Why Regularized Auto-Encoders Learn Sparse Representation?

Devansh Arpit DEVANSHA@BUFFALO.EDU
Yingbo Zhou YINGBOZH@BUFFALO.EDU
Hung Q. Ngo HUNGNGO@BUFFALO.EDU
Venu Govindaraju GOVIND@BUFFALO.EDU
SUNY Buffalo

Abstract

Sparse distributed representation is the key to learning useful features in deep learning algorithms, because not only it is an efficient mode of data representation, but also – more importantly – it captures the generation process of most real world data. While a number of regularized auto-encoders (AE) enforce sparsity explicitly in their learned representation and others don't, there has been little formal analysis on what encourages sparsity in these models in general. Our objective is to formally study this general problem for regularized auto-encoders. We provide sufficient conditions on both regularization and activation functions that encourage sparsity. We show that multiple popular models (de-noising and contractive auto encoders, e.g.) and activations (rectified linear and sigmoid, e.g.) satisfy these conditions; thus, our conditions help explain sparsity in their learned representation. Thus our theoretical and empirical analysis together shed light on the properties of regularization/activation that are conductive to sparsity and unify a number of existing auto-encoder models and activation functions under the same analytical framework.

1. Introduction

Sparse Distributed Representation (SDR) (Hinton, 1984) constitutes a fundamental reason behind the success of deep learning. On one hand, it is an efficient way of representing data that is robust to noise; in fact, some of the main advantages of sparse distributed representation in the context of deep neural networks has been shown to be information disentangling and manifold flattening (Bengio et al., 2013), as well as better linear separability and representational power (Glorot et al., 2011). On the other hand, and more importantly, SDR captures the data generation process itself and is biologically inspired (Hubel & Wiesel, 1959; Olshausen & Fieldt, 1997; Patterson et al., 2007), which makes this mode of representation useful in the first place.

For these reasons, our objective in this paper is to investigate why a number of regularized Auto-Encoders (AE) exhibit similar behaviour, especially in terms of learning sparse representations. AEs are especially interesting for this matter because of the clear distinction between their learned *encoder* representation and *decoder* output. This is in contrast with other deep models where there is no clear distinction between the encoder and decoder parts. The idea of AEs learning sparse representations (SR) is not new. Due to the aforementioned biological connection between SR and NNs, a natural follow-up pursued by a number of researchers was to propose AE variants that encouraged sparsity in their learned representation (Lee et al., 2008; Kavukcuoglu & Lecun, 2008; Ng, 2011). On the other hand, there has also been work on empirically analyzing/suggesting the sparseness of hidden representations learned after pre-training with unsupervised models (Memisevic et al., 2014; Li et al., 2013; Nair & Hinton, 2010). However, to the best of our knowledge, there has been no prior work formally analyzing why regularized AEs learn sparse representation in general. The main challenge behind doing so is the analysis of non-convex objective functions. In addition, questions regarding the efficacy of activation functions and the choice of regularization on AE objective are often raised since there are multiple available choices for both. We also try to address these questions with regards to SR in this paper.

We address these questions in two parts. First, we prove sufficient conditions on AE regularizations that encourage low pre-activations in hidden units. We then analyze the properties of activation functions that when coupled with such regularizations result in sparse representation. Multiple popular activations have these desirable properties. Second, we show that multiple popular AE objectives including *de-noising auto-encoder* (DAE) (Vincent et al., 2008) and *contractive auto-encoder* (CAE) (Rifai et al.,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

2011b) indeed have the suggested form of regularization; thus explaining why existing AEs encourage sparsity in their latent representation. Based on our theoretical analysis, we also empirically study multiple popular AE models and activation functions in order to analyze their comparative behaviour in terms of sparsity in the learned representations. Our analysis thus shows why various AE models and activations lead to sparsity. As a result, they are unified under a framework uncovering the fundamental properties of regularizations and activation functions that most of these existing models possess.

2. Auto-Encoders and Sparse Representation

Auto-Encoders (AE) (Rumelhart et al., 1986; Bourlard & Kamp, 1988) are a class of single hidden layer neural networks trained in an unsupervised manner. It consists of an *encoder* and a *decoder*. An input ($\mathbf{x} \in \mathbb{R}^n$) is first mapped to the latent space with $\mathbf{h} = f_e(\mathbf{x}) = s_e(\mathbf{W}\mathbf{x} + \mathbf{b}_e)$ is the hidden representation vector, s_e is the encoder activation, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight matrix, and $\mathbf{b}_e \in \mathbb{R}^m$ is the encoder bias. Then, it maps the hidden output back to the original space by $\mathbf{y} = f_d(\mathbf{h}) = s_d(\mathbf{W}^T\mathbf{h})$ where $\mathbf{y}$ is the reconstructed counterpart of $\mathbf{x}$ and s_d is the decoder activation. The objective of a basic auto-encoder is to minimize the following with respect to the parameters $\{\mathbf{W}, \mathbf{b}_e\}$

$$\mathcal{J}_{AE} = \mathbb{E}_{\mathbf{x}}[\ell(\mathbf{x}, f_d(f_e(\mathbf{x})))] \tag{1}$$

where $\ell(\cdot)$ is the squared loss function. The motivation behind this objective is to capture predominant repeating patterns in data. Thus although the auto-encoder optimization learns to map an input back to itself, the focus is on learning a noise invariant representation (manifold) of data.

2.1. Part I: What encourages *sparsity* during Auto-Encoder training?

2.1.1. Sparsity and our assumption

Learning a dictionary adapted to a set of training data such that the latent code is sparse is generally formulated as the following optimization problem (Olshausen & Fieldt, 1997)

$$\min_{\mathbf{W},\mathbf{h}} \quad \sum_{i=1}^{N} \left(\|\mathbf{x}_i - \mathbf{W}^T\mathbf{h}_i\|^2 + \lambda \|\mathbf{h}_i\|_1 \right) \tag{2}$$

The above objective is convex in each one of $\mathbf{W}$ and $\mathbf{h}$ when the other is fixed and hence it is generally solved alternately in each variable while fixing the other. Note that ℓ_1 penalty is the driving force in the above objective and forces the latent variable to be sparse.

