Evaluation of Sum Product Networks on Image Classification Tasks

David Bonanno[1], Sambit Bhattacharya[2], Leslie Smith[1], & David W. Aha[1]
[1]Information Technology Division (Code 5500); Naval Research Laboratory; Washington, DC
[2]Department of Mathematics and Computer Science; Fayetteville State University; Fayetteville NC
{*first.last*}@nrl.navy.mil | sbhattac@uncfsu.edu

Abstract

We present a method for classifying imagery using Sum-Product Networks. Current techniques allow for the architecture to be learned in addition to learning the weights between nodes, resulting in high accuracy classification without the need for manual architecture specifications. Our results show that such networks can perform comparably to current state of the art methods on simple image classification problems. However, applying SPNs to this task requires substantially reducing the sizes of the images, and learning the structure of an SPN for the originally-sized images would be computationally expensive.

1. Introduction

Discriminative Sum-Product Networks (SPNs) are a probabilistic model that allow for direct and indirect relationships between variables to be represented in a deep architecture [1]. Recent advances have allowed for such networks to not only train the weights between nodes of the network, but also create the structure of the network itself. This advance allows for SPNs to be used on a variety of complex classification tasks without the need for the architecture to be defined for each problem.

Recent research has shown that deep networks, such as Convolutional Neural Networks (CNNs) can achieve high accuracy for difficult image classification challenges such as ImageNet [2]. In this report we compare these current methods to SPNs on a simplified image classification problem. These results are compared to other classifiers including k-nearest neighbor (KNN) (k=1) and a CNN approach.

2. Approach

Our goal is to evaluate the performance of SPNs in classifying imagery. Algorithms for deploying SPNs are available in the Libra Toolkit [3]. This includes ID-SPN, which is used to learn both the SPN network and node weights. ID-SPN requires labeled binary data [4]. For this reason, we first needed to create binary image datasets for training, validation, and testing the SPNs.

We compared the performance of SPNs on the modified dataset with other techniques, as mentioned above, using the same training and test sets.

2.1 Dataset Creation

The MNIST dataset contains handwritten numbers 0-9 [5] (i.e., 10 classes of images). The labeled images in this dataset are grayscale imagery, with each image being 28x28 pixels in size. It contains 60,000 training and 10,000 test images.

Each pixel in an image is represented by 8 bits of information, which requires 6272 bits per image. As such it would be possible to create and input a vector of binary information for each image into the SPN algorithms directly. However, the computational requirements and time restraints necessitate data reduction.

First, we reduced the size of the training and test images by 60% using a common bilinear interpolation function. This yields 17x17 images, which we quantized using four techniques.

The first technique we used is Otsu's method for adaptive thresholds [6], which transforms images into greyscale (i.e., white or black pixels, only). This method reduces each image to 289 bits (1 bit per pixel). Additionally, we applied three basis methods to transform the reduced-size gray scale images into a new feature space. These include (1) the Discrete Cosine Transform (DCT) [7], (2) the 2D Haar Wavelet [8], and (3) a set of random bases. These feature vectors were quantized, yielding 289 bits per image (256 for the Haar basis). The raw imagery, as well as the image-domain representation for each of these methods, are shown in Figure 1.

