HANDBOOKOF NUMERICAL ANALYSIS

Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 1

Volume Editors Ron Kimmel and Xue-Cheng Tai

NORTH-HOLLAND

Chapter 1

Compressed Learning for Image Classification: A Deep Neural Network Approach

E. Zisselman*, A. Adler[†] and M. Elad^{‡,1}

*Department of Electrical Engineering, Technion Israel Institute of Technology, Haifa, Israel †McGovern Institute for Brain Research, Massachusetts Institute of Technology, Cambridge, MA, United States

[‡]Department of Computer Science, Technion Israel Institute of Technology, Haifa, Israel ¹Corresponding author: e-mail: elad@cs.technion.ac.il

Chapter Outline

1	Introduction	4	4 Performance Evaluation	10
2	Compressed Learning		4.1 MNIST Dataset	10
	Overview	6	4.2 CIFAR10 Dataset	14
	2.1 Compressed Sensing	6	5 Conclusions	16
	2.2 Compressed Learning	8	References	16
3	The Proposed End-to-End CL			
	Approach	9		

ABSTRACT

Compressed learning (CL) is a joint signal processing and machine learning framework for inference from a signal, using a small number of measurements obtained by a linear projection. In this chapter, we review this concept of compressed leaning, which suggests that learning directly in the compressed domain is possible, and with good performance. We experimentally show that the classification accuracy, using an efficient classifier in the compressed domain, can be quite close to the accuracy obtained when operating directly on the original data. Using convolutional neural network for the image classification, we examine the performance of different linear sensing schemes for the data acquisition stage, such as random sensing and PCA projection. Then, we present an end-to-end deep learning approach for CL, in which a network composed of fully connected layers followed by convolutional ones, performs the linear sensing and the nonlinear inference stages simultaneously. During the training phase, both the sensing matrix and the nonlinear inference operator are *jointly* optimized, leading to a suitable sensing matrix and better performance for the overall task of image classification in the compressed domain. The performance of the proposed approach is demonstrated using the MNIST and CIFAR-10 datasets.

Keywords: Compressed learning, Compressed sensing, Sparse coding, Sparse representation, Neural networks, Deep learning.

AMS Classification Code: 68Q32 Computational learning theory

1 INTRODUCTION

Compressed learning (CL) (Calderbank and Jafarpour, 2012) is a mathematical framework that combines compressed sensing (CS) (Candès, 2006; Candès and Wakin, 2008; Donoho, 2006) with machine learning. In contrast to CS, the goal of CL is inference from the signal rather than its reconstruction. In the CL framework, the measurement device acquires the signal by linearly projecting it to a lower dimension, and the inference is performed in this domain directly using machine learning tools.

In many cases, the data we operate on can be assumed to have a sparse representation with respect to a specific dictionary (Elad, 2010). This dictionary could be a fixed one such as wavelet for representing images or learned in order to adapt to the data source (Rubinstein et al., 2010). Compressed sensing leverages this property by replacing the traditional direct and full acquisition of the signals of interest, with their linear projection to a lower dimension. The core idea is that while such projection clearly loses information, the original signals can still be recovered from this limited data due to their inner structure, manifested by their sparse representation. Hence, this approach replaces the traditional set of steps of sensing, compressing, storing, and then decompressing, offering instead a fusion of the sensing and the compression stages. Indeed, compressed sensing provides an efficient sensing, leaving the recovery algorithm with the daunting task of reversing the process and returning back to the data domain. The theory of CS clearly shows that such recovery is practically possible, by providing clear theoretical guarantees for the successful reconstruction of the signal from its measurements (Candès, 2006; Candès and Wakin, 2008; Donoho, 2006).

In many sensing applications, the objective is classification or detection with respect to some signature, instead of a full signal reconstruction. For instance, in radar applications, signal reconstruction is not the true objective, but rather to discern whether the sensed signal is consistent with some target signatures or not. Another application is classification in a data streaming model: assume that a compressed sensing hardware (e.g., a single pixel camera (Duarte et al., 2008; Li et al., 2015)) sends compressed signals to a receiver, which is concerned with the detection of specific signal patterns or anomalies. The natural question to pose in these cases is whether one should first recover the signal form its measurements and then address the decision task or could this detection/classification be done directly on the low-dimensional data.

