Another attempt to shorten the training

Calvin Smyk
Computer Science Graduate Student
1415 Charlotte Street, Tallahassee FL
cs22bh@fsu.edu

Abstract

In this article, a new approach is presented to accelerate the training of neural networks. This approach deals with a technique for preprocessing image data, which is later used for training deep neural networks. To check if there is a training success especially in the time aspect, the performance of this approach is compared with a meanwhile very standardized approach using the CIFAR100 dataset. The goal is to shorten the training by using less image data to achieve the same result.

1. Introduction

Preprocessing a data set is one of the most crucial steps in training a deep neural network. Data preprocessing is not only useful for getting the data into the correct form and format, but through many other interventions, it also has a lot of influence on how well the network will perform in the end. A good pre-processing is therefore substantial and, especially in the field of image processing, has become a very well-rehearsed pattern, which has also proven itself over the past years. In this article, we will address this preprocessing and propose how image preprocessing can be further improved, especially with respect to the training time of the deep neural network. Before the presentation of the approach in more detail, the most important terms that are necessary to understand are explained. Therefore, the dataset, Cifar100, is presented first, followed by the chosen network architecture. It is also important to understand what image pre-processing is exactly and what it involves.
Cifar100 is a dataset that consists of 100 classes containing 600 images each. The dataset is split into two parts: a training set of 50,000 images and a test set of 10,000 images. The classes in Cifar100 are grouped into 20 coarse-grained classes and 100 fine-grained classes. The coarse-grained classes are high-level categories such as "animals" or "vehicles," while the fine-grained classes are more specific, such as "beaver" or "limousine." The dataset was created by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton at the University of Toronto. It is widely used in the field of machine learning for object recognition tasks. [1]

ResNet101 is a deep neural network architecture that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015. It was developed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun at Microsoft Research.
ResNet101 is a convolutional neural network (CNN) that is 101 layers deep. This means that it has 101 layers of learnable parameters, including the input layer and the output layer. The depth of a neural network is an important factor in its performance, and deeper networks are generally able to learn more complex patterns than shallow networks.
ResNet101 uses skip connections, which are connections that bypass one or more layers in the network. This allows the network to learn features at different scales and helps prevent the vanishing gradient problem, where the gradients of the weights become very small and the network is unable to learn effectively.
ResNet101 has achieved impressive performance on a variety of tasks, including image classification, object detection, and semantic segmentation.
Image preprocessing refers to the set of steps that are performed on image data before it is fed into a machine learning model. These steps typically involve cleaning and formatting the data, and may also include normalization, transformation, and feature extraction. The goal of image preprocessing is to make the data more suitable for the model, which can improve the model's performance and reduce the chances of overfitting.
Some common image preprocessing steps include cropping, resizing, and color space conversion.

For example, an image may be cropped to remove background information that is not relevant to the task at hand, or it may be resized to a standard size that is expected by the model. Additionally, the color space of an image may be converted from RGB to grayscale in order to reduce the amount of data that the model needs to process.
Other preprocessing steps may involve more advanced techniques, such as edge detection or feature extraction. These steps can help the model to focus on the most important parts of the image and learn more effectively.
Overall, image preprocessing is an important step in the machine learning process, and can have a significant impact on the performance of the model. It is therefore important to carefully select and apply the appropriate preprocessing steps to each dataset.

2. Relevant Works

Image Fusion, Theories, Techniques and Applications by H. B. Mitchell [2] is a book that inspired me a lot, as it deals with the different metrics that can be used to compare images while refraining from comparing every pixel. Some of these metrics are also used in this article to evaluate the images.
Furthermore, Kaiming He and Xiangyu Zhang's [2] paper Deep Residual Learning for Image Recognition, which deals with an illustrative approach for training a deep neural network using the CIFAR100 dataset, also helped me to get more involved in this topic.

3. Proposed System

In this approach, the standardized process for classifying image data is upgraded. For this purpose, not the network architecture is changed, but the preprocessing. In the standardized process, the images are simply cropped using a random crop, flipped horizontally, normalized between 0 and 1, and translated into the correct data structure. These steps can be summarized under the term data transformation. In the proposed approach, data preprocessing is applied in addition to data transformation.

