01:19
5	0.202873	0.304664	0.082544	01:19
6	0.154093	0.321207	0.075101	01:20
7	0.106817	0.292882	0.072395	01:18
8	0.066862	0.240194	0.063599	01:20
9	0.049111	0.229763	0.059540	01:19
10	0.030758	0.238424	0.058187	01:18
11	0.030074	0.235202	0.057510	01:20

Figure 1: Over training model example

2. What training methods did you find helpful-useful to prevent overtraining and why?

By monitoring model performance continuously, we can trigger the end point of training to make sure we do not overtrain the model. This point is usually determined based on the data points available. For a small dataset (e.g., 200 records), checking the train_loss and valid_loss along with model performance every 10 epochs would be good, but for a large dataset (e.g., more than 20,000 records), it would be good to check every epoch. The trigger to stop training would generally be when validation loss plateaus, or no change in performance metrics, or absolute change in a performance metric, or decrease in performance. Plotting the learning curves will also help to stop learning at the right time (Brownlee, 2019). Using methods like dropout can serve as cost effective way to regularize the model and avoid overfitting.

3. Did you reach a point where it was clear that you should stop training?

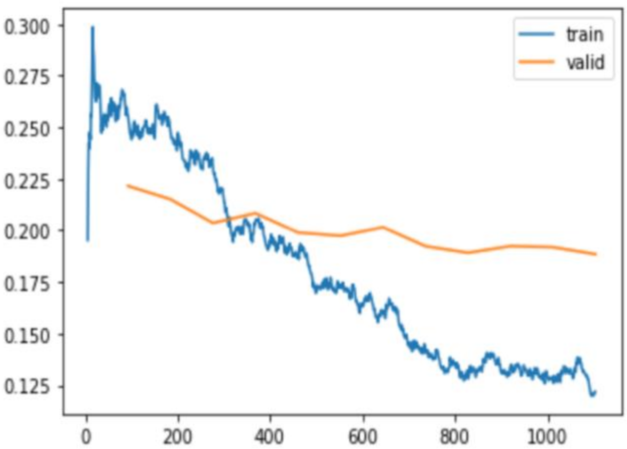


Figure 2: Loss curve for the model

A neural network is stopped training when the error, i.e., the difference between the desired output and the expected output is below some threshold value or the number of iterations or epochs is above some threshold value. When the validation metrics can no long improve, it is a good stopping point. In the case of the per classifier model, we can see that the validation plateaus after a point, which is where it is good to stop training the model or else overfit of model can surface. Accuracy of the model can also showcase quality metric to decide a stop (Igareta, 2021).

4. What image classes had the best-worst performance and for the worst performing classes, what is your recommended path for improvement?

The best performing top five categories are: Havanese, newfoundland, pug, beagle, and English cooker spainiel. The worst performing top six categories are: boxer, American bulldog, American pit bull terrier, Staffordshire bull terrier, Ragdoll, and Birman. Furthermore, we can observe that the distribution is extremely skewed when we display the confusion matrix: the model repeats the same mistakes over and over, but it seldom confuses other categories. This indicates that it is having difficulty distinguishing certain categories from one another; this is normal behaviour. Getting more data for such categories can also help avoid the issue. Employing dropout methodology can help prevent overfitting.

References

Brownlee, J. (2019). A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks. Machine Learning Mastery. https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Igareta, A. (2021). The Million-Dollar Question: When to Stop Training your Deep Learning Model. Towards Data Science. https://towardsdatascience.com/the-million-dollar-question-when-to-stop-training-deep-learning-models-fa9b488ac04d

Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Visual Geometry Group - University of Oxford. Information Engineering Main/Home Page. https://www.robots.ox.ac.uk/~vgg/data/pets/