Assignment 3: Image classification

Leo Alberge
ENS Paris-Saclay
leo.alberge@ens-paris-saclay.fr

Abstract

For this assignment, we use the well-known fine-tuning method for an image classification problem.

1. Introduction

The goal of this project is to classify birds from the Caltech-UCSD Birds-200-2011 bird dataset that contains approximately 1000 train images, 100 validation images and 500 test images. Given the small size of the dataset, it seems not possible to train a deep CNN from scratch. Instead [3] and [1] have shown that is was possible to use the output of a layer of a pre-trained CNN as universal representations of the image. It's therefore possible to transfer the learning of this CNN to perform another computer vision task. For this assignment, our idea is to use a pre-trained model for a classification task(ResNet101)[2] and retrain some of the layers for the specific task of bird classification. We will study the influence of the layers we retrain and the hyper-parameters of the optimizer.

1.1. Dataset

We first explored the dataset, the training set contains 1082 while the validation set contains 103 images. The training set seems to be well balanced whereas the validation set seems to be a little small and results on this data could have high variance.

1.2. Data Augmentation

Given the small size of the dataset, we use some tricks to perform data augmentation. The idea is to increase the size of the data by generating new examples from the dataset. It will be done "on the fly" by apply random transformations to the training dataset: random re-size and crop, affine transformations(rotations(relevant since birds are not always in the same posture) and scaling, random illumination modifications and normalization.

Figure 1. Example of bird images

Figure 2. Example of birds after data augmentation

1.3. First : Chopping only the last linear layer

The first idea was to take ResNet101 architecture and chop only the last linear layer and replace it with an new linear layer with the appropriate number of classes and retrain only this linear layer.

1.3.1 Training parameters

The loss used was the cross-entropy. We have used a Stochastic Gradient Descent(SDG) with fixed learning rate and momentum. We a did a grid search to tune the parameters. We plot some of the learning curves for different parameters(See figures 3, 4, 5, 6).

Figure 3. Linear layer, Learning curve(accuracy) for $lr = 10^{-2}$, $moment = 0.9$, batch of 16 images.

1.3.2 Best model and results

We used batch of 16 images, $lr = 10^{-3}$, $moment = 0.9$ for the training.
We achieved on the validation set: Average loss: 0.1267, Accuracy: 77/103 (74%). We observe that we perform badly for black bids, specifically the Fish crow and the American crow(see Figure 7)

1.4. Second idea: Relaxing some convolutional layers

The second idea was to relax some of the convolutional layers to augment the expressivity of our model: one way to do it is to retrain some pf the convolutional layers. We only tested retraining the fourth layer of the ResNet 101 architecture and the last linear layer together.

1.4.1 Training parameters

The loss used was the cross-entropy. We have used a Stochatisc Gradient Descent(SDG) with fixed learning rate

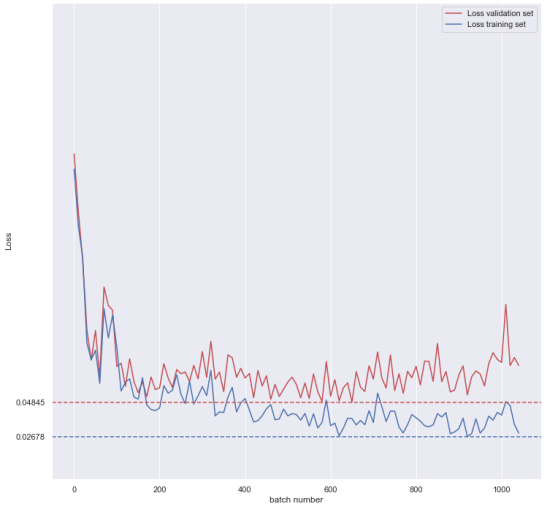

Figure 4. Linear layer, Learning curve(loss) for $lr = 10^{-2}$, $moment = 0.9$, batch of 16 images.

