Mini-Project 1: Image Classification

Comparing two digits from two-channel image

Adrian Guerra Ayyoub El Amrani Jan Benzing
EE-559 Deep Learning 2019, EPFL

***Abstract*—The goal of this project is to implement, test and compare several deep learning approaches and architectures in order to classify the digits of a two-channel image. We are given a data set containing several samples of a two-channel image with a binary target (0 if the first digit is less than or equal to the second digit and 1 otherwise). In this project, we mainly focus on four different architectures relying on Convolutional Neural Networks. This project is part of "EE-559 Deep Learning" course taught at EPFL during the spring semester 2019.**

I. Introduction

Image Classification, the task of assigning an input image one label from a fixed set of categories, has always been a significant problem in Computer Vision and Machine Learning. It may seem like a basic task, but this sort of problem has a large variety of practical applications. Convolutional Neural Networks for image classification have for instance been used by autonomous vehicles to perform road traffic sign detection and classification [1]. Another example is the detection and segmentation of photovoltaic solar panels from aerial images using Convolutional Neural Networks [2] enabling the compilation of large databases of great social and economical use, like the *Open PV project*, a comprehensive online database of photovoltaic installation and market data in the United States.

Our task is to compare two digits from the modified National Institute of Standards and Technology (MNIST) Database of Handwritten Digit Images for Machine Learning Research, a collection of handwritten digit images used extensively in optical character recognition and machine learning research, and predict which one is larger.

Originally, all black and white digits are size normalized, and centered in a fixed-size image where the center of gravity of the intensity lies at the center of the image with 28×28 pixels. Thus, the dimensionality of each image sample vector is $28 \times 28 = 784$, where each element is binary [3]. 2D average-pooling is applied on the original images to produce 14×14 images and our project is based on these transformed images.

Convolutional Neural Networks (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery including image classification. Compared to standard multi-layer perceptrons with similarly-sized layers, CNNs have much fewer connections and parameters and so are easier to train. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies) [4].

In fact, the record performance of image classification on the MNIST data set reaches about 0.2% error rate and is achieved by a committee of convolutional nets (with elastic distortion in augmenting the training set) [5].

We propose various Convolutional Neural Networks that automate the classification of digits and that use Weight Sharing and Auxiliary Losses to predict the larger digit from two-channel images. This paper first explores the design of our architectures, the methodology of the experiments performed and finally discusses the results.

II. Design

Our task is to compare two digits visible in a two-channel image and predict which one is larger. Performing an accurate image classification requires exploring several architectures aiming to improve the test error estimate. Weight Sharing and Auxiliary Losses are two techniques that we explore.

1) Weight sharing: Weight sharing is the sharing of weights between two layers in the model. This means that the same parameters will be used to represent two different transformations in the system. This design choice fits perfectly with the nature of the problem since we effectively have two similar images as input. Weight sharing helps the model generalize better, especially with the "Siamese-like" model we use, see Figure 2 and 3.

2) Auxiliary loss: An auxiliary loss is a loss computed by taking a weighted average of different losses. Since we have three classifiers, we naturally have three losses. In order to perform the back-propagation, we choose to either use an auxiliary loss to back-propagate only on the principal loss (based on the ordering digit).

A. General encoder architecture

In this section, we describe more carefully the different architectures explored in order to perform the desired task. The four different Siamese-like architectures use the same general encoder but in different ways. Our general encoder architecture is described in Figure 1. This encoder uses convolution layers with ReLu and max-pooling. Additionally, we add batch-normalization in order to prevent a vanishing gradient and dropout to prevent over-fitting. The output of our encoder is a feature vector of size 64 that is ready to feed classifiers.

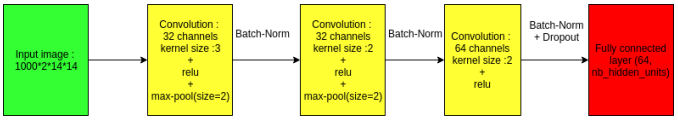

Fig. 1: Encoder architecture

B. Architectures

Architecture	Weight sharing	Auxiliary loss
1	Yes	Yes
2	Yes	No
3	No	Yes
4	No	No

TABLE I: Architectures implemented

Our main target is the prediction of the ordering between the first and second image. We use three classifiers. One for the first digit (10 classes), one for the second digit (10 classes) and a last one for the ordering prediction (2 classes). The first two classifiers are supports used for manually predicting the main target. The architectures use in different ways these two concepts:

