Stacked Robust Autoencoder for Classification

J. Mehta, K. Gupta, A. Gogna and A. Majumdar

Indraprastha Institute of Information Technology, Delhi
{mehta1485, kavya1482, anupriyag and angshul}@iiitd.ac.in

Abstract. In this work we propose an l_p-norm data fidelity constraint for training the autoencoder. Usually the Euclidean distance is used for this purpose; we generalize the l_2-norm to the l_p-norm; smaller values of p make the problem robust to outliers. The ensuing optimization problem is solved using the Augmented Lagrangian approach. The proposed l_p -norm Autoencoder has been tested on benchmark deep learning datasets – MNIST, CIFAR-10 and SVHN. We have seen that the proposed robustautoencoder yields better results than the standard autoencoder (l_2-norm) and deep belief network for all of these problems.

Keywords: autoencoder, deep learning, classification, robust estimation

1 Introduction

An autoencoder learns the analysis and the synthesis weights by minimizing the l_2-norm between the input (training samples) and the output (training samples / corrupted training samples). The l_2-norm is perhaps the most widely used data fidelity constraint in signal processing and machine learning. It arises from the Gaussian / Normal assumption of the distribution which fits a large class of problems in practice. But the practical reason behind popularity of the l_2-norm stems from the fact that it is easy to solve; it is smooth and convex and has a closed form solution (for linear problems).

The l_2-norm minimization works when the deviations are small – approximately Normally distributed; but fail when there are large outliers. In statistics there is a large body of literature on robust estimation. The Huber function [1] has been in use for more than half a century in this respect. The Huber function is an approximation of the more recent absolute distance based measures (l_1-norm). Recent studies in robust estimation prefer minimizing the l_1-norm instead of the Huber function [2]-[4]. The l_1-norm does not bloat the distance between the estimate and the outliers and hence is robust.

The problem with minimizing the l_1-norm is computational. However, over the years various techniques have been developed. The earliest known method is based on Simplex [5]; Iterative Reweighted Least Squares [6] used to be another simple yet approximate technique. Other approaches include descent based method introduced by [7] and Maximum Likelihood approach [8].

adfa, p. 1, 2011.

In this work, we propose a generalized l_p -norm autoencoder, for values of p between 0 and 1, l_p -norm fidelity is robust to outliers; l_p -norm is quasi-convex. Unfortunately this makes the problem non-differentiable; hence the standard gradient descent based techniques (e.g. backpropagation) cannot be applied here. One needs to solve it using sub-gradients. We invoke a state-of-the-art optimization approach to solve the ensuing problem; this is called the variable splitting augmented Lagrangian. This reduces our problem to a few simpler sub-problems; one of which needs to be solved using sub-gradients while the rest have an analytic solution. We test our proposed approach with standard autoencoder and deep belief network for benchmark problems in classification; we show that our results are indeed better.

The rest of the paper is organized into several sections. Section 2 describes our proposed approach. The experimental results are shown in section 3. The conclusion of this work is discussed in section 4.

2 Proposed Robust Autoencoder

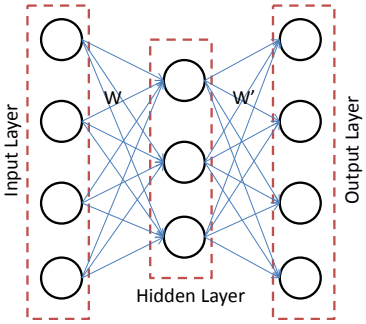

Fig. 1. Basic Autoencoder

An autoencoder consists of two parts (as seen in Figure 1) – the encoder maps the input to a latent space, and the decoder maps the latent representation to the data [9, 10]. For a given input vector (including the bias term) x, the latent space is expressed as:

$$h = \phi(Wx) \tag{1}$$

Here the rows of W are the link weights from all the input nodes to the corresponding latent node. The activation function is usually non-linear (sigmoid / tanh).

The decoder portion reverse maps the latent variables to the data space.

$$x = W'\phi(Wx) \tag{2}$$

Since the data space is assumed to be the space of real numbers, there is no sigmoidal function here.