Recent work touched on the CL concept in various ways, turning it into a practical methodology. Such is the case with the work on reconstruction-free action recognition (Kulkarni and Turaga, 2012, 2016; Lohit et al., 2015), which is an important inference problem in many security and surveillance applications, and compressive watermark detection (Wang et al., 2014) that support data storing and processing in the cloud. There are also biology applications, such as compressive prediction of protein-protein interactions (Zhang et al., 2011), where the authors proposed to analyze the protein's original, high-dimensional sequential feature vector, using compressive measurements. This way, they have reduced the redundancy in the data's feature vector for conserving computations. Another application is compressive hyperspectral image analysis (Hahn et al., 2014), which aims to achieve high classification accuracy using hyperspectral imaging, while avoiding the expensive reconstruction that usually occurs when dealing with immense amount of data. Several works suggested an extension to the original CL framework, such as compressive acquisition of dynamic scenes (Sankaranarayanan et al., 2010), which extends the CS imaging architecture to adapt video scenes and showed good video recovery and classification results. Another example is compressive least-squares regression (Maillard and Munos, 2009), which considered the problem of learning a regression function rather than classification weights from the compressed domain.

In this chapter, we review our own recent results on this topic (Adler et al., 2016, 2017), suggesting that by relying on the data to have a sparse representation, even in some unknown basis, linear compression (i.e., projection) can be used as a beneficial transform, preserving the learnability of the problem under examination, all the while bypassing the computational curse of dimensionality that is prevalent in many machine learning problems. Our approach towards this task leverages deep learning tools, which learn simultaneously the best projection to apply and the decision algorithm that follows.

Broadly speaking, learning directly from the compressed domain is beneficial both from compression and learning aspects. From the compression perspective, it reduces the required storage space and the cost of recovering irrelevant data. From the learning perspective, it mitigates the curse of dimensionality, which can place a huge computational barrier—compromising accuracy or even jeopar-dizing the feasibility of the classification task. Compressed learning can be perceived as a sieve which enables restoration of only the relevant data or even enables to skip the restoration stage altogether while preserving the intrinsic structure of the signal space. This is akin to finding a needle in a compressively sampled haystack without recovering all the hay (Calderbank and Jafarpour, 2012). Compressed learning can therefore be regarded as an efficient universal dimensionality reduction from the original data domain to a more effective and concise subspace, while preserving the intrinsic structure of the data manifold. Hence, CL can be used as a way to reduce the cost of the learning process, while maintaining classifier reliability.

In this chapter, we examine several solutions to compressed learning using a neural network architecture as an efficient classifier and compare their performance. We present an end-to-end deep learning solution (Goodfellow et al., 2016), and the effectiveness of this approach is demonstrated for the task of image classification (Lohit et al., 2016). It is worth mentioning that classification is just one discipline, and this approach can be applied to any other machine learning task. The main novelty of this approach is that the sensing matrix is jointly optimized with the inference operator. This is in contrast to previous approaches, which decouple the choice of the sensing matrices. In our proposed approach, joint optimization during the training stage allows the network to generate compressed representation that fits the dedicated learning task, thus leading to a significant advantage compared with other methods.

This chapter is organized as follows: Section 2 reviews compressed sensing and learning concepts and describes existing CL approaches. In Section 3 the end-to-end deep learning approach is introduced. Then, Section 4 discusses structure and training aspects, while evaluating the performance of the different approaches for compressively classifying images. Finally, Section 5 concludes the chapter and discusses future research directions.

2 COMPRESSED LEARNING OVERVIEW

This section details the principles of compressed sensing and introduces the concept of CL (Calderbank and Jafarpour, 2012).

2.1 Compressed Sensing

Compressed sensing (CS) is a recent, growing field that has attracted substantial attention in signal processing, statistics, computer science and other scientific disciplines. The classic Nyquist–Shannon theorem on sampling continuous-time band-limited signals asserts that signals can be recovered perfectly from a set of uniformly spaced samples, taken at a rate of twice the highest frequency present in the signal of interest. By exploiting this property, much of the signal processing has moved from the analog to the digital domain, creating sensing systems that are more robust, flexible, and costeffective than their analog counterparts. However, in many important real-life applications, the resulting Nyquist rate is so high that it is not viable, or even physically impossible, to build such a device that can acquire in this rate. Despite the rapid growth of computational power, the acquisition and processing of signals in many fields continue to pose a great challenge. Consequently, practical solutions addressing these computational and storage challenges of working with high-dimensional data often rely on compression, which aims at finding the most concise representation that is able to achieve an acceptable distortion—one kind of popular approach for signal compression relies on finding a basis that provides a sparse, and thus compressible, representation of the signal.

