BM20A6100 Advanced Data Analysis and Machine Learning

Project name: « Classification of wood species based on images»

Autors
Student: Dmitrii Shumilin
Student number: 0589870
E-mail: ShumilinDmAl@gmail.com
Dmitrii.Shumilin@student.lut.fi

13.12.2020

1. Dataset and preprocessign

The selected dataset is a macroscopic image of a Brazilian wood cut [1]. There are 2942 images in total, the name of which consists of four digits - the first two are a class label, and the second two are a unique identifier. Since the images have a huge resolution, the process of opening and loading images on the CPU on the fly will dramatically increase the network training time. There are two ways to overcome this problem:

- Make random crops of a fixed size
- Resize image to a fixed size

In the first case, the problem arises that the future model will also have to be tested on random crops from the test sample, which may not be acceptable in terms of the end use of the model. Taking into account the above, we will simply resize the image to size 224x224 - the standard image size from Imagenet. Since it is planned to use pre-trained networks on Imagenet this seems like a natural solution, as well as future image standardization using means and variances obtained on the imagenet.

Figure 1 shows an image of two objects of the first class. By the structure, as can be seen that there is no specific direction of wood grain in the images, and the brightness of the images may vary. Then, as augmentations for the training dataset, we can additionally apply image reflections about vertical and horizontal axes, rotation by a random angle, as well as augmentation to change the contrast, brightness and saturation of the image. To increase the number of “different” images, all these augmentations are applied randomly with different probabilities, so the batch may contain both the original image, the rotated image, and the image with all applied augmentations.

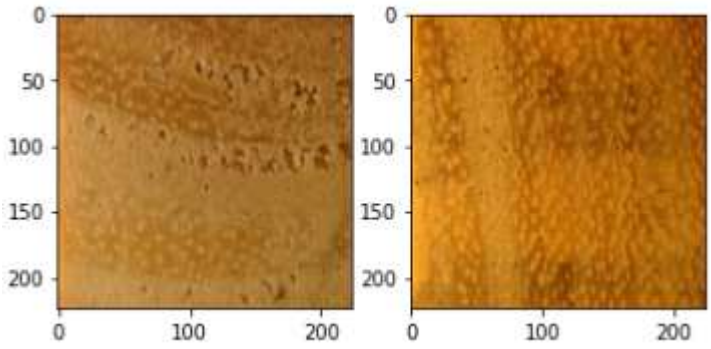

Figure 1. Images of class 1

Augmentation pipeline and their probability for train dataset:

- Vertical Flip (0.5)
- Horizontal Flip (0.5)
- Rotate (0.5)
- Shift in hue, saturation and value in HSV system (0.25)
- Random brightness and contrast (0.5)
- Normalization
- Convert image to pytorch tensor

For the validation dataset, only normalization and translation to pytorch tensor are applied. For the test set, you can use the validation set of images of the displayed images along the diagonal.

The classes are not balanced in the dataset. The maximum number of images by class is 99, and the minimum is 37. The excess of one of the classes over the other is slightly more than 2 times.

2. Modeling

Since neural networks on the first layers study low-level filters to highlight small patterns, such as lines, then using already pre-trained neural networks on Imagent can not only reduce the training time, but also give a noticeable increase in quality relative to training the model from scratch. Here is used models trained on the imagenet dataset. Images in imagent are colored and have a size of 224x224. To test ideas, is taken the architecture already built and pre-trained in torchvision of widely known, deep enough (since the dataset is quite large) and popular networks Resnet50 [2] and DenseNet [3] using the skip connections trick.

I divide the dataset into training and validation, and use the augmented validation set as a holdout set to test the final quality of the model. For validation, 20% random images from the dataset is used while maintaining the distribution of classes between the validation set and the training set.

For training models the loss function was chosen (by trial and error approach) - Focal Loss presented by Facebook [4], which draws the model's attention to difficult instances for classification or to those classes that are few. For this loss function, a $\gamma = 2$ was chosen and a balancing factor $\alpha = 0.5$.