This section analyses the factors that are required for sparsity in AEs. Note that in (2) we optimize for a different parameter $\mathbf{h}_i$ for each corresponding sample. In the case of AEs, we do not have a separate parameter that denotes the hidden representation corresponding to every sample individually. Instead the hidden representation for every sample is a function of the sample itself along with other network parameters. So in order to define the notion of sparsity of hidden representation in AEs, we will treat each hidden unit $h_i = s_e(\mathbf{W}_j\mathbf{x} + b_{e_j})$ as a random variable which itself is a function of the random variable $\mathbf{x}$. Then the average activation fraction of a unit is the (probability) mass of (data) distribution for which the hidden unit activates. For finite sample datasets, this becomes the fraction of data samples for which the unit activates.

Also note that SDR dictates that all representational units participate in data representation while very few units activate for a single data sample. Thus a major difference between SDR and SR is that of *dead units* (units that do not activate for any data sample) since sparsity can in general also be achieved when most units are dead. However, the latter scenario is undesirable because it does not truly capture SDR. Thus we model and study the conditions that encourage sparsity in hidden units; and we also empirically show these conditions are capable of achieving SDR.

For our analysis, we will use linear decoding which addresses the case of continuous real valued data distributions. We will now show that both regularization and activation function play an important role for achieving sparsity. In order to do so, we make the following assumption,

Assumption 1. *We assume that the data $\mathbf{x}$ is drawn from a distribution $\mathbf{x} \sim \mathcal{X}$ for which $\mathbb{E}_{\mathbf{x}}[\mathbf{x}] = \mathbf{0}$ and $\mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix.*

Further, let $\mathbf{r}_{\mathbf{x}}^t \triangleq \mathbf{x} - \mathbf{W}^T f_e(\mathbf{x})$ denote the reconstruction residual during auto-encoder training at any iteration t for training sample $\mathbf{x}$. Then we assume every dimension of $\mathbf{r}_{\mathbf{x}}^t$ is i.i.d. random variable following a Gaussian distribution with mean 0 and standard deviation σ_r.

Before proceeding, first we establish an important condition needed by AEs for exhibiting sparse behaviour. Consider the pre-activation of an AE

$$a_j^t = \mathbf{W}_j^t\mathbf{x} + b_{e_j}^t \tag{3}$$

Here j and t denote the j^{th} hidden unit and t^{th} training iteration respectively, and $\mathbf{W}_j^t$ denotes the j^{th} row of $\mathbf{W}$. Then notice when Assumption 1 is true, if we remove the encoding bias from the AE optimization, the expected pre-activation becomes $\mathbb{E}_{\mathbf{x}}[a_j^t] = \mathbb{E}_{\mathbf{x}}[\mathbf{W}_j^t\mathbf{x}] = 0$ unconditionally for all iterations. Consider any activation function $s_e(.)$ with activation threshold $\delta_{\min}$, i.e. any data sample with j^{th} pre-activation a_j^t would de-activate the unit if $a_j^t <= \delta_{\min}$ and activate it otherwise. Then the only way for a unit to exhibit sparse behaviour (over a data distribution) when the expected pre-activation is always zero, is for the majority of the samples to have pre-activation below $\delta_{\min}$. Then, in order for the average to be zero, the

minority above the threshold will have taken larger values on average compared to the majority. However, this strategy limits the degree of sparsity that a unit can achieve for any given data distribution following Assumption 1, when the weight lengths are upper bounded because the pre-activation value also become upper bounded. The bounded weight length condition is desired in practice for convergence and is achieved by regularizations like weight decay and Max-Norm (Hinton et al., 2012). Thus, in order for hidden units to exhibit sparse behaviour, encoding bias needs to be a part of AE optimization.

Having established the importance of encoding bias, we make the following deduction based on the above assumption,

Lemma 1. *If assumption 1 is true, and encoding activation function $s_e(.)$ has first derivative in $[0,1]$, then $\partial\mathcal{J}_{AE}/\partial b_{e_j} \in [-2\sigma_r\sqrt{n}\|\mathbf{W}_j\|, 2\sigma_r\sqrt{n}\|\mathbf{W}_j\|]$.*

Using the above result , the theorem below gives a *sufficient* condition on regularization functions needed for forcing the average pre-activation value ($\mathbb{E}[(a_j^t)]$) to keep on reducing after every training iteration.

Theorem 1. *Let $\{\mathbf{W}^t \in \mathbb{R}^{m\times n}, \mathbf{b}_e^t \in \mathbb{R}^m\}$ be the parameters of a regularized auto-encoder ($\lambda > 0$)*

$$\mathcal{J}_{RAE} = \mathcal{J}_{AE} + \lambda\mathcal{R}(\mathbf{W}, \mathbf{b}_e) \tag{4}$$

*at training iteration t with regularization term $\mathcal{R}(\mathbf{W}, \mathbf{b}_e)$, activation function $s_e(.)$ and define pre-activation $a_j^t = \mathbf{W}_j^t\mathbf{x} + b_{e_j}^t$ (thus $h_j^t = s_e(a_j^t)$). **If** $\lambda\frac{\partial\mathcal{R}}{\partial b_{e_j}} > 2\sigma_r\sqrt{n}\|\mathbf{W}_j\|$, where $j \in \{1, 2, \ldots, m\}$, **then** updating $\{\mathbf{W}^t, \mathbf{b}_e^t\}$ along the negative gradient of $\mathcal{J}_{RAE}$, results in $\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] < \mathbb{E}_{\mathbf{x}}[a_j^t]$ **and** $\mathrm{Var}[a_j^{t+1}] = \|\mathbf{W}_j^{t+1}\|^2$ **for** all $t \geq 0$.*

