

# Towards an understanding of CNNs: analysing the recovery of activation pathways via Deep Convolutional Sparse Coding

Michael Murray<sup>\*†</sup>, Jared Tanner<sup>\*†</sup>

June 26, 2018

## Abstract

Deep Convolutional Sparse Coding (D-CSC) is a framework reminiscent of deep convolutional neural nets (DCNN), but by omitting the learning of the dictionaries one can more transparently analyse the role of the activation function and its ability to recover activation paths through the layers. Pappan, Romano, and Elad conducted an analysis of such an architecture [1], demonstrated the relationship with DCNNs and proved conditions under which a D-CSC is guaranteed to recover activation paths. A technical innovation of their work highlights that one can view the efficacy of the ReLU nonlinear activation function of a DCNN through a new variant of the tensor’s sparsity, referred to as stripe-sparsity. Using this they proved that representations with an activation density proportional to the ambient dimension of the data are recoverable. We extend their uniform guarantees to a modified model and prove that with high probability the true activation is typically possible to recover for a greater density of activations per layer. Our extension follows from incorporating the prior work on one step thresholding by Schnass and Vandergheynst [2] into the appropriately modified architecture of [1].

**Keywords:** Deep Learning, Deep Convolutional Neural Networks, Sparse Coding, Deep Convolutional Sparse Coding, Rademacher Concentration

## 1 Introduction

Ever since the arrival of AlexNet (Krizhevsky et al, [3]) in 2012, Deep Convolutional Neural Networks (DCNNs) have been the state of the art technique for tackling many tasks in computer vision. The success of DCNNs is not only limited to computer vision, achieving excellent results in a host of other application areas such as Natural Language Processing (NLP) (Kim et al [4]) and Speech Recognition (Zhang et al [5]). However, although their empirical success and practical usefulness is undeniable, exactly why they are so successful remains somewhat of a mystery. A large number of open questions remain; for instance (and to consider just a few), how are DCNNs able to avoid the curse of dimensionality during training? Why is training via gradient based methods so effective? How do DCNNs achieve their invariance properties? How should we design a neural network for a given task and why do they generalise so well? In recent years, a large number of researchers have been working to better understand DCNNs and we now highlight just a few contributions in the

---

<sup>\*</sup>Mathematical Institute, University of Oxford

<sup>†</sup>The Alan Turing Institute

<sup>‡</sup>Contact emails for paper: mmurray@turing.ac.uk, tanner@maths.ox.ac.uk

<sup>§</sup>This publication is based on work supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1 and the Ana Leaf Foundation. The authors thank David L. Donoho, Vardan Pappan for motivating this work, Hemant Tyagi for his helpful feedback and Jeremias Sulam for stimulating discussions.

context of the questions above. With regard to avoiding the curse of dimensionality, Mhaskar and Poggio analysed the approximation capabilities of shallow vs. deep networks for compositional functions, proving that DCNNs, for the same level of approximation accuracy, require an exponentially lower number of training points compared with shallow networks [6]. The works of Pennington et al [7], Poggio [8], Freeman and Bruna [9] indicate that although the loss functions of neural networks in general are non-convex, for sufficiently large networks most local minima are in fact close to being global minima, furthermore many of these global minima are flat and degenerate. Additionally Haeefe and Vidal in [10] were able to prove, for a certain form of parallelised network architecture, that if the size of the network is sufficiently large then local descent can reach a global minimizer from any initialisation. Bruna and Mallat [11] analysed the invariance properties of DCNNs, showing that invariance can also be achieved by a wavelet based approach they termed the Scattering Transform. Wiatowski et al [12] built on this work to specify the number of layers this approach requires to capture most of an input signal’s energy, providing some insight and potential guidelines for more general network architecture design. Finally in regard to understanding how DCNNs generalise, Zhang et al [13] experimentally demonstrated that DCNNs can fit random data with ease, suggesting gaps in current theories concerning regularisation. In this paper we build on another work by Pappayan et al [1]. In this paper they seek to answer questions concerning the role of the forward pass algorithm and of the activation pathways (the pattern of nodes at each layer that are activated by the forward pass). Inspired by the observation that the ReLU activation function is a sparsifying operator, their paper connects deep learning with the field of compressed sensing, interpreting the forward pass algorithm as approximately solving a sequence of sparse coding problems. In doing so they are able to leverage the tools of compressed sensing to analyse under which circumstances an activation path is recovered.

ReLU’s are a highly popular choice of activation function, playing a role in most DCNN implementations. Indeed, it has been posited that the enforcement of sparsity through ReLU functions may be one of the key ingredients behind certain DCNNs’ effectiveness at developing composite representations at depth [14]. By omitting the learning of the dictionaries in DCNNs, one can more transparently analyse the role of the forward pass algorithm and its ability to recover activation paths through the layers. Pappayan et al [1] propose a deep convolutional sparse coding (D-CSC) model with this aim and use techniques from sparse coding to prove conditions under which a DCNN is guaranteed to activate the desired nodes in a given layer. This analysis however is based upon pessimistic worst case assumptions, which limit the applicability of these results. In this paper we aim to address this issue, extending from uniform guarantees to probabilistic bounds. The outline of this paper is as follows. In Section 2 we will review the D-CSC model and the conditions for uniform recovery of activation pathways, presented in [1]. In Section 3 we present probabilistic bounds for the recovery of activation pathways and provide a proof of this result. Finally, in Section 4 we conclude by providing a summary and some practical implications of our results.

## 2 Background

### 2.1 A quick overview of notation

Before proceeding we define the following measures which we will use extensively.

- $\|\mathbf{x}\|_{\alpha,\infty}^{P_i^{(l)}} \triangleq \max_i \|P_i^{(l)} \mathbf{x}\|_\alpha$  where  $P_i^{(l)}$ , the patch operator, takes  $m_l$  consecutive elements of  $\mathbf{x}$  (with wraparound) starting at  $x_i$  (see [1] for further details). In this paper we will only consider  $\alpha \in \{0, 2\}$ , hence  $\|\cdot\|_\alpha$  refers to the standard  $l_2$  norm when  $\alpha = 2$ , and a function counting the number of non-zeros in the argument vector when  $\alpha = 0$ .

