

# Convergence of gradient based pre-training in Denoising autoencoders

Vamsi K. Ithapu<sup>1</sup>, Sathya Ravi<sup>2</sup>, Vikas Singh<sup>3,1</sup>

<sup>1</sup>Computer Sciences   <sup>2</sup>Industrial and Systems   <sup>3</sup>Biostat. and Med. Informatics  
University of Wisconsin-Madison

## Abstract

The success of deep architectures is at least in part attributed to the layer-by-layer unsupervised pre-training that initializes the network. Various papers have reported extensive empirical analysis focusing on the design and implementation of good pre-training procedures. However, an understanding pertaining to the consistency of parameter estimates, the convergence of learning procedures and the sample size estimates is still unavailable in the literature. In this work, we study pre-training in classical and distributed denoising autoencoders with these goals in mind. We show that the gradient converges at the rate of  $\frac{1}{\sqrt{N}}$  and has a sub-linear dependence on the size of the autoencoder network. In a distributed setting where disjoint sections of the whole network are pre-trained synchronously, we show that the convergence improves by at least  $\tau^{3/4}$ , where  $\tau$  corresponds to the size of the sections. We provide a broad set of experiments to empirically evaluate the suggested behavior.

## 1 Introduction

In the last decade, deep learning models have provided state of the art results for a broad spectrum of problems in computer vision Krizhevsky et al. (2012); Taigman et al. (2014), natural language processing Socher et al. (2011a,b), machine learning Hamel & Eck (2010); Dahl et al. (2011) and biomedical imaging Plis et al. (2013). The underlying *deep* architecture with multiple layers of hidden variables allows for learning high-level representations which fall beyond the hypotheses space of (*shallow*) alternatives Bengio (2009). This representation-learning behavior is attractive in many applications where setting up a suitable feature engineering pipeline that captures the discriminative content of the data remains difficult, but is critical to the overall performance. Despite many desirable qualities, the richness afforded by multiple levels of variables and the non-convexity of the learning objectives makes training deep architectures challenging. An interesting solution to this problem proposed in Hinton & Salakhutdinov (2006); Bengio et al. (2007) is a hybrid two-stage procedure. The first step performs a layer-wise unsupervised learning, referred to as “pre-training”, which provides a suitable initialization of the parameters. With this warm start, the subsequent discriminative (supervised) step simply *fine-tunes* the network with an appropriate loss function. Such procedures broadly fall under two categories – restricted Boltzmann machines and autoencoders Bengio (2009). Extensive empirical evidence has demonstrated the benefits of this strategy, and the recent success of deep learning is at least partly attributed to pre-training Bengio (2009); Erhan et al. (2010); Coates et al. (2011).

Given this role of pre-training, there is significant interest in understanding precisely what the unsupervised phase does and why it works well. Several authors have provided interesting explanations to these questions. Bengio (2009) interprets pre-training as providing the downstream optimization with a suitable initialization. Erhan et al. (2009, 2010) presented compelling empirical evidence that pre-training serves as an “unusual form of regularization” which biases the parameter search by minimizing variance. The influence of the network structure (lengths of visible and hidden layers) and optimization methods on the pre-training estimates have been well studied Coates et al. (2011); Ngiam et al. (2011). Dahl et al. (2011) evaluate the role of pre-training for DBN-HMMs as a function of sample sizes and discuss the regimes which yield the maximum improvements in performance. A related but distinct set of results describe procedures that construct “meaningful” data representations. Denoising autoencoders Vincent et al. (2010) seek representations that are invariant to data corruption, while contractive autoencoders (CA) Rifai et al. (2011b) seek robustness to data variations. The manifold tangent classifier Rifai et al. (2011a) searches for low dimensional non-linear sub-manifold that approximates the input distribution. Other works have shown that with a suitable architecture, even a random initialization seems to give impressive performance Saxe et al. (2011). Very recently, Livni et al. (2014);

Bianchini & Scarselli (2014) have analyzed the complexity of multi-layer neural networks, theoretically justifying that certain types of deep networks learn complex concepts. While the significance of the results above cannot be overemphasized, our current understanding of the conditions under which pre-training is *guaranteed* to work well is still not very mature. Our goal here is to complement the above body of work by deriving specific conditions under which this pre-training procedure will have convergence guarantees.