Interpretation: The important thing to notice in the above theorem is that larger values of λ is expected to lead to lower expected pre-activation values since,

$$\mathbb{E}_{\mathbf{x}}\left[a_j^{t+1}\right] = \mathbb{E}_{\mathbf{x}}\left[a_j^t\right] - \eta(\frac{\partial\mathcal{J}_{AE}}{\partial b_{e_j}} + \lambda\frac{\partial\mathcal{R}}{\partial b_{e_j}}) \tag{5}$$

where η is the learning rate. But this may not be true in general over multiple iterations due to terms in $\frac{\partial\mathcal{R}}{\partial b_{e_j}}$ that depend on weight vectors that also change every iteration depending on the value of λ. However, we are generally interested in the direction of the weight vectors during reconstruction instead of their scale. Thus if we fix the length of weight vectors (to say, unit length), then the term $\frac{\partial\mathcal{R}}{\partial b_{e_j}}$ will be bounded by a fixed term w.r.t. weight vectors and will only depend on the bias and data distribution. Under these circumstances, increasing the value of λ is conducive to lower expected pre-activation if $\frac{\partial\mathcal{R}}{\partial b_{e_j}}$ is strictly greater than zero. On the other hand, if $\frac{\partial\mathcal{R}}{\partial b_{e_j}} = 0$, then changing the value of λ should not have significant effect on expected pre-activation values, especially when the weight length is fixed. In the case when the weight length is not fixed, changing the value of λ will affect the value of weight length, which in turn will affect the term $\frac{\partial\mathcal{J}_{AE}}{\partial b_{e_j}}$ which also affects expected pre-activation of a unit; but this effect is largely unpredictable depending on the form of $\frac{\partial\mathcal{J}_{AE}}{\partial b_{e_j}}$. In the next section, we will connect the notions of expected pre-activation and sparsity, for activation functions with certain properties which will extend the above arguments to the sparsity of hidden units.

Finally, in the relaxed cases when weight lengths are not constrained to have a fixed length, an upper bound on weight vectors' length can easily be guaranteed using *Max-norm Regularization* or *Weight Decay* which are widely used tricks while training deep networks (Hinton et al., 2012). In the prior case every weight vector is simply constrained to lie within an ℓ_2 ball ($\|\mathbf{W}_j\|_2 \leq c\ \forall j \in [m]$, where c is a fixed constant) after every gradient update.

Having shown the property of regularization functions that encourages lower pre-activations, we now introduce two classes of regularization functions that inherit this property and thus manifest the predictions made above.

Corollary 1. ***If** s_e is a non-decreasing activation function with first derivative in $[0,1]$ **and** $\mathcal{R} = \sum_{j=1}^m f(\mathbb{E}_{\mathbf{x}}[h_j])$ for any monotonically increasing function $f(.)$, **then** $\exists\lambda > 0$ such that updating $\{\mathbf{W}^t, \mathbf{b}_e^t\}$ along the negative gradient of $\mathcal{J}_{RAE}$ results in $\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] \leq \mathbb{E}_{\mathbf{x}}[a_j^t]$ **and** $\mathrm{Var}[a_j^{t+1}] = \|\mathbf{W}_j^{t+1}\|^2$ **for** all $t \geq 0$.*

Corollary 2. ***If** s_e is a non-decreasing convex activation function with first derivative in $[0,1]$ **and** $\mathcal{R} = \mathbb{E}_{\mathbf{x}}\left[\sum_{j=1}^m \left(\left(\frac{\partial h_j}{\partial a_j}\right)^q \|\mathbf{W}_j^t\|_2^p\right)\right]$, $q \in \mathbb{N}$, $p \in \mathbb{W}$, **then** $\exists\lambda > 0$ such that updating $\{\mathbf{W}^t, \mathbf{b}_e^t\}$ along the negative gradient of $\mathcal{J}_{RAE}$, results in $\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] \leq \mathbb{E}_{\mathbf{x}}[a_j^t]$ **and** $\mathrm{Var}[a_j^{t+1}] = \|\mathbf{W}_j^{t+1}\|^2$ **for** all $t \geq 0$.*

Above corollaries show that specific regularizations encourage the pre-activation of every hidden unit in AEs to reduce on average, with assumptions made only on activation function and the first/second order statistics of the data distribution. We will show in Section 2.2 that multiple existing AEs have regularizations of the form above.

2.1.2. Which activation functions are *good* for Sparse Representation?

The above analysis in general suggests that non-decreasing convex activation functions encourage lower expected pre-activation for regularization in both corollaries. Also note that a reduction in the expected pre-activation value ($\mathbb{E}[(a_j^t)]$) does not necessarily imply a reduction in the hidden unit value (h_j^t) and thus sparsity. However, these regularizations become immediately useful if we consider non-

decreasing activation functions with negative saturation at 0, *i.e.*, $\lim_{a\to-\infty} s_e(a) = 0$. Now a lower average pre-activation value directly implies higher sparsity!

Before proceeding, we would like to mention that although the general notion of sparsity in AEs entails majority of units are de-activated, *i.e.*, their value is less than a certain threshold ($\delta_{\min}$), in practice, a representation that is *truly sparse* (large number of hard zeros) usually yields better performance (Glorot et al., 2011; Wright et al., 2009; Yang et al., 2009). Extending the argument of theorem 1, we obtain:

Theorem 2. *Let p_j^t denote a lower bound of $\Pr(h_j^t \leq \delta_{\min})$ at iteration t and $s_e(.)$ be a non-decreasing function with first derivative in $[0, 1]$.* ***If*** *$\|\mathbf{W}_j^t\|_2$ is upper bounded independent of λ* ***then*** *$\exists S \subseteq \mathbb{R}^+$* ***and*** *$\exists T_{\min}, T_{\max} \in \mathbb{N}$* ***such that*** *$p_j^{t+1} \geq p_j^t \; \forall \lambda \in S$, $T_{\min} \leq t \leq T_{\max}$.*