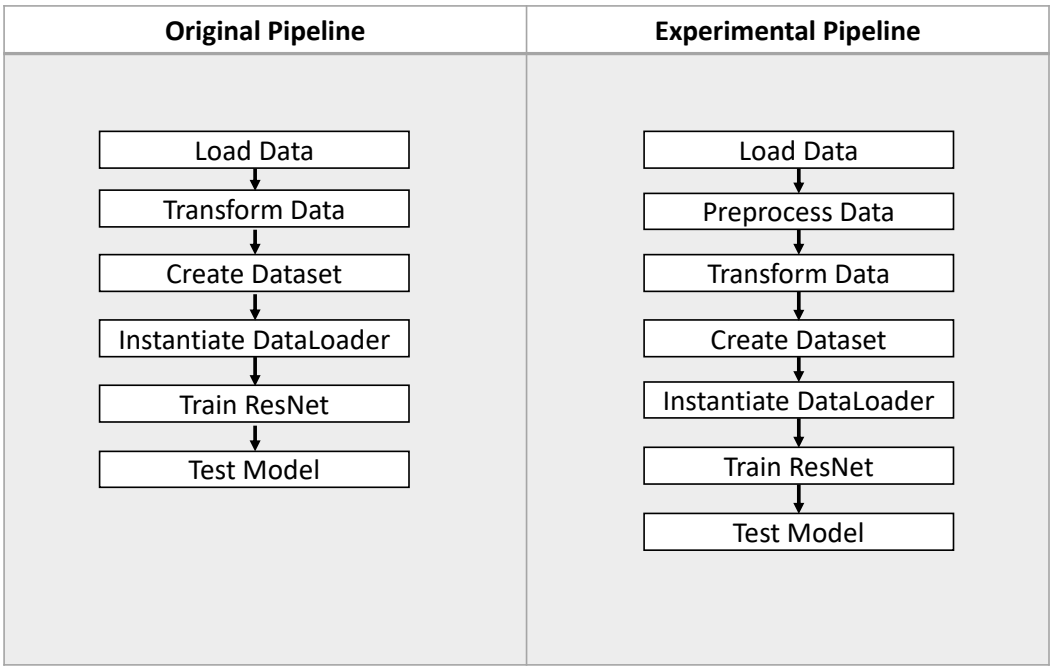

Abbildung 1: Comparison of the Pipelines

Graphic 2 shows the individual steps that are added in the data preprocessing part.

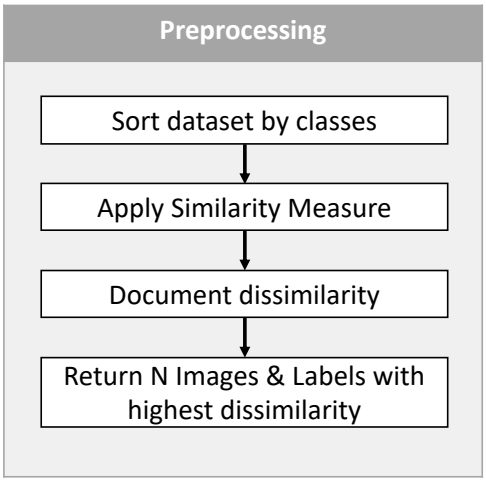

Abbildung 2: Details of the Preprocessing

More precisely, first the already presented dataset is sorted by its classes to be able to work specifically within the class. Then, the images are compared with each other using different similarity measurements. For this, first an image is selected, which is considered to be the reference image, which is then compared with all other images. The selected similarity measurement then provides information about which image has the greatest similarities. The idea is that we will address whether all 500 images per class are really necessary to train a good model, or whether there are also images in this dataset that are so similar that it does not really add value to include these images in the training process. For this purpose, different metrics are first selected, which are able to compare images with each other in order to be able to make a statement about which images contain a lot of new information and which do not.

I. Similarity measures

Similarity measures are mathematical methods that are used to determine how similar two objects are to each other. These measures are commonly used in image preprocessing, where they can be used to compare different images and select the most similar ones for further processing.
One example of a similarity measure is the Euclidean distance, which calculates the distance between two points in a multi-dimensional space. This measure is often used in image preprocessing to compare the pixel values of two images and determine how similar they are.
Overall, similarity measures are useful tools for image preprocessing, as they can help to identify similar images and select the most relevant ones for further processing. This can improve the performance of the machine learning model and reduce the chances of overfitting. Here the similarity measures will be used to select the most dissimilar pictures and use these for training. [3]

i. Root mean squared error

Root mean squared error (RMSE) is a measure of the difference between the predicted values of a model and the true values of the target variable. It is commonly used in regression analysis to evaluate the performance of a model, and can be calculated as the square root of the mean squared error (MSE) between the predicted and true values. The MSE is calculated by taking the difference between the predicted and true values for each data point, squaring the difference, and then taking the average of the squared differences. The RMSE is then calculated by taking the square root of the MSE.
Here, it will be used as a measure to compare two images. The input will be two images, the reference image and the image to be compared, as numpy arrays in the shape of [32,32,3], meaning we will compare the image over all three color channels.