Similarly, the AdamW optimizer [5] was chosen, which gives slightly better convergence. Optimizer parameters: initial learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.01. Above the optimizer, the recently presented lookahead algorithm [6] was used, which reduces the variance in gradients, accelerates convergence, and finds better local minima in the loss space. The number of

iterations of the movement of the fast weights is 5, the update of the parameters has an weight $\alpha = 0.5$.

To prevent model stagnation, a scheduler is used that reduces the learning rate by 10 times if the validation losses do not decrease for 3 consecutive epochs. In order not to miss the most successful local minimum, the quality is tracked on the validation dataset and the model is saved with the new highest quality.

The Resnet50 and DenseNet models were trained with the same parameters for 50 epochs and a batch size of 64 images. In the Resnet architecture, only the last block with the skip connection and the last fully connected layer were trained. In the DenseNet architecture, the last block, the normalization layer and the last fully connected layer were trained.

Figure 2, 3 shows the learning curves of the Resnet50 and DenseNet model.

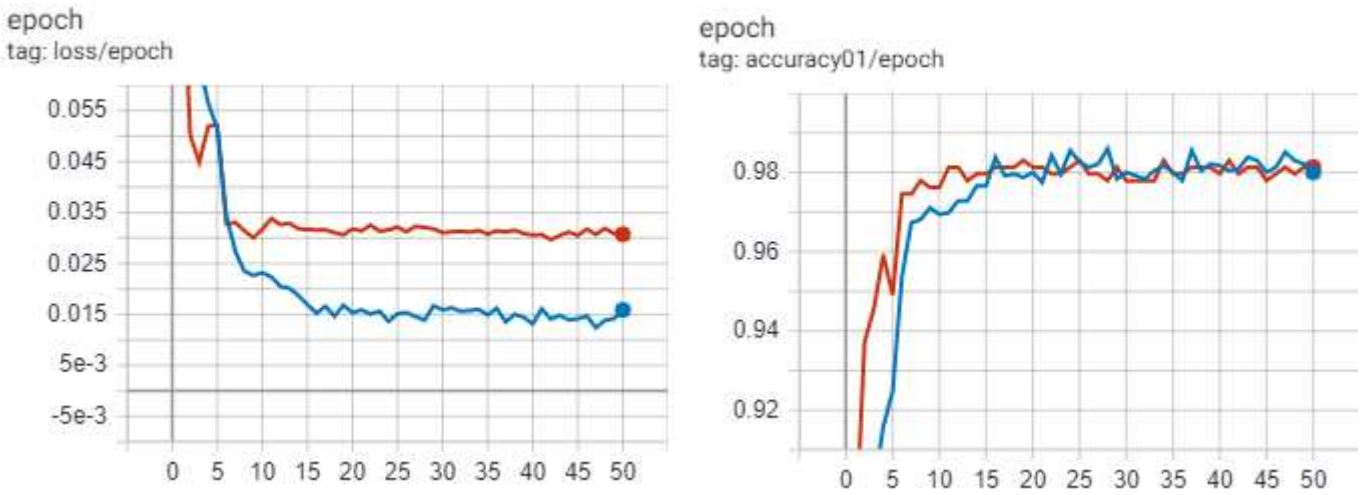

Figure 2. Learning curve for Resnet50. Blue – train, Red - validation

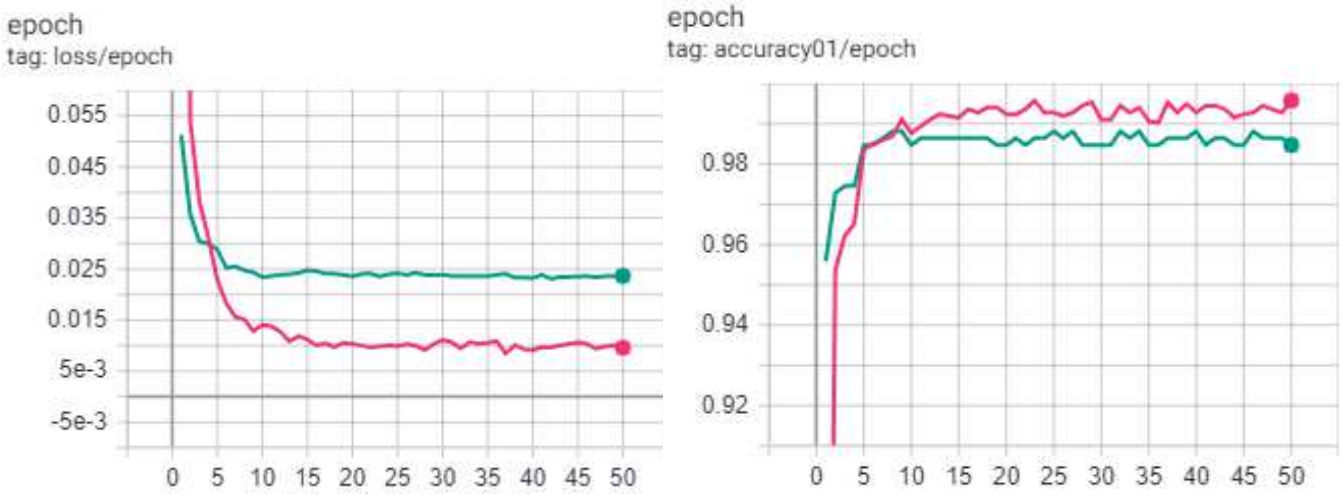

Figure 3. Learning curve for DenseNet. Purple –train, blue - validation

F1 score and ROC AUC metrics can be used to assess the quality of the model for unbalanced classes.

Results of the final models on the test dataset:

- Restent50, Accuracy: 0.983, F1 score: 0.983, ROC AUC: 0.997
- DenseNet, Accuracy: 0.988, F1 score: 0.989, ROC AUC: 0.998

3. Knowledge distillation