The above theorem formally connects the notions of expected pre-activation and expected sparsity of a hidden unit. Specifically, it shows that the usage of non-decreasing activation functions lead to lower expected pre-activation and thus a higher probability of de-activated hidden units when theorem 1 applies. This result coupled with the property $\lim_{a\to-\infty} s_e(a) = 0$ (de-activated state) implies the average sparsity of hidden units keeps increasing after a sufficient number of iterations ($T_{\min}$) for such activations. Notice that convexity in $s_e(.)$ is only desired for regularizations in corollary 2. Thus in summary, non-decreasing convex $s_e(.)$ ensure $\partial\mathcal{R}/\partial b_{e_j}$ is positive for regularizations in corollary 1 and 2, which in turn encourages low expected pre-activation for suitable values of λ. This finally leads to higher sparsity if $\lim_{a\to-\infty} s_e(a) = 0$.

Notice we derive the strict inequality ($\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] < \mathbb{E}_{\mathbf{x}}[a_j^t]$) in Theorem 1 (and used in Theorem 2) even though the corollaries suggest non-decreasing convex activations imply the relaxed case ($\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] \leq \mathbb{E}_{\mathbf{x}}[a_j^t]$). This is done for two reasons: a) ensure sparsity monotonically increases for iterations $T_{\min} \leq t \leq T_{\max}$, b) the condition $\partial\mathcal{R}/\partial b_{e_j} = 0$ (which results in $\mathbb{E}_{\mathbf{x}}[a_j^{t+1}] \leq \mathbb{E}_{\mathbf{x}}[a_j^t]$) is unlikely for activations with non-zero first/second derivatives because the term $\mathcal{R}$ (above corollaries) depends on the entire data distribution.

The most popular choice of activation functions are ReLU, Maxout(Goodfellow et al., 2013), Sigmoid, Tanh and Softplus. Maxout and Tanh are not applicable to our framework as they do not satisfy the negative saturation property.

ReLU: It is a non-decreasing convex function; thus both corollary 1 and 2 apply. Note ReLU does not have a second derivative[1]. Thus, in practice, this may lead to poor sparsity for the regularization in Corollary 2 due to lack of bias gradients from the regularization, i.e. $\partial\mathcal{R}/\partial b_{e_j} = 0$. On the flip side, the advantage of ReLU is that it enforces hard zeros in the learned representations.

Softplus: It is a non-decreasing convex function and hence encourages sparsity for the suggested AE regularizations. In contrast to ReLU, Softplus has positive bias gradients (hence better sparsity for corollary 2) because of its smoothness. On the other hand, note that Softplus does not produce hard zeros due to asymptotic left saturation at 0.

Sigmoid: Corollary 1 applies unconditionally to Sigmoid, while corollary 2 doesn't apply in general. Hence Sigmoid is not guaranteed to lead to sparsity when used with regularizations of form specified in Corollary 2.

Notice all the above activation functions have their first derivative in $[0, 1]$ (a condition required by lemma 1). In conclusion, Maxout and Tanh do not satisfy the negative saturation property at 0 and hence do not guarantee sparsity, all others– ReLU, Softplus and Sigmoid– have properties (at least in principle) that encourage sparsity in learned representations for the suggested regularizations.

2.2. Part II: Do existing Auto-Encoders learn Sparse Representation?

At this point, a natural question to ask is whether existing AEs learn Sparse Representation. To complete the loop, we show that most of the popular AE objectives have regularization term similar to what we have proposed in Corollaries 1 and 2 and thus they indeed learn sparse representation.

1) De-noising Auto-Encoder (DAE): DAE (Vincent et al., 2008) aims at minimizing the reconstruction error between every sample $\mathbf{x}$ and the reconstructed vector using its corresponding corrupted version $\tilde{\mathbf{x}}$. The corrupted version $\tilde{\mathbf{x}}$ is sampled from a conditional distribution $p(\tilde{\mathbf{x}}_i|\mathbf{x}_i)$. The original DAE objective is given by

$$\mathcal{J}_{DAE} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}[\ell(\mathbf{x}, f_d(f_e(\tilde{\mathbf{x}})))]\right] \tag{6}$$

where $p(\tilde{\mathbf{x}}_i|\mathbf{x})$ denotes the conditional distribution of $\tilde{\mathbf{x}}$ given $\mathbf{x}$. Since the above objective is analytically intractable due to the corruption process, we take a second order Taylor's approximation of the DAE objective around the distribution mean $\mu_{\mathbf{x}} = \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}[\tilde{\mathbf{x}}]$ in order to overcome this difficulty,

Theorem 3. *Let $\{\mathbf{W}, \mathbf{b}_e\}$ represent the parameters of a DAE with squared loss, linear decoding, and i.i.d. Gaussian corruption with zero mean and σ^2 variance, at any point of training over data sampled from distribution $\mathcal{D}$. Let $a_j := \mathbf{W}_j\mathbf{x} + b_{e_j}$ so that $h_j = s_e(a_j)$ corresponding*