- $\|\mathbf{x}\|_{\alpha,\infty}^{Q^{(l)}} \triangleq \max_i \|Q_i^{(l)} \mathbf{x}\|_\alpha$  where  $Q_i^{(l)}$ , the stripe operator, takes  $\lfloor ((2(m_l/s_l) - 1)n_l) \rfloor$  consecutive elements of  $\mathbf{x}$  (with wraparound) starting at  $x_i$  (see [1] for further details). As before we will only consider  $\alpha \in \{0, 2\}$ , hence  $\|\cdot\|_\alpha$  refers to the standard  $l_2$  norm when  $\alpha = 2$ , and a function counting the number of non-zeros in the argument vector when  $\alpha = 0$ .
- $\|\mathbf{X}^{(l)}\|_{\alpha,\infty}^{Q^{(l)}} \triangleq \max_j \|\mathbf{x}_j\|_{\alpha,\infty}^{Q^{(l)}}$ , which is simply the maximum of the set of measures (defined above) on the columns of  $\mathbf{X}^{(l)}$ .

## 2.2 The D-CSC model - interpreting the forward pass algorithm of a CNN as a sequence of sparse coding problems

To interpret the forward pass as solving a sequence of sparse coding problems we firstly model the data in the classic compressed sensing setup. Denoting the matrix containing our data as  $\mathbf{X}^{(0)}$ , suppose that it is formed from a linear combination of columns of a convolutional matrix  $\mathbf{A}$  (a convolutional matrix is a banded circular matrix created by shifting a locally supported matrix across all spatial locations, see [15] and [1] for more details). This can be expressed as  $\mathbf{X}^{(0)} = \mathbf{A}\mathbf{X}^{(L)}$  where  $\mathbf{X}^{(L)}$  denotes the matrix containing the sparse codes of our data (the reasons for this particular choice of notation shall become clear). For the particulars and details concerning convolutional sparse coding we refer the reader to [15]. Suppose now that  $\mathbf{A}$  can in fact be factorized as a product of convolutional matrices,  $\mathbf{A} = \Pi_{l=1}^L \mathbf{A}^{(l)}$ . To make the connection to DCNNs consider examining the representation of  $\mathbf{X}^{(0)}$  at a different level of the factorisation other than  $L$ . To explain through an example denote  $\mathbf{X}^{(l)} = \Pi_{j=l+1}^L \mathbf{A}^{(j)} \mathbf{X}^{(L)}$ , then  $\mathbf{X}^{(0)} = \Pi_{j=l}^L \mathbf{A}^{(j)} \mathbf{X}^{(l)}$ . We define  $\mathbf{X}^{(l)}$  as the latent representation of  $\mathbf{X}^{(0)}$  at *layer*  $l$ . If we consider starting at the deepest representation and work backwards by applying the matrix factors of  $\mathbf{A}$  sequentially and in order, then expressing each representation in terms of the next we formulate the model

$$\begin{aligned}
\mathbf{X}^{(L-1)} &= \mathbf{A}^{(L)} \mathbf{D}^{(L)} \mathbf{X}^{(L)} \\
&\dots \\
\mathbf{X}^{(1)} &= \mathbf{A}^{(2)} \mathbf{D}^{(2)} \mathbf{X}^{(2)} \\
\mathbf{X}^{(0)} &= \mathbf{A}^{(1)} \mathbf{D}^{(1)} \mathbf{X}^{(1)}.
\end{aligned} \tag{2.1}$$

If we additionally impose certain sparsity constraints on the representations at each layer then this model reduces to the  $DCP_\lambda$  model proposed in [1] when  $\mathbf{D}^{(l)} = \mathbf{I} \forall l$ . The reason for the inclusion  $\mathbf{D}^{(l)}$  at each layer shall become clear in Section 3. We define the *reverse pass* of the D-CSC model as the generation of each  $\mathbf{X}^{(l)}$  for  $l = L-1, L-2 \dots 0$  starting at the deepest layer,  $\mathbf{X}^{(L)}$ . That is, given  $\{\mathbf{A}^{(l)}\}_{l=1}^L$ ,  $\{\mathbf{D}^{(l)}\}_{l=1}^L$  and  $\mathbf{X}^{(L)}$ , we sequentially generate  $\mathbf{X}^{(l)}$  as in (2.1). For clarity we now describe each variable in more detail.

- $\mathbf{X}^{(l)} \in \mathbb{R}^{n_l M \times d}$  is a matrix containing the latent representation of the data at layer  $l$ , with each of the  $d$  columns a data point of dimension  $M$ . To be clear  $\mathbf{X}^{(l)} = [\mathbf{x}_1^{(l)} \mathbf{x}_2^{(l)} \dots \mathbf{x}_d^{(l)}]$  where throughout lower case bold letters will denote the columns of their respective capitalised bold letter denoted matrices. We furthermore assume that there exists scalars  $\{S_l\}_{l=1}^L$  such that  $\|\mathbf{X}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$  for all  $l$ .
- $\mathbf{D}^{(l)} \sim \mathbb{B}^{n_l M \times n_l M}$  is a sample drawn from the distribution of random, diagonal matrices whose diagonal components are independent Rademacher random variables. To be clear, these are square matrices with dimension  $n_l M$ , whose off diagonal elements are all zero and whose diagonal elements are either 1 or  $-1$  with probability 0.5.

- $\mathbf{A}^{(l)} \in \mathbb{R}^{(n_{l-1})M \times n_l M}$  is the transpose of the weight matrix mapping between layers  $l-1$  and  $l$ . This matrix has a convolutional structure (described in [15] and [1]), is circular and banded, and is created by shifting a local dictionary  $\mathbf{A}_{Local}^{(l)} \in \mathbb{R}^{m_l \times n_l}$  across all spatial locations. For  $l \geq 2$  there is a stride between each spatially shifted  $\mathbf{A}^{(l)}$  which we denote  $s_l$ , in [1]  $s_l = n_{l-1}$ . The columns of  $\mathbf{A}^{(l)}$  have unit  $\ell_2$  norm and as before we denote the columns of the matrix as  $\mathbf{A}^{(l)} = [\mathbf{a}_1^{(l)} \ \mathbf{a}_2^{(l)} \ \dots \ \mathbf{a}_d^{(l)}]$ .