To keep the presentation simple, we restrict our attention to a widely used form of pre-training — Denoising auto-encoder — as a sandbox to develop our main ideas, while noting that a similar style of analysis is possible for other (unsupervised) formulations also. Denoising auto-encoders (DA) seek robustness to partial destruction (or corruption) of the inputs, implying that a *good* higher level representation must characterize only the ‘stable’ dependencies among the data dimensions (features) and remain invariant to small variations Vincent et al. (2010). Since the downstream layers correspond to increasingly non-linear compositions, the layer-wise unsupervised pre-training with DAs gives increasingly abstract representations of the data as the depth (number of layers) increases. These non-linear transformations (e.g., sigmoid functions) make the objective non-convex, and so DAs are typically optimized via a stochastic gradients. Recently, large scale architectures have also been successfully trained in a massively distributed setting where the stochastic descent is performed asynchronously over a cluster Dean et al. (2012). The empirical evidence regarding the performance of this scheme is compelling. The analysis in this paper is an attempt to understand this behavior on the theoretical side (for both classical and distributed DA), and identify situations where such constructions *will* work well with certain guarantees.

We summarize the main **contributions** of this paper. We first derive *convergence results and the associated sample size estimates* of pre-training a single layer DA using the randomized stochastic gradients Ghadimi & Lan (2013). We show that the convergence of expected gradients is  $\mathcal{O}\left(\frac{(d_h d_v)^{3/4}}{\sqrt{N}}\right)$  and the number of calls (to a first order oracle) is  $\mathcal{O}\left(\frac{(d_h d_v)^{3/2}}{\epsilon^2}\right)$ , where  $d_h$  and  $d_v$  correspond to the number of hidden and visible layers,  $N$  is the number of iterations, and  $\epsilon$  is an error parameter. We then show that the DA objective can be distributed and present improved rates while learning small fractions of the network synchronously. These bounds provide a nice relationship between the sample size, asymptotic convergence of gradient norm (to zero) and the number of hidden/visible units. Our results extend easily to stacked and convolutional denoising auto-encoders. Finally, we provide sets of experiments to evaluate if the results are meaningful in practice.

## 2 Preliminaries

Autoencoders are single layer neural networks that learn over-complete representations by applying nonlinear transformations on the input data Vincent et al. (2010); Bengio (2009). Given an input  $\mathbf{x}$ , an autoencoder identifies representations of the form  $\mathbf{h} = \sigma(\mathbf{W}\mathbf{x})$ , where  $\mathbf{W}$  is a  $d_h \times d_v$  transformation matrix and  $\sigma$  denotes point-wise sigmoid nonlinearity. Here,  $d_v$  and  $d_h$  denote the lengths of visible and hidden layers respectively. Various types of autoencoders are possible depending on the assumptions that generate the  $\mathbf{h}$ ’s — robustness to data variations/corruptions, enforcing data to lie on some low-dimensional sub-manifolds etc. Rifai et al. (2011b,a).

Denoising autoencoders are widely used class of autoencoders Vincent et al. (2010), that learn higher-level representations by leveraging the inherent correlations/dependencies among input dimensions ( $j = 1, \dots, d_v$ ), thereby ensuring that  $\mathbf{h}$  is robust to changes in less informative input/visible units. This is based on the hypothesis that abstract high-level representations should only encode stable data dependencies across input dimensions, and be robust to spurious correlations and invariant features. This is done by ‘corrupting’ each individual visible dimension randomly, and using the corrupted version ( $\tilde{\mathbf{x}}$ ’s) instead, to learn  $\mathbf{h}$ ’s. The corruption generally corresponds to ignoring (setting to 0) the input signal with some probability (denoted by  $\zeta$ ), although other types of additive/multiplicative corruption may also be used. If  $\mathbf{x}_j$  is the input at the  $j^{th}$  unit, then the corrupted signal is  $\tilde{\mathbf{x}}_j = \mathbf{x}_j$  with probability  $1 - \zeta$  and 0 otherwise where  $j = 1, \dots, d_v$ . Note that each of the  $d_v$  dimensions are corrupted independently with the same probability  $\zeta$ . DA pre-training then corresponds to estimating the transformation  $\mathbf{W}$  by minimizing the following objective Bengio (2009),

$$\min_{\mathbf{W}} \mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}})} \|\mathbf{x} - \sigma(\mathbf{W}^T \sigma(\mathbf{W}\tilde{\mathbf{x}}))\|^2 \quad (1)$$

where the expectation is over the joint probability  $p(\mathbf{x}, \tilde{\mathbf{x}})$  of sampling an input  $\mathbf{x} \sim \mathcal{D}$  and generating the corresponding  $\tilde{\mathbf{x}}|\mathbf{x}$  using  $\zeta$ . The bias term (which is never corrupted) is taken care of by appending inputs  $\mathbf{x}$  with 1.