Sparse representation means that a signal of length N can be represented with only $S \ll N$ nonzero coefficients. By storing only the values and locations of the nonzero coefficients, we get a compressed representation of the signal. Sparse approximation has paved the way towards many standard transform-coding schemes that exploit sparsity for compression, including JPEG, JPEG2000, MPEG, and MP3 standards. Compressed sensing takes this concept a step further; it reduces complexity and the computational cost of the acquisition stage. Rather than first sampling in high rate and then compressing the sampled data, we would like to directly sense the data in a compressed form. In a series of pioneering works by Candes (Candès, 2006; Candès and Wakin, 2008), Donoho (2006), and their coauthors, it was shown that when a signal has a sparse representation in a known basis, one can vastly reduce the number of samples that are required-below the Nyquist rate and still be able to *perfectly* recover the signal (under appropriate conditions). This framework suggests to compress the data while sensing it, hence the name compressed sensing.

Compressed sensing differs from the classical sampling theory in three aspects. First, classical sampling theory deals with the question of sampling infinite length, continuous-time signals. Compressed sensing, in contrast, is a mathematical theory that disregards the physical-continuous time aspects of the signal, focusing instead on measuring or projecting finite dimensional vectors in \mathbf{R}^N to lower dimensional ones in \mathbf{R}^M . Second, instead of sampling the signal at specific points in time, the compressed sensing framework measures the signal by linearly projecting it to a known basis. Third, the recovery stage in the traditional Nyquist–Shannon framework is performed through Sinc interpolation, which is a linear process with low complexity. In compressed sensing, however, the signal recovery is more involved, typically achieved using convex-optimization-based recovery methods.

Compressed sensing has made noteworthy contributions to several fields. A prominent example is medical imaging. Scanning sessions of MRI images can be significantly accelerated by measuring fewer Fourier coefficients and reconstructing the under-sampled MRI image while preserving its diagnostic quality (Lustig et al., 2007, 2008). Other applications include building an efficient systems for sub-Nyquist sampling and filtering (Mishali and Eldar, 2010; Tropp et al., 2006), compression of networked data (Haupt et al., 2008), and compressive imaging architectures (Davenport et al., 2010; Duarte et al., 2008; Romberg, 2008).

8 SECTION | ONE Handbook of Numerical Analysis

For the completeness of the presentation of this chapter, we briefly review the mathematical formulation of compressed sensing. Given a signal $\mathbf{x} \in \mathbf{R}^N$, an $M \times N$ sensing matrix Φ (such that $M \ll N$) and a measurements vector $\mathbf{y} = \Phi \mathbf{x}$, the goal of CS is to recover the signal from its measurements \mathbf{y} . The sensing rate is defined by R = M/N, since $R \ll 1$ the recovery of \mathbf{x} is not possible in the general case. CS theory (Candès and Wakin, 2008; Donoho, 2006) suggests that for signals that admit a sparse representation with respect to a dictionary can be exactly recovered with high probability from their measurements: Let $\mathbf{x} = \Psi \mathbf{c}$, where Ψ is the aforementioned dictionary, and \mathbf{c} is a sparse coefficients vector with only $S \ll N$ nonzeros entries. Then, the recovered signal is synthesized by $\hat{\mathbf{x}} = \Psi \hat{\mathbf{c}}$, where $\hat{\mathbf{c}}$ is obtained by solving the following nonconvex optimization problem:

$$\hat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}\|_{0} \quad \text{subject to} \quad \mathbf{y} = \Phi \Psi \mathbf{c},$$
 (1)

where $\|\alpha\|_0$ is the ℓ_0 -pseudo-norm that counts the number of nonzero entries of α . The problem posed in Eq. (1) can also be approximated by its more tractable convex relaxation version,

$$\hat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \mathbf{y} = \Phi \Psi \mathbf{c}.$$
 (2)

The exact recovery of **x** is guaranteed with high probability if **c** is sufficiently sparse and if certain conditions are met by the sensing matrix and the transform (clearly $M \ge S$ is part of these conditions) (Donoho, 2006).