Suppose that we know  $\{\mathbf{A}^{(l)}\}_{l=1}^L$ , have access to  $\{\mathbf{D}^{(l)}\}_{l=1}^L$  and are given  $\hat{\mathbf{X}}^{(0)}$ , where

$$\begin{aligned}\hat{\mathbf{X}}^{(0)} &= \mathbf{X}^{(0)} + \mathbf{V}^{(0)} \\ &= \mathbf{A}^{(1)}\mathbf{D}^{(1)}\mathbf{X}^{(1)} + \mathbf{V}^{(0)}.\end{aligned}$$

$\mathbf{V}^{(0)}$  is the *model error* matrix that arises since our data may only approximately belong to this model, i.e., the data may only approximately be represented as a linear combination of elements from the global dictionary  $\mathbf{A} = \Pi_{l=1}^L \mathbf{A}^{(l)}$ . We shall consider the efficacy of the *forward pass* of the D-CSC model to recover the hidden or latent representations  $\{\mathbf{X}^{(l)}\}_{l=1}^L$  from  $\hat{\mathbf{X}}^{(0)}$ . Suppose we compute an estimate  $\hat{\mathbf{X}}^{(1)}$  and define the corresponding *estimation error* with regard to  $\mathbf{X}^{(1)}$  as  $\mathbf{V}^{(1)}$  so that  $\hat{\mathbf{X}}^{(1)} = \mathbf{X}^{(1)} + \mathbf{V}^{(1)}$ . Continuing in this vein we can therefore define the following sequence of noisy sparse coding problems which we solve, sequentially, to estimate the sparse representations  $\mathbf{X}^{(l)}$  for  $l = 1, 2, \dots, L$ .

$$\begin{aligned}\hat{\mathbf{X}}^{(0)} &= \mathbf{A}^{(1)}\mathbf{D}^{(1)}\mathbf{X}^{(1)} + \mathbf{V}^{(0)} \\ \hat{\mathbf{X}}^{(1)} &= \mathbf{A}^{(2)}\mathbf{D}^{(2)}\mathbf{X}^{(2)} + \mathbf{V}^{(1)} \\ &\dots \\ \hat{\mathbf{X}}^{(L-1)} &= \mathbf{A}^{(L)}\mathbf{D}^{(L)}\mathbf{X}^{(L)} + \mathbf{V}^{(L-1)}.\end{aligned}\tag{2.2}$$

Note that the error propagates between each layer since the estimate of the sparse representation in the previous layer is fed into the next layer. Our notation is clarified below.

- $\hat{\mathbf{X}}^{(0)} \in \mathbb{R}^{M \times d}$  is a matrix containing the observed noisy data, with each of the  $d$  columns representing a data point of dimension  $M$ . As before, we denote the columns of the matrix as  $\hat{\mathbf{X}}^{(0)} = [\hat{\mathbf{x}}_1^{(0)} \ \hat{\mathbf{x}}_2^{(0)} \ \dots \ \hat{\mathbf{x}}_d^{(0)}]$ .
- $\hat{\mathbf{X}}^{(l)} \in \mathbb{R}^{n_l M \times d}$  is a matrix containing the noisy representation of the observed data  $\hat{\mathbf{X}}^{(0)}$  at layer  $l$ . As before, we denote the columns of the matrix as  $\hat{\mathbf{X}}^{(l)} = [\hat{\mathbf{x}}_1^{(l)} \ \hat{\mathbf{x}}_2^{(l)} \ \dots \ \hat{\mathbf{x}}_d^{(l)}]$ .
- $\mathbf{V}^{(l)} \in \mathbb{R}^{n_l M \times d}$  is an unknown error matrix  $\mathbf{V}^{(l)} = \hat{\mathbf{X}}^{(l)} - \mathbf{X}^{(l)}$  which we assume has bounded energy satisfying  $\|\mathbf{V}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l$ . As before, we denote the columns as  $\mathbf{V}^{(l)} = [\mathbf{v}_1^{(l)} \ \mathbf{v}_2^{(l)} \ \dots \ \mathbf{v}_d^{(l)}]$ .
- $\mathbf{D}^{(l)}$  are the *same* samples as in 2.1 used to generate the representations at each layer.

Hopefully the connection between the D-CSC framework and DCNNs is now apparent. To summarise, we can view the forward pass of a neural network as solving the sequence of sparse coding problems defined in (2.2) with  $\hat{\mathbf{X}}^{(0)}$  being the data we feed into our network and  $\hat{\mathbf{X}}^{(l)}$  the estimate of the latent representation of the data at layer  $l$ . To dispel any potential confusion, note

that the *forward activation pattern* at layer  $l$  refers to the support, i.e. the location of the nonzero entries, of  $\hat{\mathbf{X}}^{(l)}$ . The *reverse activation pattern* at layer  $l$  refers to the support of  $\mathbf{X}^{(l)}$ . Our aim is to derive conditions for which the reverse and forward activation patterns coincide across all layers. Defining the activation pathway as the sequence of activations across all layers (e.g. the reverse activation pathway is  $\{\text{supp}(\mathbf{X}^{(l)})\}_{l=1}^L$ ), we can summarise our goal as attempting to understand when the forward and reverse activation pathways for a given data point are the same, or at least are highly likely to be the same. Assuming that we are able to recover the reverse activation pattern, we will also aim to bound the error between the estimated and true representation at a given layer.

### 2.3 Uniform guarantees for the recovery of activation pathways

In [1] Pappayan, Romano, and Elad consider the  $\{\hat{\mathbf{X}}^{(l)}\}_{l=1}^L$  obtained by applying a feed forward algorithm to solve the sequence of sparse coding problems defined in 2.2 and prove that this model admits a solution under certain sparsity constraints. Their analysis relies heavily on the notion of the coherence of a dictionary,

$$\mu(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i^* \mathbf{a}_j|, \quad (2.3)$$

where  $\mathbf{a}_i$  is the  $i^{\text{th}}$  column of  $\mathbf{A}$ . The main technical innovations in [1] include how bounds from traditional sparse approximation propagate through multiple layers, and also the aforementioned sparsity measures used to ameliorate the limited lower bound on (2.3). This is often observed for matrices with a convolutional structure as in the case of  $\mathbf{A}^{(l)}$ . In particular, the main results in [1] most relevant to our extensions lets  $\{\mathbf{X}^{(l)}\}_{l=1}^L$  be a set of sparse matrices consistent with Model (2.1). The activations in the presence of noise, as in (2.2) are estimated recursively from the data matrix  $\hat{\mathbf{X}}^{(0)}$  according to

$$\hat{\mathbf{X}}^{(l)} = \text{Proj}_{|\text{supp}(\cdot)|=k} \left( (\mathbf{A}^{(l)} \mathbf{D}^{(l)})^T \hat{\mathbf{X}}^{(l-1)} \right). \quad (2.4)$$