During training the problem is to learn the encoding and decoding weights – W and W'. In terms of signal processing lingo, W is the analysis operator and W' is the synthesis operator. These are learnt by minimizing the l_2-norm data fidelity constraint:

$$\arg\min_{W,W'} \left\| X - W'\phi(WX) \right\|_F^2 \tag{3}$$

Here $X = [x_1 | ... | x_N]$ consists all the training sampled stacked as columns. The problem (4) is clearly non-convex. But can be solved by gradient descent techniques since the usual activation functionsare smooth and continuously differentiable.

We do not change the autoencoder architecture. We only change the data fidelity constraint from l_2-norm to l_p-norm. This follows from our discussion on robust estimation. The p-norm is more generic and for values of p between 0 and 1; the estimation is more robust. There is a prior study on denoising autoencoders [11] which add noise to samples and then learn a denoising autoencoder; the goal is to learn robust encoding and decoding weights. Although intuitive, this study is at best heuristic. The robustness arising from our proposed formulation is mathematically and statistically optimal. The formulation we propose is:

$$\arg\min_{W,W'} \left\| X - W'\phi(WX) \right\|_p^p \tag{4}$$

The l_p-norm is not differentiable everywhere. Hence gradient based techniques cannot be applied. One need to compute sub-gradient. In this work, we propose to solve this problem using the Augmented Lagrangian approach.

First we substitute, $P = X - W'\phi(WX)$; thus converting (4) to the following,

$$\arg\min_{P,W,W'} \left\| P \right\|_p^p \text{ such that } P = X - W'\phi(WX) \tag{5}$$

The unconstrained Lagrangian is given by,

$$\arg\min_{P,W,W'} \left\| P \right\|_1 + L^T \left(P - \left(X - W'\phi(WX) \right) \right) \tag{6}$$

The Lagrangian imposes equality at every step; this is too stringent a requirement in practice. One can relax the equality constraint initially and enforce it only during convergence. This is the Augmented Lagrangian formulation (7),

$$\arg\min_{P,W,W'} \left\| P \right\|_1 + \lambda \left\| P - \left(X - W'\phi(WX) \right) \right\|_F^2 \tag{7}$$

In the next step, we make another substitution $Z = \phi(WX)$ and write down the Augmented Lagrangian for the same.

$$\arg\min_{P,W,W',Z} \left\| P \right\|_1 + \lambda \left\| P - \left(X - W'Z \right) \right\|_F^2 + \mu \left\| Z - \phi(WX) \right\|_F^2 \tag{8}$$

The problem with the Augmented Lagrangian approach is that, one needs to solve the full problem for every value of λ and μ; and keep on increasing them in order to enforce equality at convergence – this is time consuming. Besides, increasing the values of these hyper-parameters is heuristic. A more elegant approach is to introduce Bregman relaxation variables B_1 and B_2 [12].

$$\arg\min_{P,W,W',Z} \|P\|_1 + \lambda \|P - (X - W'Z) - B_1\|_F^2 + \mu \|Z - \phi(WX) - B_2\|_F^2 \tag{9}$$

Although this problem is not completely separable, we can segregate (9) into alternate minimization of the following subproblems.

$$\text{P1}: \arg\min_{P} \|P\|_1 + \lambda \|P - (X - W'Z) - B_1\|_F^2 \tag{10}$$

$$\text{P2}: \arg\min_{W} \|Z - \phi(WX) - B_2\|_F^2 \equiv \arg\min_{W} \|\phi^{-1}(Z - B_2) - WX\|_F^2 \tag{11}$$

$$\text{P3}: \arg\min_{W'} \|P - (X - W'Z) - B_1\|_F^2 \tag{12}$$

$$\text{P4}: \arg\min_{Z} \|Z - \phi(WX) - B_2\|_F^2 \tag{13}$$

Subproblems P2-P4 are simple linear least squares problems. They have analytic solutions in the form of pseudo-inverse. Subproblem P1 is an l_p-minimization problem. This too has a closed form solution in the form of modified soft thresholding [13], given by.

$$P = signum(X + B_1 - W'Z) \max\left(0, |X + B_1 - W'Z| - \frac{\mu}{2} p |P|^{p-1}\right) \tag{14}$$

The last step is to update the Bregman relaxation variables:

$$B_1 \leftarrow P - (X - W'Z) - B_1 \tag{15}$$

$$B_2 \leftarrow Z - \phi(WX) - B_2 \tag{16}$$

The problem is non-convex thus there is no guarantee of reaching a global optimum. In this case, we continue the iterations till the objective function does not change significantly in subsequent iterations. We also have a cap on the maximum number of iterations; we have kept it to be 50.