ii. Peek signal to noise ratio

Peak signal-to-noise ratio (PSNR) is a measure of the quality of a reconstructed signal, compared to the original signal. It is commonly used in image and video processing to evaluate the performance of image and video compression algorithms, and can be calculated as the ratio of the maximum signal power to the power of the noise. The PSNR is calculated by taking the logarithm of the maximum signal power, divided by the mean squared error (MSE) between the original and reconstructed signals. In our case the maximum signal power correlates to the maximum number of pixels, which equals to 32x32.

iii. Entropy_Histogram_Similarity

Entropy is a measure of disorder or randomness in a system. A histogram is a graphical representation of data showing the distribution of values within a given range. The similarity between two entropy histograms are a measure of how similar the level of disorder or randomness is in the two systems being compared. We create a 2D histogram for the two images, which represents the probability that a pixel intensity value from the reference image cooccurs with pixel intensity value from the image to be compared to.

iv. Signal to Reconstruction Error Ratio

The signal to reconstruction error ratio, also known as the signal-to-distortion ratio (SDR), is a measure of the quality of a reconstructed signal. It is defined as the ratio of the energy of the original signal to the energy of the error signal, which is the difference between the original and the reconstructed signals. A higher SDR indicates a higher quality of reconstruction, while a lower SDR indicates a lower quality of reconstruction.

4. Analysis

The first step is to use the standardized process to get a benchmark of how well the new approach actually works. To do this, it is important to define what should be compared to assess the success of the approach.
One of the best known and most important metrics for judging a model is Accuracy, which in a multi-class problem is expressed by Precision. This indicates how many images the network was able to assign to the correct class. More precisely, it is:

$$\frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

Another relevant metric is the loss, which provides information about how bad (how far off) the model prediction was.
However, the goal is to find out if the model can achieve the same training success with less image data in the same time. Therefore, the number of epochs is used in conjunction with the precision to assess how fast the model learns. Since the model shows the greatest training success in the first 50% of the training, in this approach we will only use the first 25 epochs to evaluate the training success. It should be noted that the model will by no means have finished the training after 25 epochs, but that at this point it will be clear whether the new approach is having the desired effect or not.

The second step, before all the different similarity measurements are used for data preprocessing, is to determine which data set size to train with. The standardized process, uses all 500 images per class for training, which translates to a dataset size of 50,000 images.
In our approach, we will drastically reduce the dataset size to only 40,000 and 30,000 images, that is, 400 and 300 images per class. Depending on how much the performance decreases, we will then determine which dataset size to train with. For the two experiments, the root mean squared error (RMSE) is used to select the images to be eliminated.

Dataset size	Loss @ 10	Precision @ 10	Loss @ 20	Precision @ 20	Loss @ 25	Precision @ 25
50.000	2.252	43.71	1.747	52.59	1.566	56.88
40.000	2.205	45.43	1.772	48.35	2.070	58.14
30.000	2.480	36.36	2.279	47.40	2.148	48.05

Abbildung 3: Result of Experiments with Dataset size

As you can see in chart 3, the difference between 50,000 and 40,000 is very small, especially if you look at the precision after 25 epochs, you can see that the data set with only 40,000 images is even a little higher here, but the loss is also higher. The attempt where the data set is reduced with the preprocessing strategy to only 30,000 images does not seem to work so well. Already after the first 10 epochs, the other two models are significantly more advanced. Thus, the decision is made to set the dataset size for all similarity measurements to 40,000.

The last step is to train models for all similarity measurements, keeping all hyperparameters like batch size, learning rate and number of epochs constant. In order to compare the results, a graph is created which shows the precision of the models over the number of epochs.