[1] In other words, $\partial^2 h_j/\partial a_j^2 = \delta(\mathbf{W}_j\mathbf{x} + b_{e_j})$, where $\delta(.)$ is the Dirac delta function. Although strictly speaking, $\partial^2 h_j/\partial a_j^2$ is always non-negative, this value is zero everywhere except when the argument is exactly 0, in which case it is $+\infty$

to sample $\mathbf{x} \sim \mathcal{D}$. *Then,*

$$\mathcal{J}_{DAE} = \mathcal{J}_{AE} + \sigma^2 \mathbb{E}_{\mathbf{x}} \left[\sum_{j=1}^{m} \left(\left(\frac{\partial h_j}{\partial a_j} \right)^2 \|\mathbf{W}_j\|_2^4 \right) + \sum_{\substack{j,k=1 \\ j \neq k}}^{m} \left(\frac{\partial h_j}{\partial a_j} \frac{\partial h_k}{\partial a_k} (\mathbf{W}_j^T \mathbf{W}_k)^2 \right) + \sum_{i=1}^{n} \left((\mathbf{b}_d + \mathbf{W}^T \mathbf{h} - \mathbf{x})^T \mathbf{W}^T \left(\frac{\partial^2 \mathbf{h}}{\partial \mathbf{a}^2} \odot \mathbf{W}^i \odot \mathbf{W}^i \right) \right) \right] + o(\sigma^2) \tag{7}$$

where $\frac{\partial^2 \mathbf{h}}{\partial \mathbf{a}^2} \in \mathbb{R}^m$ *is the element-wise* 2^{nd} *derivative of* $\mathbf{h}$ *w.r.t.* $\mathbf{a}$ *and* $\odot$ *is element-wise product.*

The first term of the above regularization is of the form stated in corollary 2. Even though the second term doesn't have the exact suggested form, it is straight forward to see that this term generates non-negative bias gradients for non-decreasing convex activation functions (and should have behaviour similar to that predicted in corollary 2). Note the last term depends on the reconstruction error which practically becomes small after a few epochs of training and the other two regularization terms take over. Besides, this term is usually ignored as it is not positive-definite. This suggests that DAE is capable of learning sparse representation.

2) Contractive Auto-Encoder (CAE): CAE (Rifai et al., 2011b) objective is given by

$$\mathcal{J}_{CAE} = \mathcal{J}_{AE} + \lambda \mathbb{E}_{\mathbf{x}} \left[\|J(\mathbf{x})\|_F^2 \right] \tag{8}$$

where $J(\mathbf{x}) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ denotes the Jacobian matrix and the objective aims at minimizing the sensitivity of the hidden representation to slight changes in input.

Remark 1. *Let* $\{\mathbf{W}, \mathbf{b}_e\}$ *represent the parameters of a CAE with regularization coefficient* λ*, at any point of training over data sampled from some distribution* $\mathcal{D}$*. Then,*

$$\mathcal{J}_{CAE} = \mathcal{J}_{AE} + \lambda \mathbb{E}_{\mathbf{x}} \left[\sum_{j=1}^{m} \left(\left(\frac{\partial h_j}{\partial a_j} \right)^2 \|\mathbf{W}_j\|_2^2 \right) \right] \tag{9}$$

Thus CAE regularization also has a form identical to the form suggested in corollary 2. Thus the hidden representation learned by CAE should also be sparse. In addition, since the first order regularization term in Higher order CAE (CAE+H) (Rifai et al., 2011a) is the same as CAE, this suggests that CAE+H objective should have similar properties in term of sparsity.

3) Marginalized De-noising Auto-Encoder (mDAE): mDAE (Chen et al., 2014) objective is given by:

$$\mathcal{J}_{mDAE} = \mathcal{J}_{AE} + \frac{1}{2} \mathbb{E}_{\mathbf{x}} \left[\sum_{i=1}^{n} \sigma_{\mathbf{x}i}^2 \sum_{j=1}^{m} \frac{\partial^2 \ell}{\partial {h_j}^2} \left(\frac{\partial h_j}{\partial \tilde{x}_i} \right)^2 \right] \tag{10}$$

where $\sigma_{\mathbf{x}i}^2$ denotes the corruption variance intended for the i^{th} input dimension. The authors of mDAE proposed this algorithm with the primary goal of speeding up the training of DAE by deriving an approximate form that omits the need to iterate over a large number of explicitly corrupted instances of every training sample.

Remark 2. *Let* $\{\mathbf{W}, \mathbf{b}_e\}$ *represent the parameters of a mDAE with linear decoding, squared loss and* $\sigma_{\mathbf{x}i}^2 = \lambda$ $\forall i$*, at any point of training over data sampled from some distribution* $\mathcal{D}$*. Then,*

$$\mathcal{J}_{mDAE} = \mathcal{J}_{AE} + \lambda \mathbb{E}_{\mathbf{x}} \left[\sum_{j=1}^{m} \left(\left(\frac{\partial h_j}{\partial a_j} \right)^2 \|\mathbf{W}_j\|_2^4 \right) \right] \tag{11}$$

Apart from justifying sparsity in the above AEs, these equivalences also expose the similarity between DAE, CAE and mDAE regularization as they all follow the form in corollary 2. Note how the goal of achieving invariance in hidden and original representation respectively in CAE and mDAE show up as a mere factor of weight length in their regularization in the case of linear decoding.

4) Sparse Auto-Encoder (SAE): Sparse AEs are given by:

$$\mathcal{J}_{SAE} = \mathcal{J}_{AE} + \lambda \sum_{j=1}^{m} (\rho \log(\rho/\rho_j) + (1-\rho) \log((1-\rho)/(1-\rho_j))) \tag{12}$$

where $\rho_j = \mathbb{E}_{\mathbf{x}}[h_j]$ and ρ is the desired average activation (typically close to 0). Thus SAE requires one additional parameter (ρ) that needs to be pre-determined. To make SAE follow our paradigm, we set $\rho = 0$ and thus tuning the value of λ would automatically enforce a balance between the final level of average sparsity and reconstruction error. Thus the SAE objective becomes

$$\mathcal{J}_{SAE} = \mathcal{J}_{AE} - \lambda \sum_{j=1}^{m} \log(1-\rho_j) \quad (when \;\; \rho = 0) \tag{13}$$

Note for small values of ρ_j, $\log(1-\rho_j) \approx -\rho_j$. Thus the above objective has a very close resemblance with sparse coding (equation 2, except that SC has a non-parametric encoder). On the other hand, the above regularization has

a form as specified in corollary 1 which we have showed enforces sparsity. Thus, although it is expected of the SAE regularization to enforce sparsity from an intuitive standpoint, our results show that it indeed does so from a more theoretical perspective.