for  $l = 1, 2, \dots, L$  and in Pappayan et al's analysis  $\mathbf{D}^{(l)} = \mathbf{I}$  for all  $l$  (note that in all our results that follow we shall assume that  $\mathbf{D}^{(l)}$  are samples drawn from distributions of Rademacher random matrices, as described in model 2.2).  $\text{Proj}_{|\text{supp}(\cdot)|=k}(\cdot)$  denotes a family of sparsifying operators that sets all but  $k$  entries of the each column of the matrix argument to 0. By substituting  $\text{Proj}_{|\text{supp}(\cdot)|=k}(\cdot)$  with a ReLU operator we obtain the standard forward pass algorithm across a ReLU layer of a neural network with  $(\mathbf{A}^{(l)} \mathbf{D}^{(l)})^T$  the weight matrix between layer  $l$  and  $l-1$ . In what follows we will assume that  $\text{Proj}_{|\text{supp}(\cdot)|=k}(\cdot)$  refers to basic thresholding with the hard thresholding operator [16]. To be clear, this operator keeps the  $k$  largest entries (in terms of absolute value) of each column of the argument matrix unchanged while setting all other elements to zero.

Pappayan et al prove <sup>1</sup>, under worst case assumptions, that if the noise free data satisfies a sparsity bound  $\|\mathbf{X}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$  and  $\|\mathbf{V}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l$  for some sets of scalars  $\{S_l\}_{l=1}^L$  and  $\{\zeta_l\}_{l=1}^L$ , then so long as

$$S_l < \frac{\mu^{(l)-1}}{|X_{\max}^{(l)}|} \left( \frac{1}{2} |X_{\min}^{(l)}| - \zeta_l \right) + \frac{1}{2} \quad (2.5)$$

the activation (or support) of  $\hat{\mathbf{X}}^{(l)}$  computed using 2.4 is exactly the same as the activation of  $\mathbf{X}^{(l)}$ . In other words, the reverse activation pattern at layer  $l$  is recovered. Note that we require

<sup>1</sup>Pappayan et al's analysis is wide-ranging, including conditions under which the solution to Model (2.2) is unique. They also consider a variety of thresholding operators such as soft and hard thresholding as well as more advanced algorithms to compute  $\hat{\mathbf{X}}^{(l)}$  from  $\mathbf{A}^{(l)}$  and  $\hat{\mathbf{X}}^{(l-1)}$ . We speak here only of a few of their contributions.

knowledge of the cardinality of the support at each layer. Here  $|X_{min}^{(l)}|$  and  $|X_{max}^{(l)}|$  are the smallest and largest non-zeros in  $\mathbf{X}^{(l)}$  respectively and note also that we have adopted the shortened notation  $\mu_l = \mu(\mathbf{A}^{(l)})$ .

### 3 An average case analysis of the D-CSC model

#### 3.1 Probabilistic bound for the recovery of activation pathways

Notable in the sparsity bound (2.5) is the presence of  $\mu_l$ , which is the factor that allows for a nontrivial  $S_l$ . Bounds of the form (2.5) are prevalent in the theory of sparse approximation, see for instance [16, Chapter 5], where it is known [17] that for a generic matrix  $\mathbf{B} \in \mathbb{R}^{m \times \gamma m}$  that  $\mu(\mathbf{B}) > m^{-1/2} \sqrt{1 - \gamma^{-1}}$ . This is colloquially referred to as the square-root bottleneck in that  $\mu^{-1} \sim m^{1/2}$ . In many applications, e.g. imaging,  $m_i$  is typically not more than  $7^2$  and  $n_i$  is typically about  $2m_i$ . Solving the sequence of sparse coding problems in (2.2) has the additional challenge due to  $\mathbf{A}^{(l)}$  being convolutional in structure. This can result in a large mutual coherence if the stride between shifted versions of the local dictionaries  $\mathbf{A}_{Local}^{(l)}$  is insufficient. As a result the proportionality of the bound on the signal complexity, measured in terms of the number of nonzeros, on  $\mu_l^{-1}$  in (2.5) limits its ability to provide guarantees in many practical situations.

It is well known from the work of Schnass and Vandergheynst [2] that in the single layer context, if one introduces a randomized sign pattern then the Rademacher concentration inequality can be used to derive bounds demonstrating that the recovery of the correct activations is *typically* possible with the sparsity bounds relaxed to depend on  $\mu_l^{-2}$ . Our main result is to extend the techniques used in [2] to the multi-layer setting of [1]. In order to do so we adapt Model (2.2) to include randomized sign patterns on the masks of the network, or equivalently activations at the next layer. That is the inclusion of the matrix  $\mathbf{D}^{(l)}$ , a diagonal matrix whose diagonal entries are independent Rademacher random variables as we have already discussed. This matrix applies a random sign pattern to the columns of  $\mathbf{A}^{(l)}$ . Although this matrix is primarily an artifact necessary for our analysis, it is interesting to note its connection with dropout, a technique commonly used when training DCNNs in which a random set of nodes (or columns of the weight matrix) are ignored in every batch. Indeed one can tentatively interpret  $\mathbf{D}^{(l)}$  as a special form of dropout which selects either the positive or negative signed column from a wider dictionary that contains both. Under this adaption we are able to provide Theorem 1.

**Theorem 1.** *Let  $\hat{\mathbf{X}}^{(0)}$  be a data matrix consistent with Model (2.2) with  $\|\mathbf{X}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$ ,  $\|\mathbf{V}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l$  for  $l = 0, \dots, L-1$ . Suppose that we know the cardinality of the support of each column of  $\mathbf{X}^{(l)}$  for  $l = 1, \dots, L$  and that we calculate each column of  $\hat{\mathbf{X}}^{(l)}$  independently using (2.4). Denote the event that the location of the non-zeros in  $\mathbf{X}^{(l)}$  and  $\hat{\mathbf{X}}^{(l)}$  exactly coincide for  $l = 0, 1, \dots, L$ , as  $Z_L$ , i.e., the event that the reverse and forward activation pathways are the same. The probability that this event doesn't hold,  $\bar{Z}_L$ , has an upper bound*

$$P(\bar{Z}_L) \leq 2dM \sum_{l=1}^L n_l \exp \left( - \frac{|X_{min}^{(l)}|^2}{8 \left( |X_{max}^{(l)}|^2 \mu_l^2 S_l + \zeta_{l-1}^2 \right)} \right). \quad (3.1)$$