2.2 Compressed Learning

CL was introduced in Calderbank and Jafarpour (2012), showing theoretically that a direct inference from compressive measurements is feasible with high classification accuracy. In particular, this work provided analytical bounds for training a linear support vector machine (SVM) classifier in the compressed sensing domain $\mathbf{y} = \Phi \mathbf{x}$: it proved that under certain conditions on the sensing matrix, called the Distance-Preserving Property, the performance of a linear SVM classifier operating in the compressed sensing domain is almost equivalent to the performance of the best linear threshold classifier operating in the signal \mathbf{x} directly. These results were also shown to be robust to noise in the measurements. The work also showed that a large family of standard compressed sensing matrices satisfies the required Distance-Preserving Property. Moreover, the work demonstrated an application of compressed learning in texture classification, where the goal is to classify a texture-image into one of three classes: "horizontal," "vertical," or "other." The classification was performed based on the horizontal and vertical wavelet coefficients of the image, hence exploiting the underlining sparse representation of texture images. The work demonstrated that an SVM classifier trained directly over the compressed images has high accuracy, close to the one obtained by an SVM that was trained in the data domain.

A different yet very closely related approach termed *smashed filters* was presented in Davenport et al. (2007). This work has shown that accurate classification can be done in the compressed domain, under the assumption that the number of measurements matches the dimensionality of the data manifold. The follow-up paper (Baraniuk and Wakin, 2009) further strengthened these results, showing that if a sufficient number M of random projections are provided, the essential structure of the manifold is preserved. Moreover, the projection dimension M that ensures satisfactory classification performance depends only on this manifold's intrinsic dimension K. This approach was extended and termed *smashed correlation filters* for activity recognition by Kulkarni and Turaga (2016) and for face recognition by Lohit et al. (2015).

Another work of relevance is the one reported in Lohit et al. (2016). A deep learning approach was introduced by the authors, in which random sensing matrices were employed for image classification in the compressed domain. Their work utilized convolutional neural networks (CNNs) that operated on the image domain, and used the following projected measurement vector as the input to the network:^a

$$\mathbf{z} = \boldsymbol{\Phi}^T \mathbf{y} \in \mathbf{R}^N. \tag{3}$$

By training a network similar to LeNet (LeCun et al., 1998) for classifying MNIST handwritten digits images, and using the projected measurement z, good classification results were obtained in this work, significantly outperforming the smashed filters approach at sensing rates as low as R = 0.01. This approach was also successfully verified for the challenging task of classifying a subset of the ImageNet dataset, consisting of 1.2 million images and 1000 categories, also demonstrating excellent classification performance.

3 THE PROPOSED END-TO-END CL APPROACH

This section presents an end-to-end deep learning solution for CL, which jointly optimizes the sensing matrix Φ and the inference operator, parameterized by a coefficients matrix W. The proposed method provides a solution to the following joint optimization problem:

$$\{\widetilde{\Phi}, \widetilde{W}\} = \arg \min_{\Phi, W} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{N}_{W}(\Phi \mathbf{x}_{i}), \mathbf{d}_{i}),$$
(4)

where $\{\mathbf{x}_i, \mathbf{d}_i\}_{i=1}^N$ is the collection of N pairs of signals \mathbf{x}_i and their corresponding labels \mathbf{d}_i . The loss function $\mathcal{L}(\bullet, \bullet)$ measures the distance between the true label and the estimated one, provided by the inference operator $\mathcal{N}_W(\bullet)$, whose input is the compressed sample, denoted by $\Phi \mathbf{x}_i$. Here we have employed the negative-log-likelihood loss function, which is commonly used for learning classification networks. Note that during training the sensing

^aThis was fed as a replacement to the true image, after reshaping it to $\sqrt{N} \times \sqrt{N}$ pixels.

layer (matrix) Φ and the subsequent layers represented by $\mathcal{N}_W(\bullet)$ are treated as a single deep network. Thus, the goal of the learning phase is to propose Φ and W that would perform best classification. However, once training is complete, the sensing matrix is detached from the subsequent inference layers, and used for performing signal sensing. The input of the inference operator is therefore the second layer of the end-to-end learned network.