3. Empirical Analysis and Observations

We use the following two datasets for our experiments:

1. MNIST (Lecun & Cortes): It is a 10 class dataset of handwritten digit images of which $50,000$ images are provided for training.

2. CIFAR-10 (Krizhevsky, 2009): It consists of 60,000 32×32 color images of objects in 10 classes. For CIFAR-10, we randomly crop $50,000$ patches of size 8×8 for training the auto-encoders.

Experimental Protocols: Since neural network (NN) optimization is non-convex, training with different optimization conditions (eg. learning rate, data scale and mean, gradient update scheme e.t.c.) can lead to drastically different outcomes. However, one of the very things that make training NNs difficult is well designed optimization strategies without which they do not learn useful features. Our analysis is based on certain assumptions on data distribution and conditions on weight matrices. Thus in order to empirically verify our analysis, we use the following experimental protocols that make the optimization well conditioned.

For all experiments, we use mini-batch stochastic gradient descent with momentum (0.9) for optimization, 50 epochs, batch size 50 and hidden units 1000. We train DAE, CAE, mDAE and SAE (using eq. 13) with the same hyper-parameters for all the experiments. For regularization coefficient (σ^2), we use the values in the set $\{0, 0.001, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.6^2, 0.7^2, 0.8^2, 0.9^2, 1.0\}$ for all models except DAE where σ^2 values represent the *variance* of Gaussian noise added. For all models and activation functions, we use squared loss and linear decoding. We initialize the bias to zeros and use normalized initialization (Glorot & Bengio, 2010) for the weights. Further, we subtract mean and divide by standard deviation for all samples. [2]

Learning Rate (LR): Too small a LR won't move the weights from their initialized region and the convergence would be very slow. On the other hand, if we use too large a learning rate, it will change weight direction very drastically (may diverge), something we don't desire for our predictions to hold. So, we find a middle ground and choose LR in the range $(0.001, 0.005)$ for our experiments.

Terminology: We are interested in analysing the sparsity of hidden units as a function of regularization coefficient σ^2 through out our experiments. Recall that our notion of sparsity 2.1 is denoted by the fraction of data samples that deactivate a hidden unit instead of the fraction of hidden units that deactivate for a given data sample. This choice was made in order to treat each hidden unit as a random variable. Since we cannot identify a particular hidden unit across auto-encoders trained with different values of σ^2, the only way for measuring the level of sparsity in auto-encoder units is compute the *Average Activation Fraction*, which is defined as follows:

$$Avg.Act.Fraction = \frac{\sum_{i=1}^{N}\sum_{j=1}^{m}\mathbf{1}(h_j^i > \delta_{\min})}{N \times m} \quad (14)$$

Here $\mathbf{1}(.)$ is the indicator operator, h_j^i denotes the j^{th} hidden unit for the i^{th} data sample, and $\delta_{\min}$ is the activation threshold. In the case ReLU, $\delta_{\min} = 0$, and in the case of Sigmoid and Softplus, $\delta_{\min} = 0.1$. Also N and m denote the total number of data samples and number of hidden units respectively. Notice sparsity of a hidden unit is inversely related to the average activation fraction for a single unit. Thus our definition of *Avg. Activation Fraction* is the indicator of average sparsity across all hidden units. Finally, while measuring Avg. Activation Fraction during training, we also keep track of fraction of *dead units*. Dead units are those hidden units which deactivate for all data samples and are thus unused by the network for data reconstruction. Notice while achieving sparsity, it is desired that minimal hidden units are dead and all *alive* units activate only for a small fraction of data samples.

3.1. Sparsity when Bias Gradient is zero

One of the main predictions made based on theorem 1 is that the sparsity of hidden units should remain unchanged with respect to σ^2 when the bias gradient $\frac{\partial \mathcal{R}}{\partial b_{e_j}} = 0$ and weight lengths are fixed to a pre-determined value because the expected pre-activation becomes completely independent of σ^2. Notice this prediction only accounts for change in sparsity as a result of change in expected pre-activation of the corresponding unit. Sparsity can also increase when expected pre-activation for that unit is fixed, as a result of change in weight directions such that majority samples take pre-activation values below activation threshold while the minority takes values above it such that the overall expected value remains unchanged. This change in weight directions is also affected by σ^2 since regularization functions specified in corollary 2 and 1 contain both weight and bias terms. However, the latter factor contributing to change in sparsity is unpredictable in terms of changing σ^2 values. Hence it is desired for sparsity to be largely affected only when bias gradient is present for better predictive power.

[2] We noticed in case of MNIST, it is important to add a large number (0.1) to the standard deviation before dividing. We believe this is because MNIST (being binary images with uniform background) does not follow our assumption on data distribution.

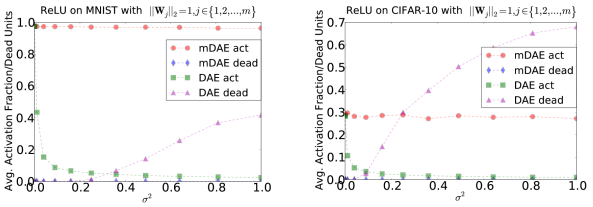

Figure 1. Trend of average activation fraction *vs.* σ^2 with weight length constraint using ReLU on MNIST (left) and CIFAR-10 (right).

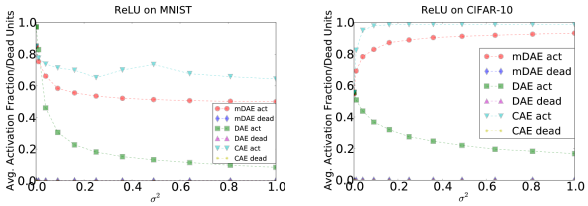

Figure 2. Trend of average activation fraction *vs.* σ^2 without weight length constraint using ReLU on MNIST (left) and CIFAR-10 (right).