Since the resulting models take up a lot of memory and the inference time is quite large, can be done distillation of the knowledge [7] from the resulting ensemble of models (ensemble by using soft voting method) into some simpler model, preferably also pre-trained on the imagenet. For this, the AlexNet network [8] was chosen and trained on softmax responses of an ensemble of models with a temperature of 1.5. The best resulting model has an F1 score of 0.96-0.972.

4. Results

As a result of the project, the ResNet, DenseNet and AlexNet models were trained. All results are shown in Table 1.

Table 1

Model	Accuracy	F1 score	Inference time, s
ResNet50	0.983	0.983	1.58
DenseNet	0.988	0.989	3.53
AlexNet after KD	0.96-0.971	0.96-0.971	0.55

5. References

[1] Pedro L. Paula Filho, Luiz S. Oliveira, Silvana Nisgoski, and Alceu S. Britto. (2014) Forest species recognition using macroscopic images. Machine Vision and Applications, 25(4):1019–1031

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015). Deep Residual Learning for Image Recognition. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1512.03385

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, & Kilian Q. Weinberger. (2016). Densely Connected Convolutional Networks. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1608.06993

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, & Piotr Dollár. (2017). Focal Loss for Dense Object Detection. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1708.02002

[5] Ilya Loshchilov, & Frank Hutter. (2019). Decoupled Weight Decay Regularization. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1711.05101

[6] Michael R. Zhang, James Lucas, Geoffrey Hinton, & Jimmy Ba. (2019). Lookahead Optimizer: k steps forward, 1 step back. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1907.08610

[7] Geoffrey Hinton, Oriol Vinyals, & Jeff Dean. (2015). Distilling the Knowledge in a Neural Network. [online database] [Cited 2020-12-12]. Available: https://arxiv.org/abs/1503.02531

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105). Curran Associates, Inc.