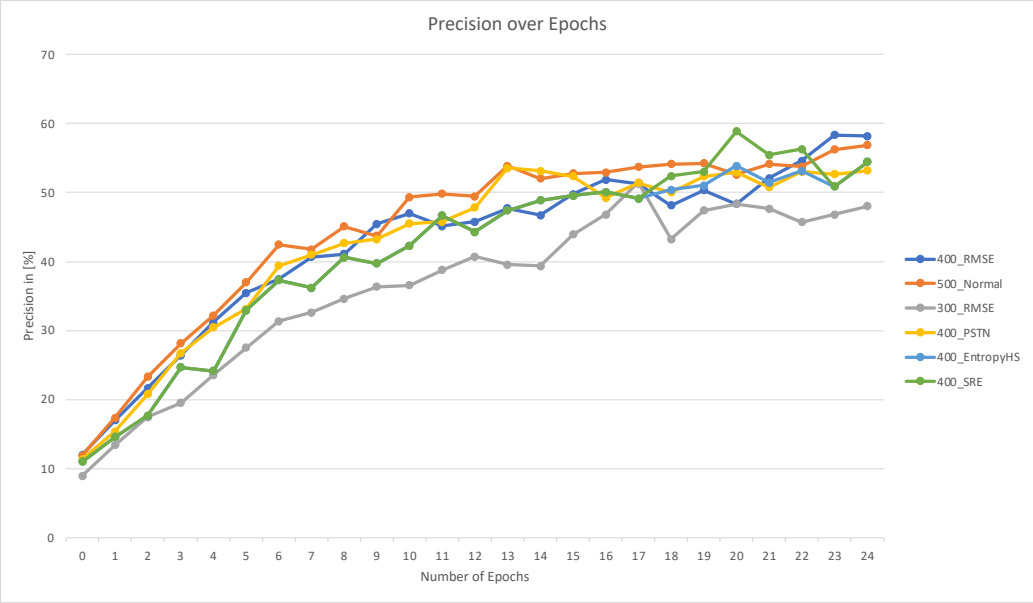

Abbildung 4: Precision over epochs for all Similarity measurement experiments

Analyzing this graph, one notices that the results are all very close to each other. The only line that clearly deviates from the rest is the one that symbolizes the experiment with the dataset size of 30,000 images. This confirms the decision that a reduction of 40% is too much. Looking at the other lines, we can see how the learning success is already slowly approaching its plateau, since the slope from epoch 10 onwards does not increase as much as before. Initially, the slope of the standardized process is very similar to the approaches where the RMSE and PSTN were used for preprocessing. The approach where the SRE was used, instead, has a longer strong learning phase at the beginning, which allows this approach to catch up with the others again at epoch 11. Although the training success of the standardized approach always seems to finish best, there are several models that achieve significantly higher values for precision at certain time points. For example, the SRE approach achieves a Precision value of almost 60% at epoch 20, which is the highest of all approaches for all 25 epochs. This is not negligible, because the model is always stored where the test precision is highest, which means that at the time of epoch 20 the model for the SRE was saved and the weights of the individual layers are stored exactly in this constellation, until the time that at an epoch a higher precision is achieved. In the same way, the RMSE approach ends at epoch 25 with a higher precision than the standardized approach.

5. Conclusion

In conclusion, the research conducted in this paper has shown that the proposed strategy is effective in helping with the training of a deep neural network, since we were able to show that we can reach the same or even better training success with a smaller dataset, which correlates to less time during training. Through our experiments, we have demonstrated that some of the proposed similarity measures used in the preprocessing pipeline achieve a higher precision and outperforms the standardized approach in terms of training time. This validates the idea that deep neural networks may well derive little or no benefit from adding images to training that look very similar to other images. Furthermore, the proposed preprocessing strategy has been

shown to have several key advantages, including filtering the data for unnecessities and reducing the duration of time that is necessary for training a deep neural network. Overall, the findings suggest that the preprocessing strategy has potential for real-world applications and warrants further more detailed investigation.

6. Future work

For the future, there are many different ways to continue this work. For example, one can further extend the preprocessing by developing a methodology that generates the reference image from all the images and not simply selecting one arbitrary image as the reference image.
Also, a dataset containing 50,000 images is not a large dataset, so it would also be interesting to see how much one can reduce the dataset which does not have this complexity, i.e. 500 images for 100 classes, but perhaps 5000 images for 10 classes. There could be more room for improvement in the reduction of the dataset.
Also, this approach was only tested for the first 25 epochs, it might also be interesting to see how the training success develops over an entire training with 200 epochs.
Another possibility would be to apply the approach of filtering out unnecessary information from datasets also in Natural Language Processing, by not only assigning stronger weights to words, but by removing certain words completely from the context.

7. References

[1] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.
https://www.cs.toronto.edu/~kriz/cifar.html

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[3] Mitchell, H.B. (2010). Image Similarity Measures. In: Image Fusion. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-642-11216-4_14

[4] Müller, M. U., Ekhtiari, N., Almeida, R. M., and Rieke, C.: SUPER-RESOLUTION OF MULTISPECTRAL SATELLITE IMAGES USING CONVOLUTIONAL NEURAL NETWORKS, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-1-2020, 33–40, https://doi.org/10.5194/isprs-annals-V-1-2020-33-2020, 2020.