Introduction to Machine Learning Project Report

Marco Berardelli
University of Trento
marco.berardelli@studenti.unitn.it

1. Introduction

This assignment aim to automatically recognize different type of galaxies. Given an arbitrary image of a galaxy, the algorithm should predict the correct galaxy class. There are in total ten different classes, some of them have similar features: *barrel spiral, cigar shaped smooth, disturbed, edge-on with bulge, edge-on without bulge, in-between round smooth, merging, round smooth, unbarred loose spiral, unbarred tight spiral*.

The dataset has a total of 17736 images and it is splitted in two smaller datasets, train and test. The first set has 12415 images and the second only 5321. Since the validation dataset is not present, 20% of the train images are extracted to create one.

2. Proposed Method

Since this is an image classification problem, I considered using a neural network having convolutional layers. At first, the main idea was to implement my own neural net but, since there are already differents implementations of neural netork, I opted to use an existing one.

The one that I have choose is a residual neural netork, ResNet[1]. The main characteristic of this model is the presence of *"identity shortcut connections"*(1), that skip one or more layer. This improves performance on deeper models and also helps against the vanishing gradient problem, that happens when the gradient is back propagated to the initial layers, and the varius moltipiclations can make the gradient very small, preventing the update of the weights.

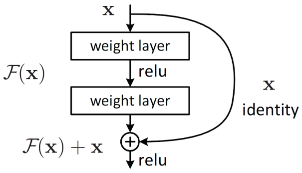

Figure 1. Rapresentation of the identity shortcut connection

I tried three approach, all of them use the ResNet model. The first uses a model not pretrained and trains it from scratch, the last two uses a technique called tranfer learning. With transfer learning is possible to use an already trained model to train it again using a new dataset. There are two ways of implementing the transfer learning. The first approach is called *fine tuning model*, that uses a pretrained model and trains it with a new dataset. The second one is called *fixed features extractor*, like the fine tuning, uses an already trained model but it freezes all the hidden layers parameters, updating just the weights of the output layer.

To calculate the loss function, all models use the Cross Entropy function. This function measures how much the real label is close to the predicted one. The math formula for Cross Entropy is

$$-\sum_{m=1}^{M} y \log(p) \tag{1}$$

Where *M* is the number of classes, *y* the correct label regarding *m* (0 or 1) and *p* is the predicted probability for class *m*.

To optimize the models, I'm using Stochastic Gradient Descend (SGD). This algorithm tries to find the minimun of a function (in this case the loss function): we choose a starting point and then the algorithm, through derivatives of weights and bias, tells the model how to change its parameters.

3. Results

The metrics used to evaluate the model are global accuracy A_{Glob} and class accuracy A_{Tot}. Also, to have a deeper analisys, the Intersection Over Union (IoU) was measured.

3.1. Not Pretrained

This is the first approach used. Having a model that has not been trained before, the accuracies started very low (2), and then increased over 0.60, exception for the intersection over union, that stopped at 0.4785. The low IoU shows us that this model has difficulties recognizing the right area to predict.

Best performances have reached an A_{Glob} of 0.6692 and an A_{Tot} of 0.6126.

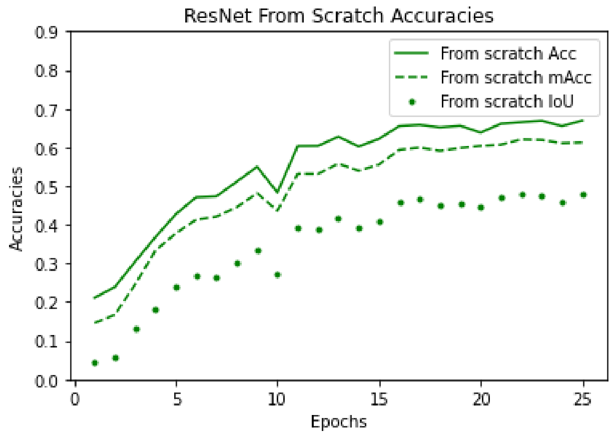

Figure 2. ResNet not pretrained accuracies

3.2. Fine Tuning

Fine tuning approach reached the best accuracy of all three, A_{Glob} 0.8259 and A_{Tot} 0.8052.

Using a pretrained model and then update all weights based on the new dataset helped reaching better accuracies (3), and also improving the correct area to recognize, achieving the IoU of 0.6926

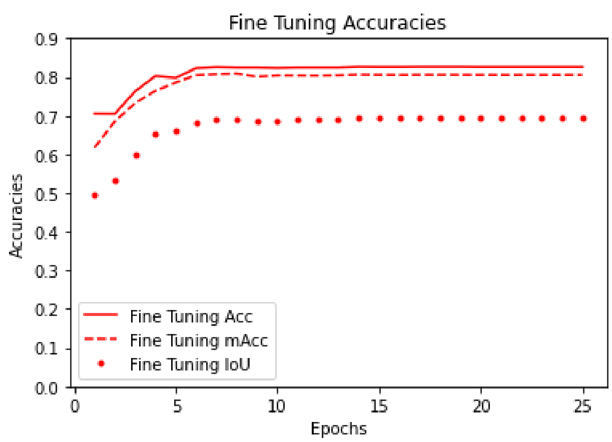

Figure 3. Fine tuning accuracies

3.3. Fixed Feature Extractor

With fixed feature extraction (4), the weights of the pretrained model are froozen, with the exception of the last layer, which in this case has ten neurons for the ten galaxy classes. The original resnet was trained on the ImageNet dataset, which contains 1000 different classes, none of them about galaxies.

Only one layers to train and the lack of useful classes in the original dataset stagnated the metrics very fast, reaching A_{Glob} to 0.5218 and A_{Tot} to 0.4648.

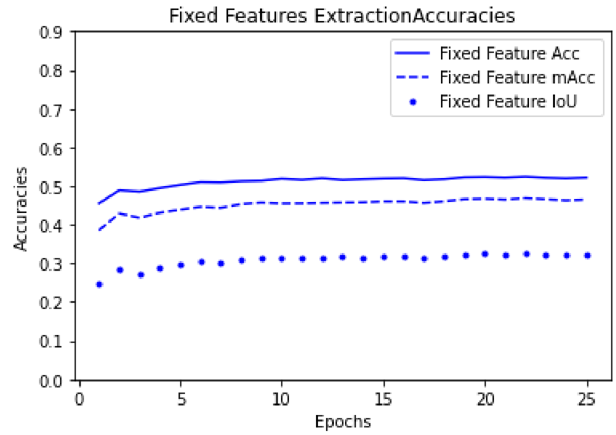

Figure 4. Fixed features accuracies

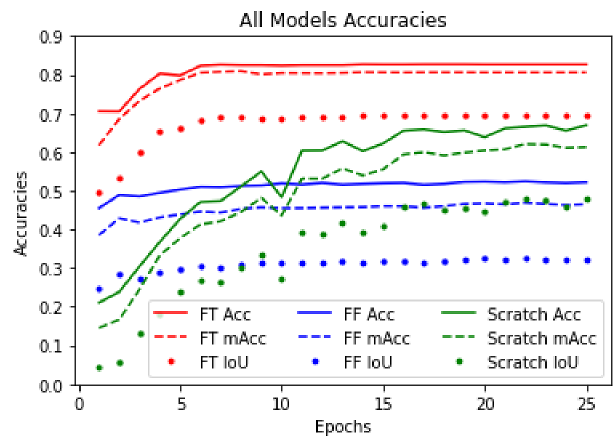

Figure 5. Accuracies of all models

References

[1] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: `1512.03385 [cs.CV]`.