Hence we analyse the effect of regularization coefficient (σ^2) on the sparsity of representations learned by AE models using ReLU activation function with weight lengths constrained to be one. Notice ReLU has zero bias gradient for CAE and mDAE, but also for the equivalent regularization derived for DAE 3. The plots are shown in figure 1.[3]

We see that the effect of bias gradient largely dominates the behaviour of hidden units in terms of sparsity. Specifically, as predicted, average activation fraction (and thus sparsity) remains unchanged with respect to regularization coefficient σ^2 when ReLU is applied to CAE and mDAE due to the absence of bias gradient.

We also analyse the effect of regularization coefficient (σ^2) on the sparsity of representations learned by AE models using ReLU activation functions when weight lengths are not constrained. These plots can be seen in fig 2. We find that the trend becomes unpredictable for both CAE and mDAE (both datasets have different trends). As discussed after theorem 1, without weight length constraint, σ^2 affects weight length which in turn affects $\frac{\partial \mathcal{J}_{AE}}{\partial b_{e_j}}$ that changes the value of expected pre-activation. However, this effect is unpredictable and thus undesired.

On the other hand, we see that for DAE, in the constrained length case (fig 1), the number of dead units start rising only after the average activation fraction reaches around 0.05. However, in case of unconstrained weight length, ReLU does not go below the avg. activation fraction of 0.1. This shows that constrained weight length achieves higher level of sparsity before giving rise to dead units.

In summary, we find that bias gradient dominates the behaviour of hidden units in terms of sparsity. Also, these experiments suggest we get both more predictive power and better sparsity with hidden weights constrained to have fixed (unit) length. Notice this does not restrict the usefulness of the representation leaned by auto-encoders since we are only interested in the filter shapes, and not their scale.

[3] For weight length constrained to 1, CAE and mDAE objectives become equivalent.

3.1.1. Why is DAE affected by σ^2 when ReLU has zero bias gradient?

The surprising part of the above experiments is that DAE has a stable decreasing sparsity trend (across different values of σ^2) for ReLU although DAE (similar to CAE, mDAE) has a regularization form given in corollary 2. The fact that ReLU practically does not generate bias gradients from this form of regularization brings our attention to an interesting possibility: ReLU is generating the positive bias gradient due to the first order regularization term in DAE. Recall that we marginalize out the first order term in DAE (during Taylor's expansion, see proof of theorem 3) while taking expectation over all corrupted versions of a training sample. However, the mathematically equivalent objective of DAE obtained by this analytical marginalization is not what we optimize in practice. While optimizing with explicit corruption in a batch-wise manner, we indeed get a non-zero first order term, which does not vanish due to finite sampling (of corrupted versions); thus explaining sparsity for ReLU. We test this hypothesis by optimizing the explicit Taylor's expansion of DAE (eDAE) with only the first order term on MNIST and CIFAR-10 using our standard experimental protocols:

$$\mathcal{J}_{eDAE} = \mathbb{E}_{\mathbf{x}}[\ell(\mathbf{x}, f_d(f_e(\mathbf{x}))) + (\tilde{\mathbf{x}} - \mathbf{x})^T \nabla_{\tilde{\mathbf{x}}} \ell]$$

where $\tilde{\mathbf{x}}$ is a Gaussian corrupted version of $\mathbf{x}$. The activation fraction *vs.* corruption variance (σ^2) for eDAE is shown in figure 3 which confirms that the first order term contributes towards sparsity. On a more general note, lower order terms (in Taylor's expansion) of highly non-linear functions generally change slower (hence less sensitive) compared to higher order terms. In conclusion we find that explicit corruption may have advantages at times compared to marginalization because it captures the effect of both lower and higher order terms together.

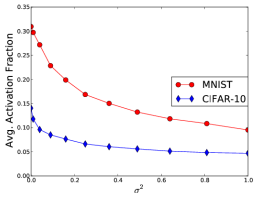

Figure 3. Activation fraction *vs.* σ^2 for eDAE.

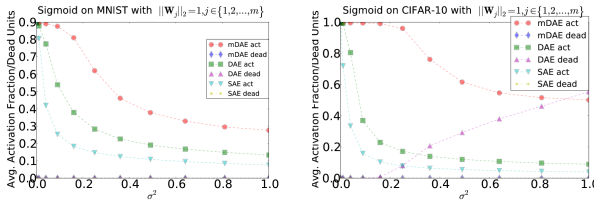

Figure 4. Trend of average activation fraction *vs.* σ^2 with weight length constraint using Sigmoid activation MNIST (left) and CIFAR-10 (right).

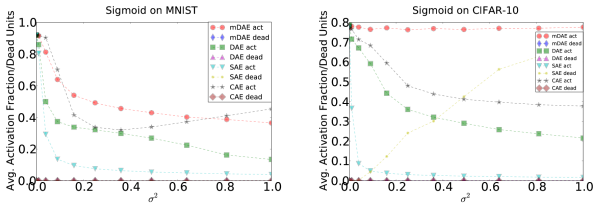

Figure 5. Trend of average activation fraction *vs.* σ^2 without weight length constraint using Sigmoid activation MNIST (left) and CIFAR-10 (right).

3.2. Sparsity when Bias Gradient is positive

As predicted by theorem 1, if the bias gradient is strictly positive ($\frac{\partial \mathcal{R}}{\partial b_{e_j}} > 0$), then increasing the value of σ^2 should lead to smaller expected pre-activation and thus increasing sparsity. This is specially true when the weight lengths are fixed to some length. This is because term $\frac{\partial \mathcal{R}}{\partial b_{e_j}}$ may depend on weight length (depending on the regularization) which is also affected by σ^2. However, since this effect is hard to predict, sparsity may not always be proportional to σ^2 for un-constrained weight length.

In order to verify these intuitions, we first analyse the effect of regularization coefficient (σ^2) on the sparsity of representations learned by AE models using Sigmoid[4][5] activation function with weight lengths constrained to one. The plots are shown in figure 4. These plots show a stable increasing sparsity trend with increasing regularization coefficient as predicted by our analysis.