3 Experimental Evaluation

Fig. 2. Samples from Datasets. Top – MNIST, Middle – SVHN, Bottom – CIFAR

To test our formulation we used three datasets, MNIST, Street View House Numbers (SVHN) and CIFAR-10. The MNIST dataset is a handwriting recognition dataset developed by Y. LeCun et al. using the larger NIST dataset. It has 60,000 images of handwritten digits, which were used as training images and 10,000 images were used as test images. The SVHN dataset is obtained from Google Street View Images dataset. It also involves recognition of digits, like the MNIST, however it is significantly harder to do so because of clustering of nearby digits and variety of backgrounds. It is a real world problem of recognizing the digits from natural scene images. It is a colored images database, with 73,257 images for training, and 26,032 images for test. There are also 531,131 simpler training images; however we do not use them. We use format 2 of the dataset, which is like the MNIST dataset. The CIFAR-10 dataset was compiled by Alex Krizhevsky et al. from the 80 million tiny images dataset. This dataset contains 50,000 32x32 training images with 10 classes which are mutually exclusive. CIFAR-10 contains images from various categories such as ship, frog, truck and more. This dataset contains 10,000 test images.

Preprocessing

SVHN. We contrast normalize the Y channel of the YUV images of the dataset and use only the Y channel for training and classification. The Y channel is locally contrast normalized using a Gaussian neighborhood, with a 7x7 window. This made the images look more like the MNIST database. The resultant images reside in a $\mathbf{R}^{1024}$ space. From figure 3 we see that the Y channel contains the shape information in a clear and precise manner as compared to the U and V channels. Figure 3c, shows the preprocessed Y channels of the SVHN dataset. We only use the Y channel for training. The same preprocessing is applied to the test set before the classification step.

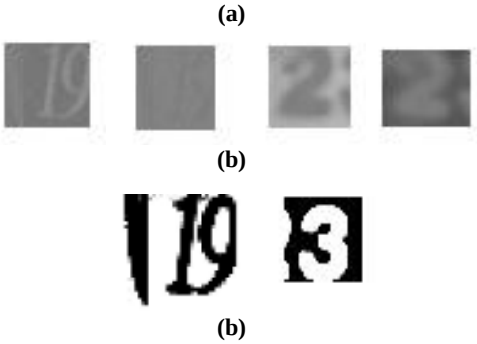

Fig. 3. (a) Y Channels of SVHN, (b) U and V Channels, (c) SVHN Preprocessed Images

CIFAR-10. From each pixel we subtracted the mean of the image for all images in the dataset. This suppressed the brightness variation in the image. The resulting image is converted to greyscale. This is a very challenging problem as mentioned in [14].

MNIST. No preprocessing was required on this dataset.

Results

We show that by using the l_p-autoencoder, we can improve upon the (SAE) standard autoencoder (l_2-norm). The stacked autoencoders (both l_p-norm and l_2-norm) are of three levels; the number of nodes are halved in every successive level.The value of p is kept at 1 for the layers. Other combinations of p might yield better results, but we did not have time to test these. For the sake of comparison we also employ the deep belief network (DBN) for feature extraction; here also the number of nodes is halved in every successive.

We choose to use two non-parametric classifiers KNN and Sparse Representation based Classifier (SRC) [15], and a parametric classifier – SVM with RBF kernel. The SVM was tuned (via grid search) to yield the best results for proposed l_p-autoencoder, SAE and DBN.

Table 1. Classification with KNN (K=1)

Dataset	Proposed	SAE	DBN
MNIST	**97.44**	97.33	97.05
CIFAR-10	**50.01**	45.02	48.49
SVHN	**67.23**	63.93	65.70

Table 2. Classification with SRC

Dataset	Proposed	SAE	DBN
MNIST	**98.36**	98.33	88.43
CIFAR-10	**52.37**	45.11	46.38
SVHN	**69.90**	65.70	66.82

Table 3. Classification with SVM

Dataset	Proposed	SAE	DBN
MNIST	**98.64**	97.05	88.44
CIFAR-10	**53.29**	46.78	48.04
SVHN	**71.19**	66.42	68.01

The results show that the proposed stacked l_p-autoencoder always yields the best results. MNIST is a simple dataset, so the improvement on this dataset is not much. But for other datasets the difference between the proposed robust autoencoder and the non-robust version is significant. This is evident from the results between stacked autoencoder and deep belief network – here the difference in accuracy is marginal – between 1% and 1.5%; on the other hand our method improves over these by 4% or more.