This approach is motivated by the success of CNNs for the task of compressive image classification (Lohit et al., 2016), which employed a random sensing matrix (with Gaussian entries) for classifying the MNIST (LeCun et al., 1998) dataset, and a Hadamard matrix for classifying a subset of the ImageNet dataset. In our approach, the first layer learns the sensing matrix $\tilde{\Phi}$, and subsequent layers (a fully connected layer followed by LeNet (LeCun et al., 1998) or ResNet (Zagoruyko and Komodakis, 2016) layers as described in the next section) perform the nonlinear inference stage. Fig. 1 illustrates the aforementioned flow. Note that the second fully connected layer performs a similar operator to the one posed in Eq. (3), however, a different matrix $\tilde{\Psi} \in \mathbf{R}^{N \times M}$ is learned.

The proposed method was tested on two well-known datasets: MNIST hand written digit recognition database and CIFAR10 image recognition database, and the behaviour of this scheme is detailed in the next section.

4 PERFORMANCE EVALUATION

This section describes the proposed architectures and provides performance evaluation of the results.

4.1 MNIST Dataset

The MNIST dataset (LeCun et al., 1998) contains 70,000 grayscale $28 \times 28 =$ 784 pixel images of handwritten digits, each belongs to one of 10 classes, i.e., denoting one digit in the range [0, 1, 2, ..., 9]. The dataset is split into 60,000



FIG. 1 A scheme of the proposed approach: end-to-end solution for CL that jointly optimizes the sensing and the inference operators. The specific CNN architecture depends on the dataset, as described in Section 4.

training images and 10,000 test images. The proposed network architecture for MNIST includes the following layers and elements:

- **1.** An input layer with *N* nodes.
- 2. A compressed sensing fully connected layer with NR nodes, $R \ll 1$ (its weights denote the sensing matrix).
- 3. A fully connected reprojection layer that expands the output of the sensing layer to the original image dimensions N.
- 4. Tanh activation units to control the values entering the network.
- 5. A fully connected layer with N nodes.
- 6. Reshape operator to two-dimensional $\sqrt{N} \times \sqrt{N}$ tensor (size of the original image).
- 7. A convolution layer with kernel sizes of 5 \times 5, which generates six feature map.
- 8. ReLU activation units.
- 9. Max pooling layer which selects the maximum of 2×2 feature maps elements, with a stride of 2 in each dimension.
- 10. A convolution layer with kernel sizes of 5 \times 5, which generates 16 feature map.
- **11.** ReLU activation units.
- 12. Max pooling layer which selects the maximum of 2×2 feature maps elements, with a stride of 2 in each dimension.
- 13. Reshape operator that reshapes the 16 4 \times 4 max-pooled features maps into a single 256-dimensional vector.
- 14. A fully connected layer of 256 to 120 nodes.
- 15. ReLU activation units.
- 16. A fully connected layer of 120 to 84 nodes.
- 17. ReLU activation units.
- **18.** A softmax layer with 10 outputs (corresponding to the 10 MNIST classes).

We have trained^b the proposed network using the training images of the MNIST dataset, using stochastic gradient descent (SGD) with a learning rate of 0.0025, over 100 epochs. The network was initialized with random weights. The classification error performance was evaluated using the test set for sensing rates in the range of R = 0.25 to R = 0.01, and averaged over the collection of 10,000 MNIST test images.

Table 1 summarizes the classification error results of MNIST database compared to smashed filters (Davenport et al., 2007), and random sensing matrix followed by convolutional network (Lohit et al., 2016). In addition, we compared our results with a compression using principal component analysis (PCA) (Jolliffe, 2002), followed by convolutional network for classification. In this case, the sensing matrix Φ is obtained by taking the *NR* first

^bThe network was implemented using Torch7 (Collobert et al., 2011) scripting language and trained on NVIDIA Titan X GPU card.