Finally, we now analyse the effect of regularization coefficient (σ^2) on the sparsity of representations learned by AE models using Sigmoid activation function when weight lengths are unconstrained. These plots are shown in figure 5. As mentioned above, unconstrained weight length leads to unpredictable behaviour of sparsity with respect to regularization coefficient. This can be seen for mDAE and CAE for both datasets (different trends).

In summary, we again find that weight lengths constrained to have some fixed value lead to better predictive power in terms of sparsity. However in either case, the empirical observations substantiate our claim that sparsity in auto-encoders is dominated by the effect of bias gradient from regularization instead of weight direction. This explains why existing regularized auto-encoders learn sparse representation and the effect of regularization coefficient on sparsity.

[4]Due to lack of space and because Softplus had trends similar to Sigmoid, we don't show its plots.

[5]Although Sigmoid only guarantees sparsity for regularizations in corollary 1 (eg. SAE), we find it behaves similarly for corollary 2(eg. mDAE, CAE).

4. Conclusion and Discussion

We establish a formal connection between features learned by regularized auto-encoders and sparse representation. Our contribution is multi-fold, we show: a) AE regularizations with positive encoding bias gradient encourage sparsity (theorem 1), while those with zero bias gradient are not affected by regularization coefficient; b) activation functions which are non-decreasing, with negative saturation at zero, encourage sparsity for such regularizations (theorem 2) and that multiple existing activations have this property (eg. ReLU, Softplus and Sigmoid); c) existing AEs have regularizations of the form suggested in corollary 1 and 2, which not only brings them under a unified framework, but also shows more general forms of regularizations that encourage sparsity.

On the empirical side, a) bias gradient dominates the effect on sparsity of hidden units; specifically sparsity is in general proportional to the regularization coefficient when bias gradient is positive and remains unaffected when it is zero (section 3); b) Constraining the weight vectors during optimization to have fixed length leads to better sparsity and behaviour as predicted by our analysis. Notice this does not restrict the usefulness of the representation leaned by auto-encoders since we are only interested in the filter shapes (weight direction), and not their scale. On the flip side, without length constraint, the behaviour of auto-encoders w.r.t. regularization coefficient becomes unpredictable in some cases. c) explicit corruption (*eg.* DAE) may have advantages over marginalizing it out (*eg.* mDAE, see section 3.1.1) because it captures both first and second order effects.

In conclusion, our analysis combined together unifies existing AEs and activation functions by bringing them under a unified framework, but also uncovers more general forms of regularizations and fundamental properties that encourage sparsity in hidden representation. Our analysis also yields new insights into AEs and provides novel tools for analysing existing (and new) regularization/activation functions that help predicting whether the resulting AE learns sparse representations.

References

Bengio, Yoshua, Mesnil, Grégoire, Dauphin, Yann, and Rifai, Salah. Better mixing via deep representations. In *ICML*, pp. 552–560, 2013.

Bourlard, H. and Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, 1988. ISSN 0340-1200.

Chen, Minmin, Weinberger, Kilian Q., Sha, Fei, and Bengio, Yoshua. Marginalized denoising auto-encoders for nonlinear representations. In *ICML*, pp. 1476–1484, 2014.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. Deep sparse rectifier neural networks. In *AISTATS*, pp. 315–323, 2011.

Goodfellow, Ian J., Warde-Farley, David, Mirza, Mehdi, Courville, Aaron C., and Bengio, Yoshua. Maxout networks. In *ICML*, pp. 1319–1327, 2013.

Hinton, Geoffrey, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

Hinton, Geoffrey E. Distributed representations. 1984.

Hubel, D. H. and Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148:574–591, October 1959.

Kavukcuoglu, Koray and Lecun, Yann. Fast inference in sparse coding algorithms with applications to object recognition. Technical report, Courant Institute, NYU, 2008.

Krizhevsky, Alex. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Lecun, Yann and Cortes, Corinna. The MNIST database of handwritten digits. URL http://yann.lecun.com/exdb/mnist/.

Lee, Honglak, Ekanadham, Chaitanya, and Ng, Andrew Y. Sparse deep belief net model for visual area v2. In *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

Li, Jun, Luo, Wei, Yang, Jian, and Yuan, Xiaotong. Why does the unsupervised pretraining encourage moderate-sparseness? *CoRR*, abs/1312.5813, 2013. URL http://arxiv.org/abs/1312.5813.

Memisevic, Roland, Konda, Kishore Reddy, and Krueger, David. Zero-bias autoencoders and the benefits of co-adapting features. In *ICLR*, 2014.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *ICML*, pp. 807–814, 2010.

Ng, Andrew. Sparse autoencoder. *CSE294 Lecture notes*, 2011.

Olshausen, Bruno A. and Fieldt, David J. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.

Patterson, Karalyn, Nestor, Peter, and Rogers, Timothy. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature Rev. Neuroscience*, 8 (12):976–987, 2007.

Rifai, Salah, Mesnil, Grégoire, Vincent, Pascal, Muller, Xavier, Bengio, Yoshua, Dauphin, Yann, and Glorot, Xavier. Higher order contractive auto-encoder. In *ECML/PKDD*, pp. 645–660, 2011a.

Rifai, Salah, Vincent, Pascal, Muller, Xavier, Glorot, Xavier, and Bengio, Yoshua. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, pp. 833–840, 2011b.

Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. Learning representations by back-propagating errors. *Nature*, pp. 533–536, 1986.

Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Extracting and composing robust features with denoising autoencoders. In *ICML*, pp. 1096–1103, 2008.

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., and Ma, Yi. Robust face recognition via sparse representation. *IEEEE TPAMI*, 31(2):210 –227, Feb. 2009.

Yang, Jianchao, Yu, Kai, Gong, Yihong, and Huang, Thomas. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pp. 1794–1801, 2009.