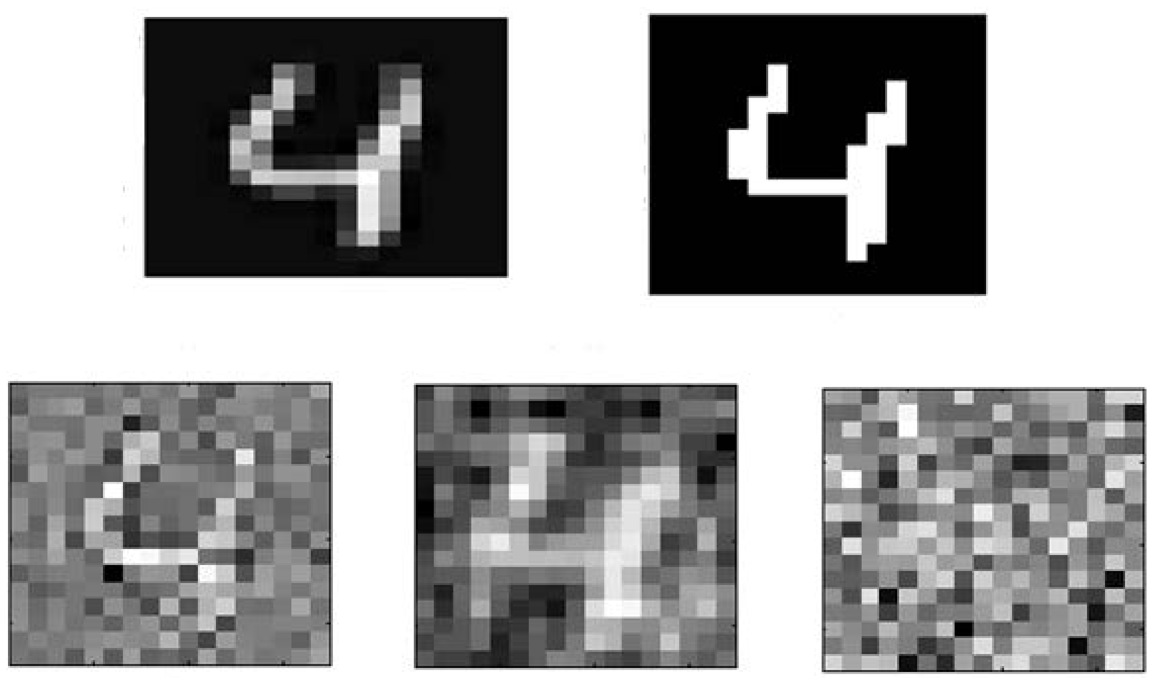

Figure 1: Effects of quantizing an image of the digit "4" using four techniques: (Top left) Grayscale MNIST image; (Top Right) Otsu's Method; (Bottom Left) DCT; (Bottom Center) Haar Wavelet; (Bottom Left) Random Basis.

2.2 Classification using Sum-Product Networks (SPNs)

For each of these four data reduction techniques, we generated ten SPNs (one for each image class). We trained the SPNs on the modified MNIST training data using ID-SPN. These networks each return a single value, which is the output of the highest level.

Once these SPNs were trained, we used them to classify the modified MNIST testing data. A test datum was classified as belonging to the SPN with the highest value at the output.

2.3 Comparison to k-Nearest Neighbor (KNN with k=1)

We compared the results of the SPNs with KNN (k=1) [9]. The results are shown in Table 2. Using Otsu's method for data reduction results in the highest classification accuracies, independent of which classifier was used. Also, the SPNs consistently outperform the KNN (k=1) classifier.

Encoding	% Correct 1-NN	% Correct SPN
Otsu	95.72	97.47
Random	93.47	94.74
DCT	91.88	92.48
Wavelet	92.77	95.88

Table 2: Classification results for each of the data reduction techniques using KNN (k=1) and the trained SPNs.

2.4 Comparison with LeNet CNN

We also applied the LeNet CNN architecture [10] using the modified dataset with Otsu's encoding. We zero-padded the data to fit into LeNet and trained and tested using Caffe [11]. With this implementation, the trained CNN recorded an accuracy of 97.42% on the test data, which is comparable to the SPN approach.

3 Conclusions

Our initial results show that it is possible to classify imagery using SPNs, and that these models perform comparably to other methods on the one task we examined. However, the current implementations we used for learning an SPN's structure and weight settings requires substantial data reduction steps.

4 Acknowledgements

This work was funded as part of the NRL IT Focus Area project *Recursive Structure Learning*.

5 References

[1] Gens, Robert, and Pedro Domingos. "Discriminative learning of sum-product networks." *Advances in Neural Information Processing Systems*. 2012.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012

[3] Lowd, Daniel. “The Libra Toolkit.” *http://libra.cs.uoregon.edu/* University of Oregon. 30 March 2012.

[4] Rooshenas, Amirmohammad, and Daniel Lowd. "Learning sum-product networks with direct and indirect variable interactions." *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.

[5] LeCun, Yann, and Corinna Cortes. "The MNIST database of handwritten digits." (1998).

[6] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms."*Automatica* 11.285-296 (1975): 23-27.

[7] Ahmed, Nasir, T. Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." *Computers, IEEE Transactions on* 100.1 (1974): 90-93.

[8] Raviraj, P., and M. Y. Sanavullah. "The modified 2D-Haar Wavelet Transformation in image compression." *Middle-East Journal of Scientific Research* 2.2 (2007): 73-78.

[9] Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." *The American Statistician* 46.3 (1992): 175-185.

[10] LeCun, Yann, et al. "Gradient-based learning applied to document recognition."*Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. "Caffe: Convolutional architecture for fast feature embedding." In Proceedings of the ACM International Conference on Multimedia, pp. 675-678. ACM, 2014.