TABLE 1 Classification Error (%) for the MNIST Handwritten Digits Dataset vs Sensing Rate R = M/N (Averaged Over 10,000 Test Images)

Sensing Rate	No. of Measurements	Smashed Filters (Davenport et al., 2007)	Random Sensing + CNN (Lohit et al., 2016)	PCA + CNN	End-to-End Network
0.25	196	27.42%	1.63%	1.38 %	1.48%
0.1	78	43.55%	2.99%	1.6%	1.51 %
0.05	39	53.21%	5.18%	1.87%	1.67 %
0.01	8	63.03%	41.06%	6.9%	5.1 %

The lowest classification error rates are marked with bold.

eigenvectors corresponding to the largest eigenvalues of the covariance matrix (estimated using the training set). This is followed by a reprojection of the data $z = \Phi^T \mathbf{y}$ using the projection matrix transpose Φ^T , and the resulting z is then fed into the network as an image. Table 1 reveals the advantage of the proposed approach, which increases significantly for lower sensing rates.

A strange behaviour is observed for high sensing rate, where the PCA projection slightly outperforms our trained approach. This can be explained by the fact that for medium compression levels as in this case, PCA essentially captures all the visual information in the given digit images, thus losing nothing. Indeed, after recovery by Ψ , the images fed to the CNN are of nearly the same quality as the original ones. The question that remains is why the learning method did not converge to a PCA projection? We believe that several hundreds of additional epochs in the training would have made the necessary difference.

Fig. 2 shows the sensing matrix Φ obtained through our end-to-end network, compared to the sensing of the PCA projection, both obtained for a sensing rate of R = 0.25. Each image tile is the result of reshaping a row from Φ to the size of an original MNIST image 28×28 . Note that every value in the resulting compressed image is obtained by an inner product between each of these images and the input image. Thus, a high value after projection would indicate a high correlation between the original image and the corresponding row in the sensing matrix. As shown in Fig. 2B, the uppermost rows of the PCA projection matrix have structures akin to digits or a composition of them, as expected from a PCA projection, which optimizes for signal reconstruction. Interestingly, Fig. 2A shows that the rows of the resulting learned sensing



FIG. 2 A comparison between the sensing matrix obtained by the PCA projection and the matrix learned using our end-to-end network for R = 0.25. Note that each image tile shows a row in its respective sensing matrix, reshaped to a 28 \times 28 image. (A) Our compressed-learning results; (B) PCA.



FIG. 3 A comparison between the sensing matrix obtained by the PCA projection and the matrix learned using our end-to-end network for R = 0.01. (A) Our compressed-learning results; (B) PCA.

matrix do not exhibit any recognizable structures, neither are they representative of independent random sensing. In Fig. 3 we repeat this comparison for R = 0.01. Again, the sensing matrix of the network presents some structure that differentiates it from a complete random matrix, but it is also dissimilar to the structure of the PCA projection.

4.2 CIFAR10 Dataset

The CIFAR10 dataset (Krizhevsky et al., 2014) contains 60,000 colour images of $32 \times 32 = 1024$ pixels, drawn from 10 different classes. This dataset is divided into training and test sets, containing 50,000 and 10,000 images respectively. For training on CIFAR10 we used an architecture based on ResNet—wide residual networks (WRN) (Zagoruyko and Komodakis, 2016) composed of the following:

- **1.** A three-channel (red, green, and blue) parallel network with fully connected layers for compression, and fully connected reprojection layers (which expand the output size to the original dimension) for each channel.
- 2. Reshape operator to three-dimensional tensor (size of the original image).
- **3.** WRN layers (Zagoruyko and Komodakis, 2016) perform the classification stage.

The proposed network was trained on the training images of the CIFAR10 dataset, with initialization as follows:

- **1.** The first two layers were initialized with weights obtained by minimizing the mean squared reconstruction error (MSE) of the compressed signal.
- **2.** The following ResNet layers were initialized with weights learned from CIFAR10 without any compression.

The optimization algorithm of choice was SGD with an initial learning rate of 0.001 and learning rate decay of 0.2 every 50 epochs, with momentum of 0.9, over 200 epochs. For this dataset we also used data augmentation to expand the training set as in Zagoruyko and Komodakis (2016): horizontal flips and random crops taken from image padded by four pixels on each side using reflection of the image original boundary pixels. Table 2 shows the