Furthermore when  $Z_L$  does occur then for all  $j$ ,

$$\|\hat{\mathbf{x}}_j^{(l)} - \mathbf{x}_j^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l \quad (3.2)$$

where

$$\zeta_l = \sqrt{\|\hat{\mathbf{X}}^{(l)}\|_{0,\infty}^{P^{(l)}}} \left( \mu_l(S_l - 1)|X_{max}^{(l)}| + \zeta_{l-1} \right). \quad (3.3)$$

A key implication of Theorem 1 is that the derived probability bound scales proportional to  $\mu_l^{-2}$  across a given layer, rather than  $\mu_l^{-1}$ . To be precise, for a given representation  $\hat{\mathbf{x}}^{(l-1)}$  and an arbitrary  $\delta \in [0, 1]$ , then  $P(\bar{W}_l) \leq \delta$  if

$$S_l \leq \left( \frac{|x_{min}^{(l)}|^2}{8|x_{max}^{(l)}|^2 \ln\left(\frac{2Mn_l}{\delta}\right)} - \frac{\zeta_{l-1}^2}{|x_{max}^{(l)}|^2} \right) \mu_l^{-2} \quad (3.4)$$

is satisfied. For this theorem to be meaningful (since we require  $S_l \in \mathbb{Z}^{\geq 0}$ ) it is necessary that the noise satisfies

$$\zeta_{l-1} < \frac{|x_{min}^{(l)}|}{\left(8 \ln\left(\frac{2Mn_l}{\delta}\right)\right)^{1/2}}.$$

### 3.2 Proof of Theorem 1

We develop the proof of Theorem 1 by first considering the failure to recover the sparse representation of a single vector across a single layer (Lemma 1) and then a single vector across multiple layers (Lemma 2). With these results in place it is a simple step to prove Theorem 1. In what follows we will make use of the following Rademacher concentration inequality.

**Theorem 2** (Rademacher concentration [18]). *Let  $\alpha$  be an arbitrary real vector and  $\varepsilon$  a random vector whose elements are independent random variables pulled from a Rademacher distribution  $\{-1, 1\}$ . Then for all  $t \in \mathbb{R}_+$*

$$P\left(\left|\sum_i \varepsilon_i \alpha_i\right| > t\right) \leq 2 \exp\left(-\frac{t^2}{2\|\alpha\|_2^2}\right). \quad (3.5)$$

#### 3.2.1 Recovery of the activation of a single vector across a single layer

Consider the recovery of the support, or reverse pass activation pattern, of a single vector across a single layer of model 2.2; the signal model for this layer we can write as

$$\hat{\mathbf{x}}^{(l-1)} = \mathbf{A}\mathbf{D}\mathbf{x}^{(l)} + \mathbf{v}^{(l-1)}, \quad (3.6)$$

where we have omitted the superscript layer notation on the matrices  $\mathbf{A}$  and  $\mathbf{D}$  for typographical clarity. For signals consistent with model (3.6) we are able to provide Lemma 1.

**Lemma 1.** *Suppose we have an estimate of the representation of a data point, generated under model 2.1, at layer  $l - 1$ , which we denote  $\hat{\mathbf{x}}^{(l-1)}$  and assume it is consistent with model (3.6). Let  $\mathbf{x}^{(l)}$ , whose support we denote as  $\Lambda$ , be a solution to (3.6) with  $\|\mathbf{x}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$ ,  $\|\mathbf{v}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_{l-1}$  and assume that we know the cardinality of the support,  $|\Lambda|$ . Suppose that  $\hat{\mathbf{x}}^{(l)}$  is computed according to 2.4. Denoting the event that the locations of the nonzeros in  $\mathbf{x}^{(l)}$  and  $\hat{\mathbf{x}}^{(l)}$  differ as  $\bar{W}_\Lambda^{(l)}$ , then*

$$P(\bar{W}_\Lambda^{(l)}) \leq 2nM \exp\left(-\frac{|x_{min}^{(l)}|^2}{8\left(|x_{max}^{(l)}|^2 \mu_l^2 S_l + \zeta_{l-1}^2\right)}\right). \quad (3.7)$$

Furthermore in the setting where the location of the nonzeros in  $\mathbf{x}^{(l)}$  and  $\hat{\mathbf{x}}^{(l)}$  are the same, then

$$\|\hat{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l$$

where

$$\zeta_l = \sqrt{\|\hat{\mathbf{x}}^{(l)}\|_{0,\infty}^{P^{(l)}}} \left( \mu_l(S_l - 1)|x_{max}^{(l)}| + \zeta_{l-1} \right).$$

*Proof.* Lemma 1 extends bounds in [2] to include additive noise and the stripe sparsity model  $\|\cdot\|_{0,\infty}^{Q^{(l)}} \leq S_l$  present for the convolutional matrices  $\mathbf{A}^{(l)}$  considered here.

Firstly denote the diagonal elements of  $\mathbf{D}$ ,  $\mathbf{D}_{jj}$ , as  $\varepsilon_j$ , which are mutually independent random variables with value either  $-1$  or  $1$  with probability  $0.5$ . For  $\bar{W}_\Lambda^{(l)}$  to occur requires the condition that there exists some  $i \in \Lambda$  and some  $k \in \bar{\Lambda}$  such that  $|\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle| < |\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|$ . This condition is equivalent to requiring  $\min_{i \in \Lambda} |\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle| < \max_{k \in \bar{\Lambda}} |\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|$  and hence

$$P(\bar{W}_\Lambda^{(l)}) = P(\min_i |\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle| < \max_k |\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|).$$

Introducing an arbitrary real valued scalar threshold  $p > 0$ , we form the inequality

$$P(\bar{W}_\Lambda^{(l)}) \leq P(\min_{i \in \Lambda} |\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle| < p) + P(\max_{k \in \bar{\Lambda}} |\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle| > p).$$

