MULTI-LABEL CLASSIFICATION OF IMAGE DATA

Nariman Zendehrooh
Student ID: 260700556
McGill University
nariman.zendehroohkermani@mail.mcgill.ca

Javier Tobar
Student ID: 260868593
McGill University
javier.tobar@mail.mcgill.ca

Marjan Manosurian
Student ID:260978482
McGill University
marjan.mansourian@mail.mcgill.ca

ABSTRACT

In this project, we implemented two convolutional neural networks for image classification on a modified MNIST dataset. The task is to identify every digit individually from an image that contains between 1-5 digits. We found that using two neural networks together yielded an accuracy of 99.73% on the validation set and 99.97% on the training set.

1 Introduction

We divided this project into two sub-tasks: determining the number of digits in an image and determining the value of a single digit. Our first CNN was used to predict the number of digits present in an image, we then used its prediction to split each digit into a 1x12x12 dimension. Our second CNN was used to predict the value of a digit in a 1x12x12 dimension. We had to achieve high accuracy on our first CNN because our second CNN depended on it.

2 Datasets

We first started by removing the padding from the images to create an image with 13x60 dimensions. Then, we rescaled the images to convert them from RGB to greyscale. To train our first CNN, we had to map our labels to the number of digits that weren't "10", e.g. [5,4,7,10,10] would've been mapped to [3]. Once we knew how many digits were present in the image, we could split the image into sub-images of 1x12x12 dimensions where each one was containing a digit from the original image.

3 Results

For our first CNN, we managed to achieve 100% accuracy on the validation set and 99.25% on the training set. For our second CNN, we managed to achieve 99.91% accuracy on the validation set and 99.94% on the training set. Together, for the main task, we obtained an accuracy of 99.73% on the validation set and 99.97% on the training set.

4 Creativity and Originality

We decided to standardize the inputs of each mini-batch using batch normalization. This method helped us to stabilize the learning process and to reduce the number of training epochs required to train our models.
Also, instead of training a single neural network to predict each image in one shot, we decided to train two models so that we can first predict the number of digits in each image and then predict a single digit. This method highly improved our predictions and resulted in high accuracy.

5 Discussion and Conclusion

We decided to use a convolutional neural network for the following reasons: our main task is image recognition and we have a lot of training data to successfully optimize our weights. A lot of our decisions were based on avoiding computationally expensive methods while trying to maintain an acceptable accuracy in our models. For instance, Sigmoid is more computationally expensive than ReLu, furthermore, ReLu also reduces the vanishing gradient problem. Thus, we opted for ReLu over Sigmoid. For our optimizer, initially, we were using stochastic gradient descent, however, we decided to use Adam because the difference in accuracy was negligible despite the noticeable decrease in training time.
We were also considering using another machine learning model to predict the number of digits in an image as opposed to using another CNN. However, the performance of our first CNN was very satisfactory so we opted against implementing another model. We're confident that CNN is the best suited neural network for our task, but it still would've been interesting to see how it compared with ANN and RNN in terms of accuracy and training speeds.

6 Statement of Contributions

Javier was in charge of implementing both CNNs, making the data compatible with Data Loader and co-writing the report.
Nariman was in charge of load and pre-processing of the datasets, code review, refactoring and optimizing the CNNs and co-writing the report.

Marjan was in charge to prepare the references.

References

1. Synced. (2019, March 07). ICLR 2019: 'Fast as Adam Good as SGD'- New Optimizer Has Both. Retrieved December 13, 2020, from https://medium.com/syncedreview/iclr-2019-fast-as-adam-good-as-sgd-new-optimizer-has-both-78e37e8f9a34

2. Sebastian Ruder. (2020, March 20). An overview of gradient descent optimization algorithms. Retrieved December 13, 2020, from https://ruder.io/optimizing-gradient-descent/

3. Sebastian Ruder. (2020, March 20). An overview of gradient descent optimization algorithms. Retrieved December 13, 2020, from https://ruder.io/optimizing-gradient-descent/