Dutubuse	Dutubase vs sensing hate K = M/M (Wendged Over 10,000 Test images)									
Sensing Rate	No. of Measurements	Random Sensing + WRN (Lohit et al., 2016)	PCA + WRN	End-to- End Network	End-to- End Network					
				Shared Weights	Nonshared Weights					
1 (Oracle)	1024	4.65%	4.65%	4.65%	4.65%					
0.25	256	30.25%	7.4%	7.68%	9.24%					
0.1	102	40.61%	12.71 %	12.73%	15.29%					
0.025	26	55.63%	29.44%	26.68%	28.14%					
0.01	10	68.03%	42.6%	40.65%	40.75%					

TABLE 2 Classification Error (%) for the CIFAR10 Image Recognition Database vs Sensing Rate R = M/N (Averaged Over 10,000 Test Images)

The lowest classification error rates are marked with bold.

classification results obtained by averaging over 10,000 CIFAR10 test images vs random sensing followed by WRN layers (Zagoruyko and Komodakis, 2016), and PCA compression followed by the same layers. Since this dataset is composed of colour images, we perform PCA projection and reprojection on each colour separately, and reshape the output to a size of an image before feeding it to the network. We evaluated the classification performance on sensing rates in the range of R = 0.25 to R =0.01; on each of these we considered two setting options: (1) imposing the projection and reprojection matrices of the colours to be the same (shared weights) or (2) learning the projection and reprojection matrices of each colour independently (nonshared weights).

Table 2 demonstrates the advantage of the end-to-end approach at low sensing rate over alternative methods. Note that here as well we see that PCA seems to perform rather well in sufficiently high sampling rates, and the explanation to this is the same as in the MNIST experiment. The results in Table 2 suggest that the shared weights setting leads to better classification accuracy. Shared matrices reduce the number of parameters, improve generalization, and lessen over-fit. Note that although nonshared weights allow the network to learn a different matrix for each channel (and thus enable more flexible projection structures), the increased number of parameters degrades performance.

5 CONCLUSIONS

In this chapter we reviewed the concept of compressed learning as a remedy to the complexity and storage obstacles when working on high-dimensional data. We indicated its advantages from compression and learning points of view. We have shown that the simplest linear dimensionality reduction processes (random sensing, PCA, etc.) are sufficient for crude classification tasks. Moreover, we experimentally showed that optimizing the sensing matrix jointly with a nonlinear inference operator using neural networks, improved upon the other methods which used a simpler, standard linear projection matrices. In our examples, the signals were returned to the full dimension prior entering the CNN section, since we used redesigned networks and added compression-decompression layers. In future work, other options that do not return to the full dimension should be investigated. Other future research directions include analyzing the properties of the learned sensing matrices from an RIP perspective, and applying further constraints during learning to the sensing matrices, such as limiting them to binary coefficients. Beyond classification, the proposed approach can be extended to other CL applications, such as detection and recognition of patterns in single and multichannel images or signals.

REFERENCES

- Adler, A., Elad, M., Zibulevsky, M., 2016. Compressed learning: a deep neural network approach. arXiv preprint arXiv:1610.09615.
- Adler, A., Boublil, D., Elad, M., Zibulevsky, M., 2017. A deep learning approach to block-based compressed sensing of images. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- Baraniuk, R.G., Wakin, M.B., 2009. Random projections of smooth manifolds. Found. Comput. Math. 9 (1), 51–77.
- Calderbank, R., Jafarpour, S., 2012. Finding needles in compressed haystacks. In: Eldar, Y.C., Kutyniok, G. (Eds.), Compressed Sensing: Theory and Applications, Cambridge University Press, pp. 439–484.
- Candès, E.J., 2006. Compressive sampling. Proceedings of the International Congress of Mathematicians, vol. 3, pp. 1433–1452. Madrid, Spain.
- Candès, E.J., Wakin, M.B., 2008. An introduction to compressive sampling. IEEE Signal Process. Mag. 25 (2), 21–30.
- Collobert, R., Kavukcuoglu, K., Farabet, C., 2011. Torch7: a MATLAB-like environment for machine learning. In: BigLearn, NIPS Workshop, EPFL-CONF-192376.
- Davenport, M.A., Duarte, M.F., Wakin, M.B., Laska, J.N., Takhar, D., Kelly, K.F., Baraniuk, R.G., 2007. The smashed filter for compressive classification and target recognition. In: Computational Imaging V, vol. 6498. International Society for Optics and Photonics, p. 64980H.
- Davenport, M.A., Hegde, C., Duarte, M.F., Baraniuk, R.G., 2010. Joint manifolds for data fusion. IEEE Trans. Image Process. 19 (10), 2580–2594.
- Donoho, D.L., 2006. Compressed sensing. IEEE Trans. Inf. Theory 52 (4), 1289-1306.
- Duarte, M.F., Davenport, M.A., Takhar, D., Laska, J.N., Sun, T., Kelly, K.F., Baraniuk, R.G., 2008. Single-pixel imaging via compressive sampling. IEEE Signal Process. Mag. 25 (2), 83–91.