We now provide bounds on each of the terms on the right hand side of the above inequality using the Rademacher concentration inequality provided in Theorem 2. Denoting  $\bar{W}'_\Lambda$  as the event  $\max_{k \notin \Lambda} \{|\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|\} > p$  then

$$\begin{aligned} P(\bar{W}'_\Lambda) &= P\left(\max_k \{|\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|\} > p\right) \\ &\leq \sum_{k \in \bar{\Lambda}} P\left(|\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle| > p\right) \\ &= \sum_{k \in \bar{\Lambda}} P\left(\left|\sum_{j \in \Lambda} \varepsilon'_j x_j^{(l)} \langle \mathbf{a}_k, \mathbf{a}_j \rangle + \varepsilon_k \langle \mathbf{a}_k, \mathbf{v} \rangle\right| > p\right) \\ &\leq 2 \sum_{k \in \bar{\Lambda}} \exp\left(\frac{-p^2}{2\left(\sum_{j \in \Lambda \cap \Gamma} |x_j^{(l)}|^2 |\langle \mathbf{a}_k, \mathbf{a}_j \rangle|^2 + \zeta_{l-1}^2\right)}\right) \\ &\leq 2(mN - |\Lambda|) \exp\left(\frac{-p^2}{2\left(|x_{max}^{(l)}|^2 S_l \mu_l^2 + \zeta_{l-1}^2\right)}\right). \end{aligned} \tag{3.8}$$

The second line and inequality arises from  $\max_{k \in \bar{\Lambda}} \{|\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|\} > p$  implying  $\cup_{k \in \bar{\Lambda}} \{|\langle \varepsilon_k \mathbf{a}_k, \hat{\mathbf{x}}^{(l-1)} \rangle|\} > p$ . The third line is a simple expansion of the inner product, where we define a new Rademacher random variable  $\varepsilon'_j = \varepsilon_j \varepsilon_k$ . Note that the set of random variables  $\{\cup_j \varepsilon'_j\} \cup \varepsilon_k$  is also mutually independent. Moving from the third to the fourth line we use Theorem 2. The set  $\Gamma$  refers to the indices of columns of  $\mathbf{A}^{(l)}$  which have a nonzero inner product with the column  $\mathbf{a}_k$ . The final line is the result given that  $|\Lambda \cap \Gamma| \leq S_l$ , which follows from our assumption that  $\|\mathbf{x}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$ .

Denoting  $\bar{W}''_\Lambda$  as the event  $\min_i \{|\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle|\} < p$ , we firstly expand the inner product and then use the triangle inequality to derive

$$|\langle \varepsilon_i \mathbf{a}_i, \hat{\mathbf{x}}^{(l-1)} \rangle| \geq |x_i| - \left| \sum_{j \in \Lambda, j \neq i} \varepsilon'_j x_j^{(l)} \langle \mathbf{a}_i, \mathbf{a}_j \rangle + \varepsilon_i \langle \mathbf{a}_i, \mathbf{v} \rangle \right|.$$



We are then able to bound  $P(\bar{W}_\Lambda'')$  using essentially the same steps as before for  $P(\bar{W}_\Lambda')$ .

$$\begin{aligned}
P(\bar{W}_\Lambda'') &\leq P\left(\max_{i \in \Lambda} \left\{ \left| \sum_{j \in \Lambda, j \neq i} \varepsilon'_j x_j^{(l)} \langle \mathbf{a}_i, \mathbf{a}_j \rangle + \varepsilon_i \langle \mathbf{a}_i, \mathbf{v} \rangle \right| \right\} > |x_{min}^{(l)}| - p \right) \\
&\leq \sum_{i \in \Lambda} P\left(\left| \sum_{j \in \Lambda, j \neq i} \varepsilon'_j x_j^{(l)} \langle \mathbf{a}_i, \mathbf{a}_j \rangle + \varepsilon_i \langle \mathbf{a}_i, \mathbf{v} \rangle \right| > |x_{min}^{(l)}| - p \right) \\
&\leq 2 \sum_{i \in \Lambda} \exp\left(-\frac{(|x_{min}^{(l)}| - p)^2}{2 \left( \sum_{j \in \Lambda \cap \Gamma/i} |x_j|^2 |\langle \mathbf{a}_k, \mathbf{a}_j \rangle|^2 + \zeta_{l-1}^2 \right)}\right) \\
&\leq 2|\Lambda| \exp\left(\frac{-(|x_{min}^{(l)}| - p)^2}{2 \left( |x_{max}^{(l)}|^2 S_l \mu_l^2 + \zeta_{l-1}^2 \right)}\right).
\end{aligned}$$

The first line is a result of rearranging and minimising the expanded inner product with respect to being less than  $p$ . The subsequent lines follow in the same manner as in (3.8). Since  $p$  can be arbitrary, and since it is equidistant between the expectations of the two distributions of  $|\langle \varepsilon_i \mathbf{a}_i, y \rangle|$  and  $|\langle \varepsilon_k \mathbf{a}_k, y \rangle|$ , choose  $p = |x_{min}|/2$ . As a result combining the bounds on  $P(\bar{W}_\Lambda')$  and  $P(\bar{W}_\Lambda'')$  we obtain the desired probability bound on  $P(\bar{W}_\Lambda)$ .

To now bound the error under the assumption that the support  $\Lambda$  is recovered we provide a proof very similar to that provided in Theorem 8 of [1]. Firstly

$$\begin{aligned}
\|\hat{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)}\|_{2,\infty}^{P^{(l)}} &= \max_i \|P_i \mathbf{x}^{(l)} - P_i \hat{\mathbf{x}}^{(l)}\|_2 \\
&= \sqrt{\|\mathbf{x}^{(l)}\|_{0,\infty}^{P^{(l)}}} \left( \max_i \|P_i \mathbf{x}^{(l)} - P_i \hat{\mathbf{x}}^{(l)}\|_\infty \right) \\
&\leq \sqrt{\|\mathbf{x}^{(l)}\|_{0,\infty}^{P^{(l)}}} \left( \|\mathbf{x}^{(l)} - \hat{\mathbf{x}}^{(l)}\|_\infty \right).
\end{aligned}$$