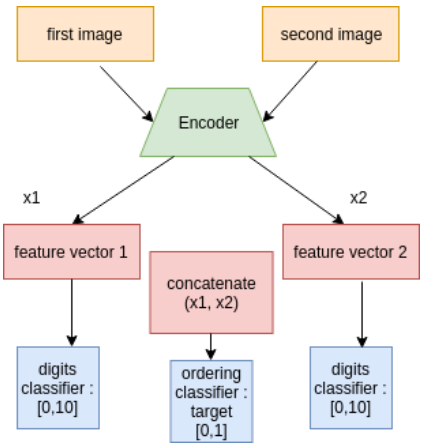

Fig. 2: Weight sharing architecture

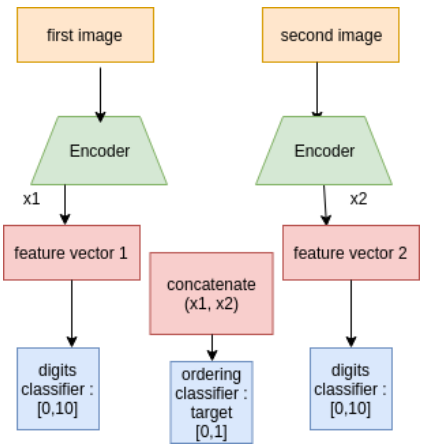

Fig. 3: Without Weight sharing architecture

1) Classifiers: Our three classifiers use Cross-entropy loss. the digits classifier have an output of 10 (10 classes of digits) and the principal classifier have an output of 2 (1 or 0 with respect to the ordering). Each classifier adds one fully connected hidden layer and a sigmoid activation.

III. Methodology

To perform the desired task. We evaluate the four architectures with two different accuracies. We compute an accuracy based on a manual comparison of the predicted digits and one based on the ordering prediction directly. Concerning Auxiliary loss, we can omit it by simply setting the loss's weight of digit predictions to 0.

IV. Results

	Acc 1	Acc 2	Weight Sharing	Auxiliary Loss	σ_1^2	σ_2^2
1	**93.1%**	83.7%	Yes	Yes	0.39	0.81
2	63.1%	82.2%	Yes	No	12.07	1.11
3	77.6%	64.5%	No	Yes	0.25	0.78
4	43.0%	77%	No	No	9.48	1.19

TABLE II: Results Table.

Accuracy 1 and σ_1^2 are based on digit prediction, while Accuracy 2 and σ_2^2 are based on the main target (ordering prediction).

The results show us that using weight sharing and the auxiliary loss give the best accuracy.

As previously mentioned, a two-channel image input fits perfectly with a model allowing weight sharing as the two images are generally similar and only differ on the digit they represent. Weight sharing not only makes the prediction more consistent and stable but also allows the training to be more optimal since less weights are trained compared to a model where weight sharing is absent.

Concerning the auxiliary loss, we notice that it helps the model give a better prediction. This is due to a part of the digit prediction loss being back-propagated through the network. The model appears to be learning from the digits prediction which adds information that help the model predict its main target.

Finally, we notice that the accuracy computed after manually comparing the predicted digits is visibly higher than the accuracy based on the main target. In fact, manual comparison of digits doesn't contain any error as long as the digit prediction is accurate enough. This makes sense since classifiers tend to be excellent at predicting digits while classifying directly the ordering can induce more implicit errors.

V. Conclusion

Weight sharing performs very well, on top of its memory optimization, and auxiliary loss seems to be very helpful in training the model and comparing approaches. All in all, this project made us aware of several ways to tackle classification problems. Before naively attacking the problem, an analysis of our data set along with a careful procedure on how to approach the problem should be made in order to get the most optimal results. Deep learning frameworks offer a panel of layers and

facilities that allow us to implement a wide array of different architectures. Having a balance between creativity and rigor while playing with these layers have proved to make enormous improvements. Although not explored in this project, augmenting the data set by performing label-preserving transformations is a common practice in the analysis of visual imagery and would have almost certainly yielded better results [6].

REFERENCES

[1] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," *IEEE transactions on industrial electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[2] J. Yuan, H. L. Yang, O. A. Omitaomu, and B. L. Bhaduri, "Large-scale solar panel mapping from aerial images using deep convolutional networks," in *2016 IEEE International Conference on Big Data (Big Data)*, Dec 2016, pp. 2703–2708.

[3] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov 2012.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *2011 International Conference on Document Analysis and Recognition*, Sep. 2011, pp. 1135–1139.

[6] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *null*. IEEE, 2003, p. 958.