- Elad, M., 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. vol. 1MIT Press, Cambridge.
- Hahn, J., Rosenkranz, S., Zoubir, A.M., 2014. Adaptive compressed classification for hyperspectral imagery. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 1020–1024.
- Haupt, J., Bajwa, W.U., Rabbat, M., Nowak, R., 2008. Compressed sensing for networked data. IEEE Signal Process. Mag. 25 (2), 92–101.
- Jolliffe, I.T., 2002. Principal component analysis and factor analysis. In: Principal Component Analysis, second ed. Springer, New York, NY, pp. 150–166 (Chapter 7).
- Krizhevsky, A., Nair, V., Hinton, G., 2014. The CIFAR-10 dataset. https://www.cs.toronto. edu/~kriz/cifar.html.
- Kulkarni, K., Turaga, P., 2012. Recurrence textures for human activity recognition from compressive cameras. 19th International Conference on Image Processing (ICIP), IEEE, pp. 1417–1420.
- Kulkarni, K., Turaga, P., 2016. Reconstruction-free action inference from compressive imagers. IEEE Trans. Pattern Anal. Mach. Intell. 38 (4), 772–784.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.
- Li, Y., Hegde, C., Sankaranarayanan, A.C., Baraniuk, R., Kelly, K.F., 2015. Compressive image acquisition and classification via secant projections. J. Opt. 17 (6), 065701.
- Lohit, S., Kulkarni, K., Turaga, P., Wang, J., Sankaranarayanan, A.C., 2015. Reconstruction-free inference on compressive measurements. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 16–24.
- Lohit, S., Kulkarni, K., Turaga, P., 2016. Direct inference on compressive measurements using convolutional neural networks. In: IEEE Image Processing (ICIP), International Conference on Image Processing, IEEE, pp. 1913–1917.
- Lustig, M., Donoho, D., Pauly, J.M., 2007. Sparse MRI: the application of compressed sensing for rapid MR imaging. Magn. Reson. Med. 58 (6), 1182–1195.
- Lustig, M., Donoho, D.L., Santos, J.M., Pauly, J.M., 2008. Compressed sensing MRI. IEEE Signal Process. Mag. 25 (2), 72–82.
- Maillard, O., Munos, R., 2009. Compressed least-squares regression. Advances in Neural Information Processing Systems, pp. 1213–1221.
- Mishali, M., Eldar, Y.C., 2010. From theory to practice: sub-Nyquist sampling of sparse wideband analog signals. IEEE J. Sel. Top. Sign. Proces. 4 (2), 375–391.
- Romberg, J., 2008. Imaging via compressive sampling. IEEE Signal Process. Mag. 25 (2), 14-20.
- Rubinstein, R., Bruckstein, A.M., Elad, M., 2010. Dictionaries for sparse representation modeling. Proc. IEEE 98 (6), 1045–1057.
- Sankaranarayanan, A.C., Turaga, P.K., Baraniuk, R.G., Chellappa, R., 2010. Compressive acquisition of dynamic scenes. European Conference on Computer Vision, pp. 129–142.
- Tropp, J.A., Wakin, M.B., Duarte, M.F., Baron, D., Baraniuk, R.G., 2006. Random filters for compressive sampling and reconstruction. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings, vol. 3. IEEE, p. III.
- Wang, Q., Zeng, W., Tian, J., 2014. A compressive sensing based secure watermark detection and privacy preserving storage framework. IEEE Trans. Image Process. 23 (3), 1317–1328.
- Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. arXiv preprint arXiv:1605.07146.
- Zhang, Y.-N., Pan, X.-Y., Huang, Y., Shen, H.-B., 2011. Adaptive compressive learning for prediction of protein-protein interactions from primary sequence. J. Theor. Biol. 283 (1), 44–52.