The first equality follows from the definition of the measure  $\|\cdot\|_{2,\infty}^{P^{(l)}}$ . In the second inequality we use the vector norm relationship that for  $\mathbf{z} \in \mathbb{R}^m$ , with  $k$  nonzeros, then  $\|\mathbf{z}\|_2 \leq \sqrt{k} \|\mathbf{z}\|_\infty$ . Note that  $\hat{\mathbf{x}}^{(l)}$  and  $\mathbf{x}^{(l)}$  have the same support. The inequality on the third line is due to fact that the largest element in a vector is at least as large as the largest element of any subset of elements of the vector. Since  $\|\mathbf{x}^{(l)} - \hat{\mathbf{x}}^{(l)}\|_\infty = \|\mathbf{x}_\Lambda^{(l)} - \hat{\mathbf{x}}_\Lambda^{(l)}\|_\infty$ , and recalling that the  $\ell_\infty$  matrix norm is the maximum row sum of the absolute elements in each row of the matrix, then

$$\begin{aligned}
\|\mathbf{x}_\Lambda^{(l)} - \hat{\mathbf{x}}_\Lambda^{(l)}\|_\infty &= \|(\mathbf{A}_\Lambda \mathbf{D}_\Lambda)^+ (\mathbf{A}_\Lambda \mathbf{D}_\Lambda) \mathbf{x}_\Lambda^{(l)} - (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)^T \mathbf{x}^{(l-1)}\|_\infty \\
&= \|(\mathbf{I} - (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)^T (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)) \mathbf{x}_\Lambda^{(l)} - (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)^T \mathbf{v}^{(l-1)}\|_\infty \\
&\leq \|(\mathbf{I} - \mathbf{A}_\Lambda^T \mathbf{A}_\Lambda)\|_\infty \|\mathbf{x}_\Lambda^{(l)}\|_\infty + \|\mathbf{A}_\Lambda^T \mathbf{v}^{(l-1)}\|_\infty \\
&\leq \mu(\|\mathbf{x}^{(l)}\|_{0,\infty}^{Q^{(l)}} - 1) |x_{max}| + \zeta_{l-1}.
\end{aligned}$$

The equality in line 1 uses  $\mathbf{I} = (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)^+ (\mathbf{A}_\Lambda \mathbf{D}_\Lambda)$  and the definition of the estimate derived using (2.4). The equality in line 2 is obtained by introducing a positive and negative  $\mathbf{A}_\Lambda \mathbf{D}_\Lambda \mathbf{x}_\Lambda^{(l)}$  (which cancel one another hence preserving equality) and then rearranging. The inequality on the third line is obtained by applying the triangle inequality and then using the submultiplicative property of the induced matrix norm. The fourth and final inequality is a result of the definition of the  $\ell_\infty$

matrix norm;  $\|(\mathbf{I} - \mathbf{A}_\Lambda^T \mathbf{A}_\Lambda)\|_\infty = \max_i \sum_{j \in \Lambda} |1 - \langle \mathbf{a}_i, \mathbf{a}_j \rangle|$ . The diagonal elements are all zero since  $|\langle \mathbf{a}_i, \mathbf{a}_i \rangle| = 1$  and by the definition of the mutual coherence the off-diagonal elements are bounded as  $|\langle \mathbf{a}_i, \mathbf{a}_j \rangle| \leq \mu_l$ . Defining  $\Gamma$  as the set of indices of columns whose supports overlap with  $\mathbf{a}_i$  then

$$\begin{aligned} \max_i \sum_{j \in \Lambda} |1 - \langle \mathbf{a}_i, \mathbf{a}_j \rangle| &= \max_i \sum_{j \in (\Lambda \cap \Gamma)/i} |1 - \langle \mathbf{a}_i, \mathbf{a}_j \rangle| \\ &\leq \mu^{(l)} (\|\mathbf{x}^{(l)}\|_{0,\infty}^{Q^{(l)}} - 1). \end{aligned}$$

Finally since  $\|\mathbf{A}_\Lambda^T \mathbf{v}^{(l-1)}\|_\infty = \max_i |\langle \mathbf{a}_i, \mathbf{v}^{(l-1)} \rangle|$ , then, since  $\mathbf{a}_i$  is only supported over  $m_{l-1}$  elements,  $|\langle \mathbf{a}_i, \mathbf{v}^{(l-1)} \rangle| \leq \zeta_{l-1}$ . Combining these results we arrive at the stability bound presented in Lemma 1.  $\square$

### 3.2.2 Recovery of the activations of a single vector across multiple layers

With the single layer, single vector case proven in Lemma 1, we now proceed to investigate the single vector, multilayer case.

**Lemma 2.** *Suppose we have a data point  $\hat{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)} + \mathbf{v}^{(0)}$ , where  $\mathbf{x}^{(0)}$  is a column from a matrix generated under Model (2.1) and  $\mathbf{v}^{(0)}$  is some noise. We estimate the latent representations of  $\mathbf{x}^{(0)}$  at every layer by recursively applying (2.4). Assume that there exists  $\{S_l\}$  and  $\{\zeta_l\}$  such that  $\|\mathbf{x}^{(l)}\|_{0,\infty}^{Q^{(l)}} \leq S_l$  and  $\|\mathbf{V}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l$  are satisfied for all  $l$ . If  $\bar{Y}_L$  denotes the event that the thresholding operation of the forward pass fails to exactly recover the set of supports  $\{\Lambda\}_{l=1}^L$  (or reverse activation paths), then*

$$P(\bar{Y}_L) \leq 2M \sum_{l=1}^L n_l \exp \left( - \frac{|x_{min}^{(l)}|^2}{8 \left( |x_{max}^{(l)}|^2 \mu_l^2 S_l + \zeta_{l-1}^2 \right)} \right). \quad (3.9)$$

Furthermore in the setting where the location of the nonzeros in  $\mathbf{x}^{(l)}$  and  $\hat{\mathbf{x}}^{(l)}$  are the same, we are able to characterise  $\{\zeta_l\}_{l=0}^{L-1}$  in a recursive manner as

$$\|\hat{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)}\|_{2,\infty}^{P^{(l)}} \leq \zeta_l,$$

where

$$\zeta_l = \sqrt{\|\hat{\mathbf{x}}^{(l)}\|_{0,\infty}^{P^{(l)}}} \left( \mu_l (S_l - 1) |x_{max}^{(l)}| + \zeta_{l-1} \right).$$

*Proof.* This result can be proved easily via induction. For the sake of convenience let

$$\gamma^{(l)} = 2M n_l \exp \left( - \frac{|x_{min}^{(l)}|^2}{8 \left( |x_{max}^{(l)}|^2 \mu_l^2 S_l + \zeta_{l-1}^2 \right)} \right).$$