4 Conclusion

In this work we make a fundamental change to the basic autoencoder cost function. Instead of using the popular Euclidean norm to learn the encoding and decoding weights, we propose employing the l_p-norm. For small values of p (less than 1) – this makes the autoencoder more robust to outliers.

Minimizing the l_p-norm is more involved compared to the l_2-norm; this is because unlike the later, l_p-norm is not differentiable everywhere and hence gradient based techniques cannot be applied directly. We solve it using variable splitting and Augmented Lagrangian. This segregates the problem into several sub-problems; one of which needs to be solved using sub-gradient based techniques while the others have simple least squares solutions.

We carry out experiments on three benchmark deep learning datasets – MNIST, CIFAR-10 and SVHN. Three classifiers (KNN, SRC and SVM) were tested upon. In all three cases the proposed method yields the best results compared to the standard l_2-autoencoder and the deep belief network (DBN).

One may get better results by incorporating convolutional techniques into autoencoder [16] and DBN [17]; that is an entirely different direction of research and beating those results is not the goal of this work. Rather, our goal is to show that by moving from a non-robust Euclidean norm to a robust l_p-norm, one can achieve significant improvement in classification accuracy. Our technique does not bar incorporating convolutional techniques in our proposed robust framework; we plan to work on this topic in the future and expect to achieve further improvement in results.

5 Reference

1. P. J. Huber, "Robust Estimation of a Location Parameter", The Annals of Mathematical Statistics, Vol. 35 (1), pp. 73-101, 1964.
2. R. L. Branham Jr., "Alternatives to least squares", Astronomical Journal 87, pp. 928-937, 1982.

3. M. Shi and M. A. Lukas, "An L1 estimation algorithm with degeneracy and linear constraints", Computational Statistics & Data Analysis, Vol. 39 (1), pp. 35-55, 2002.
4. L. Wang, M. D. Gordon and J. Zhu, "Regularized Least Absolute Deviations Regression and an Efficient Algorithm for Parameter Tuning", IEEE ICDM. pp. 690-700, 2006.
5. I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete L1 linear approximation", SIAM Journal on Numerical Analysis, Vol. 10 (5), pp. 839-848, 1973.
6. E. J. Schlossmacher, "An Iterative Technique for Absolute Deviations Curve Fitting", Journal of the American Statistical Association, Vol. 68 (344), pp. 857-859, 1973.
7. G. O. Wesolowsky, "A new descent algorithm for the least absolute value regression problem" Communications in Statistics - Simulation and Computation, Vol. B10 (5), pp. 479-491, 1981.
8. Y. Li and G. R. Arce, "A Maximum Likelihood Approach to Least Absolute Deviation Regression", EURASIP Journal on Applied Signal Processing, Vol. (12), pp. 1762-1769, 2004.
9. D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", Nature, Vol. 323, pp. 533-536, 1986.
10. P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima", Neural Networks, Vol. 2, pp. 53-58, 1989.
11. P. Vincent, H. Larochelle, I. Lajoie Y. Bengio and P. A. Manzagol, “Stacked denoising autoen coders: Learning useful representations in a deep network with a local denoising criterion”, Journal of Machine Learning Research, Vol. 11, pp. 3371-3408, 2010.
12. R. Chartrand, “Nonconvex splitting for regularized low-rank + sparse decomposition”, IEEE Trans. Signal Process., Vol. 60, pp. 5810-5819, 2012.
13. A. Majumdar and R. K. Ward, “On the Choice of Compressed Sensing Priors: An Experimental Study”, Signal Processing: Image Communication, Vol. 27 (9), pp. 1035–1048, 2012.
14. S Rifai, P Vincent, X Muller, X Glorot, Y Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction", ICML 2011.
15. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, “Robust Face Recognition via Sparse Representation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, pp. 210-227, 2009.
16. J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction", ICANN, 2011.
17. H. Lee, R. Grosse and A. Ng. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations", ICML, 2009.