For notational simplicity, let us denote the process of generating  $\{\mathbf{x}, \tilde{\mathbf{x}}\}$  by a random variable  $\eta$ , i.e., one sample of  $\eta$  corresponds to a pair  $\{\mathbf{x}, \tilde{\mathbf{x}}\}$  where  $\tilde{\mathbf{x}}$  is constructed by randomly corrupting each of the  $d_v$  dimensions of  $\mathbf{x}$

with some probability  $\zeta$ . Then, if the reconstruction loss is  $\mathcal{L}(\eta; \mathbf{W}) := \mathcal{L}(\tilde{\mathbf{x}}, \mathbf{x}; \mathbf{W}) = \|\mathbf{x} - \sigma(\mathbf{W}^T \sigma(\mathbf{W} \tilde{\mathbf{x}}))\|^2$ , the objective in (40) becomes

$$\min_{\mathbf{W}} f(\mathbf{W}) := \mathbb{E}_{\eta} \mathcal{L}(\eta; \mathbf{W}) \quad (2)$$

Observe that the loss  $\mathcal{L}(\eta; \mathbf{W})$  and the objective in (2) constitutes an expectation over the randomly corrupted sample pairs  $\eta := \{\mathbf{x}, \tilde{\mathbf{x}}\}$ , which is non-convex. Analyzing convergence properties of such an objective using classical techniques, especially in a (distributed) stochastic gradient setup, is difficult. Therefore, given that the loss function is a composition of sigmoids, one possibility is to adopt convex variational relaxations of sigmoids in (2) and then apply standard convex analysis. But non-convexity is, in fact, the most interesting aspect of deep architectures, and so the analysis of a loose convex relaxation will be unable to explain the empirical success of DAs, and deep learning in general.

*High Level Idea.* The starting point of our analysis is a very recent result on stochastic gradients which only makes a weaker assumption of Lipschitz differentiability of the objective (rather than convexity). We assume that the optimization of (2) proceeds by querying a stochastic first order oracle ( $\mathcal{SFO}$ ), which provides noisy gradients of the objective function. For instance, the  $\mathcal{SFO}$  may simply compute a noisy gradient with a single sample  $\eta^k := \{\mathbf{x}_s, \tilde{\mathbf{x}}_s\}$  at the  $k^{th}$  iteration and use that alone to evaluate  $\nabla_{\mathbf{W}} \mathcal{L}(\eta^k; \mathbf{W}^k)$ . The main idea adapted from Ghadimi & Lan (2013) to our problem is to express the stopping criterion for the gradient updates by a probability distribution  $\mathbb{P}_R(\cdot)$  over iterations  $k$ , i.e., the stopping iteration is  $k \sim \mathbb{P}_R(\cdot)$  (and hence the name randomized stochastic gradients, RSG). Observe that this is the only difference from classical stochastic gradients used in pre-training, where the stopping criterion is assumed to be the last iteration. RSG will offer more useful theoretical properties, and is a negligible practical change to existing implementations. This then allows us to compute the expectation of the gradient norm, where the expectation is over stopping iterations sampled according to  $\mathbb{P}_R(\cdot)$ . For our case, the updates are given by,

$$\mathbf{W}^{k+1} \leftarrow \mathbf{W}^k - \gamma^k G(\eta^k; \mathbf{W}^k) \quad (3)$$