Note that the bound on the error at each layer is conditioned on the correct recovery of the support at the previous layer. Letting  $\bar{W}_\Lambda^{(l)}$  be the event that the support at the  $l$ th layer is not correctly recovered, then

$$\begin{aligned} P(\bar{W}_\Lambda^{(1)}) &\leq \gamma^{(1)} \\ P(\bar{W}_\Lambda^{(2)} | W_\Lambda^{(1)}) &\leq \gamma^{(2)} \\ &\dots \\ P(\bar{W}_\Lambda^{(l)} | \cap_i^{(l-1)} W_\Lambda^{(i)}) &\leq \gamma^{(l)}. \end{aligned}$$

The result for  $L = 1$  is trivial so we will proceed with the proof by induction by considering the base case  $L = 2$ . Applying De Morgan's theorem to  $\bar{Y}_2$  we obtain

$$\begin{aligned}
P(\bar{Y}_2) &= P(\bar{W}_\Lambda^{(1)} \cup \bar{W}_\Lambda^{(2)}) \\
&= P(\bar{W}_\Lambda^{(1)}) + P(\bar{W}_\Lambda^{(2)} \cap W_\Lambda^{(1)}) \\
&= P(\bar{W}_\Lambda^{(1)}) + P(\bar{W}_\Lambda^{(2)} | W_\Lambda^{(1)}) P(W_\Lambda^{(1)}) \\
&\leq \gamma^{(1)} + \gamma^{(2)} P(W_\Lambda^{(1)}) \\
&\leq \gamma^{(1)} + \gamma^{(2)}.
\end{aligned}$$

Hence our theorem is correct for  $L = 1$  and  $L = 2$ . Now assume that the result holds true for the  $k$ th layer, i.e.

$$P(\bar{Y}_k) \leq \sum_{l=1}^k \gamma^{(l)}.$$

Considering  $\bar{Y}_{k+1}$  then

$$\begin{aligned}
P(\bar{Y}_{k+1}) &= P(\bar{W}_\Lambda^{(k+1)} \cup \bar{Y}_k) \\
&= P(\bar{Y}_k) + P(\bar{W}_\Lambda^{(k+1)} \cap Y_k) \\
&= P(\bar{Y}_k) + P(\bar{W}_\Lambda^{(k+1)} | Y_k) P(Y_k) \\
&\leq \sum_{l=1}^k \gamma^{(l)} + \gamma^{(k+1)} P(Y_k) \\
&\leq \sum_{l=1}^k \gamma^{(l)} + \gamma^{(k+1)} \\
&= \sum_{l=1}^{k+1} \gamma^{(l)}.
\end{aligned}$$

This proves the result holds for the  $k + 1$ th case and given this and our base case hold then all others must follow. The bound on the error at each layer follows in exactly the same manner as that of Lemma 1.  $\square$

### 3.2.3 Recovery of the activations of multiple vectors across multiple layers

With Lemma 1 and 2 in place it is a fairly simple step to derive Theorem 1. Suppose that  $Y_j^{(L)}$  is the event that supports  $\{\Lambda_j^{(l)}\}_{l=1}^L$  of the representations  $\{\mathbf{x}_j^{(l)}\}_{l=1}^L$  of the  $j$ th data point are recovered from  $\{\hat{\mathbf{x}}_j^{(l)}\}_{l=1}^L$ . Let

$$Z_L \triangleq \bigcap_{j=1}^d Y_j^{(L)}.$$

Applying De Morgan’s Theorem then  $\bar{Z}^{(L)} = \bigcup_{j=1}^d \bar{Y}_j^{(L)}$ . As a result we obtain

$$\begin{aligned} P(\bar{Z}_L) &\leq \sum_{j=1}^d P(\bar{Y}_j^{(L)}) \\ &\leq 2M \sum_{j=1}^d \sum_{l=1}^L \gamma_j^{(l)} \\ &\leq 2Md \sum_{l=1}^L \gamma_{max}, \end{aligned}$$

which gives the probability bound claimed in Theorem 1.

## 4 Conclusion

Interpreting the forward pass algorithm as a sparse coding problem allows us to make claims and conditions sufficient in theory for the successful operation of the forward pass. Our contributions in this paper are a) an approach to carrying out an average rather than worst case analysis for the recovery of activation pathways in deep sparse coding networks (as stated in 2.2) and b) Theorem 1, which extends the prior uniform bounds in [1] to high probability bounds. The key benefit of this result is the proportionality to the dictionary coherence of the upper bound on the signal’s complexity (measured in terms of the number of non-zeros per stripe) improving from  $\mu^{-1}$  to  $\mu^{-2}$ . Assuming the weight matrices are suitably conditioned, this indicates that the forward pass algorithm is likely to recover the latent representations in Model 2.2 for a more complex (again measured in terms of the number of non-zeros per stripe) family of signals than previously thought. In summary, this suggests that if optimal sparse coding is indeed an important factor explaining the success of certain DCNN architectures, then explicitly encouraging weight matrices with low coherence during training could improve the DCNN’s performance.

## References

- [1] V. Pappayan, Y. Romano, and M. Elad. Convolutional Neural Networks Analyzed via Convolutional Sparse Coding. *ArXiv e-prints*, July 2016.
- [2] K. Schnass and P. Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, Nov 2007.
- [3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [4] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.
- [5] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Yoshua Bengio, and A. Courville. Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks. *ArXiv e-prints*, January 2017.

- [6] H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.
- [7] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. 2017.
- [8] Tomaso A. Poggio and Qianli Liao. Theory ii: Landscape of the empirical risk in deep learning. *CoRR*, abs/1703.09833, 2017.
- [9] C. D. Freeman and J. Bruna. Topology and Geometry of Half-Rectified Network Optimization. *ArXiv e-prints*, November 2016.
- [10] Benjamin D. Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *CoRR*, abs/1506.07540, 2015.
- [11] Joan Bruna and Stephane Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, August 2013.
- [12] T. Wiatowski, P. Grohs, and H. Bölcskei. Energy Propagation in Deep Convolutional Neural Networks. *ArXiv e-prints*, April 2017.
- [13] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *ArXiv e-prints*, November 2016.
- [14] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.
- [15] V. Pappyan, J. Sulam, and M. Elad. Working Locally Thinking Globally: Theoretical Guarantees for Convolutional Sparse Coding. *IEEE Transactions on Signal Processing*, 65:5687–5701, November 2017.
- [16] R. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2013.
- [17] L R. Welch. Lower bounds on the maximum cross correlation of signals (corresp.). 20:397 – 399, 06 1974.
- [18] M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Classics in Mathematics. Springer-Verlag Berlin Heidelberg, 2011.