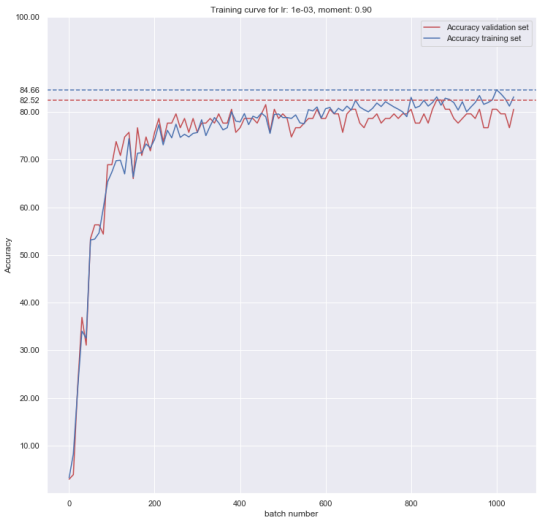

Figure 5. Linear layer, Learning curve(accuracy) for $lr = 10^{-3}$, $moment = 0.9$, batch of 16 images.

and momentum. We a did grid search to tune the parameters. We plot some of the learning curves for different parameters(See figures 8, 9, 10, 11). We found that $lr = 10^{-3}$ and $moment = 0.9$ seem to be fair choices for the training parameters, the convergence was fast and stable.

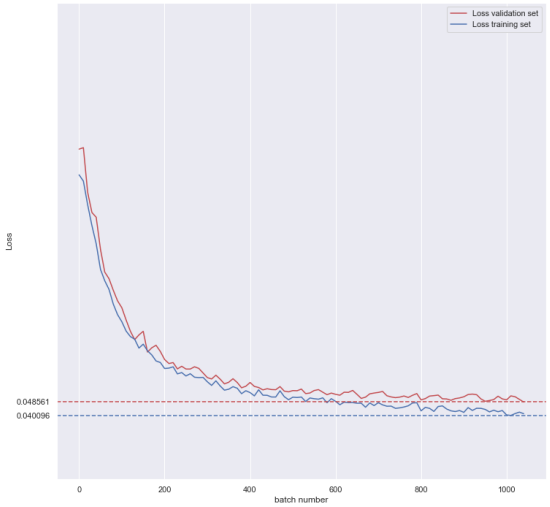

Figure 6. Linear layer, Learning curve(accuracy) for $lr = 10^{-3}$, $moment = 0.9$, batch of 16 images.

Figure 8. Learning curve(accuracy) for $lr = 10^{-2}$, $moment = 0.5$, batch of 16 images.

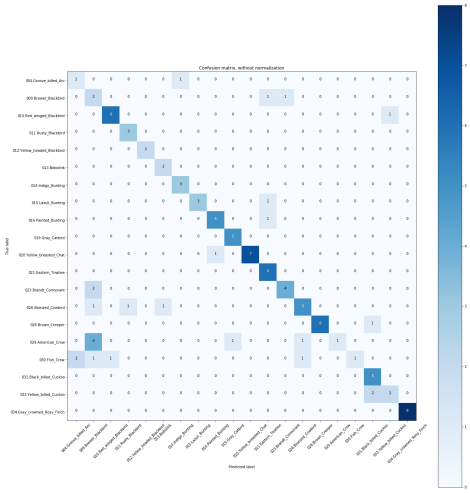

Figure 7. Confusion matrix for the best model(Linear layer retrained).

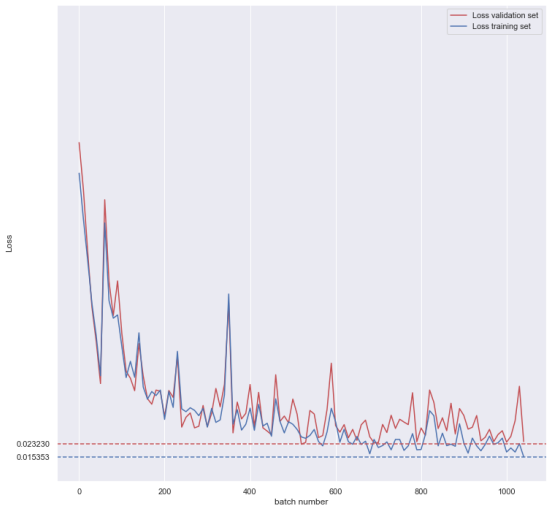

Figure 9. Learning curve(loss) for $lr = 10^{-2}$, $moment = 0.5$, batch of 16 images.

1.4.2 Best model and results

We used batch of 16 images, $lr = 10^{-3}$, $moment = 0.9$ for the training.
We achieved on the validation set: Average loss: 0.0619, Accuracy: $88/103$ (85%) (See confusion matrix Figure 12). It outperforms(10% better) the model with only the linear layer retrained. Yet this model has a high variance and should be regularized.

2. Conclusion

In this project, we tried to use transfer learning for an object recognition task. We have seen that if we retrain some of the convolution layers of the network we could achieve better results than with only the linear layer retrained. How-

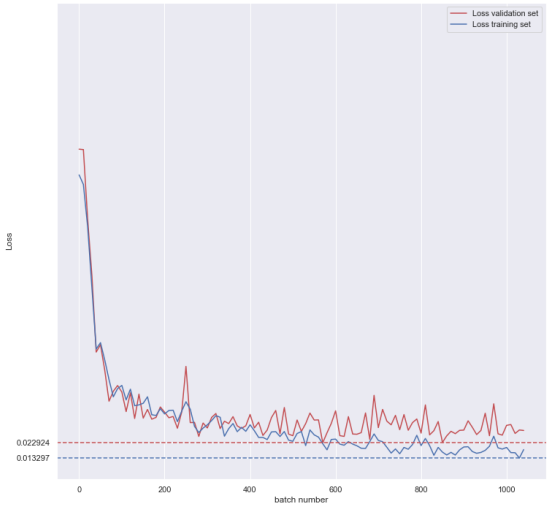

Figure 10. Learning curve(accuracy) for $lr = 10^{-3}$, $moment = 0.9$, batch of 16 images.

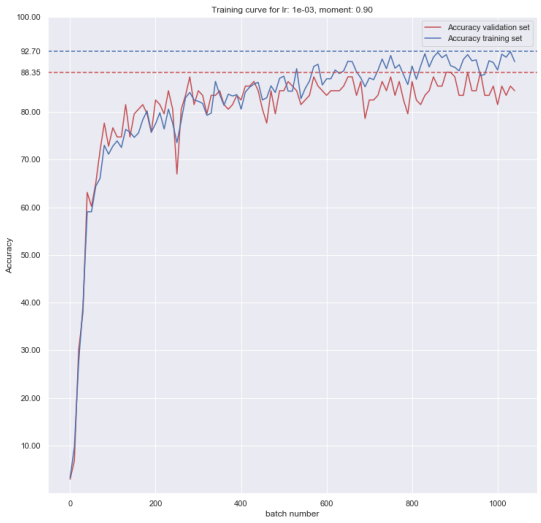

Figure 11. Learning curve(accuracy) for $lr = 10^{-3}$, $moment = 0.9$, batch of 16 images.

ever the number of parameters to learn increases a lot which increase the variance of our predictor and the dataset is small, so the next step would be to study the use of regularization: for CNNs there exists several ways regularize: dropout, weight decay, early stopping ... The influence of

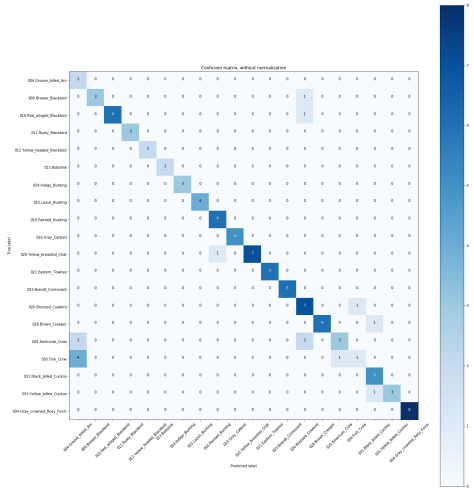

Figure 12. Confusion matrix for the best model(Linear and convolution layer retrained).

the batch size was not studied as well. We also observed that the misclassified birds in the validation set were from the same classes: Fish crow, American crow, one idea could be to penalize more these classes. Another way to improve our results would be to use a R-CNN for region proposal, indeed the way we crop the images is not perfect and could be improved by such network that will first find where the bird in the image is.

References

[1] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.