where,  $G(\eta^k; \mathbf{W}^k) = \nabla_{\mathbf{W}} \mathcal{L}(\eta^k; \mathbf{W}^k)$  is the noisy gradient computed at  $k^{th}$  iteration ( $\gamma^k$  is the stepsize). We have flexibility in specifying the distribution of stopping criterion  $\mathbb{P}_R(\cdot)$ . It can be fixed a priori or selected by a hyper-training procedure that chooses the best  $\mathbb{P}_R(\cdot)$  (based on an accuracy measure) from a pool of distributions  $\mathcal{P}_R$ . With these basic tools in hand, we first compute the expectation of gradients where the expectation accounts for both the stopping criterion  $k \sim \mathbb{P}_R(\cdot)$  and  $\eta := \{\mathbf{x}, \tilde{\mathbf{x}}\}$ . We show that if the stepsizes  $\gamma^k$  in (3) are chosen carefully, the expected gradients decrease monotonically and converge. Based on this analysis, we derive the rate of convergence and corresponding sample size estimates for DA pre-training. We describe the one-layer DA (i.e., with one hidden layer) in detail, and all our results extend easily to the stacked and convolutional settings since the pre-training is done layer-wise in multi-layer architectures.

### 3 Denoising Autoencoders (DA) pre-training

We first present some results on the continuity and boundedness of the objective  $f(\mathbf{W})$  in (2), followed by the convergence rates for the optimization. Denote the element in  $i^{th}$  row and  $j^{th}$  column of  $\mathbf{W}$  by  $\mathbf{W}_{ij}$  where  $i = 1, \dots, d_v$  and  $j = 1, \dots, d_h$ . We require the following Lipschitz continuity assumptions on  $\mathcal{L}(\eta; \mathbf{W}_{ij})$  and the gradient  $\nabla_{\mathbf{W}_{ij}} f(\mathbf{W}_{ij})$ , which are fairly common in numerical optimization.  $L$  and  $L'$  are Lipschitz constants.

**Assumption (A1).**

$$\|\mathcal{L}(\eta; \mathbf{W}_{ij}) - \mathcal{L}(\eta; \hat{\mathbf{W}}_{ij})\| \leq L \|\mathbf{W}_{ij} - \hat{\mathbf{W}}_{ij}\| \quad \forall i, j$$

**Assumption (A2).**

$$\|\nabla_{\mathbf{W}_{ij}} f(\mathbf{W}_{ij}) - \nabla_{\mathbf{W}_{ij}} f(\hat{\mathbf{W}}_{ij})\| \leq L' \|\mathbf{W}_{ij} - \hat{\mathbf{W}}_{ij}\| \quad \forall i, j,$$

We see from (40) that  $\mathcal{L}(\eta; \mathbf{W}_{ij})$  is symmetric in  $\mathbf{W}_{ij}, \forall i, j$ . Depending on where  $\mathbf{W}_{ij}$  is located in the parameter space (and the variance of each data dimension  $j$ ), each  $\mathcal{L}(\eta; \mathbf{W}_{ij})$  corresponds to some  $L_{ij}$ , and  $L$  will then be the maximum of all such  $L_{ij}$ 's (similarly for  $L'$ ).

Based on the definition of  $G(\eta^k; \mathbf{W}^k)$  and (2), we see that the noisy gradients  $G(\eta^k; \mathbf{W}^k)$  are unbiased estimates of the true gradient since  $\nabla_{\mathbf{W}} f(\mathbf{W}^k) = \mathbb{E}_{\eta^k} G(\eta^k; \mathbf{W}^k)$ . To compute the expectation of the gradients,  $\nabla_{\mathbf{W}} f(\mathbf{W}^k)$ , over the distribution governing whether the process stops at iteration  $k$ , i.e.,  $R \sim \mathbb{P}_R(\cdot)$ , we first state a result regarding the variance of the noisy gradients and the Lipschitz constant of  $\nabla_{\mathbf{W}} f(\mathbf{W}^k)$ . All proofs are included in the supplement.

**Lemma 3.1 (Variance bound and Lipschitz constant).** Using A1, A2 and  $\nabla_{\mathbf{W}} f(\mathbf{W}^k) = \mathbb{E}_{\eta^k} G(\eta^k; \mathbf{W}^k)$ , we have

$$\begin{aligned} \text{Var}(G(\eta^k; \mathbf{W}^k)) &\leq d_h d_v L^2 \\ \|\nabla_{\mathbf{W}} f(\mathbf{W}) - \nabla_{\mathbf{W}} f(\hat{\mathbf{W}})\| &\leq \sqrt{d_h d_v} L' \|\mathbf{W} - \hat{\mathbf{W}}\| \end{aligned} \quad (4)$$

*Proof.* Recall that the assumptions [A1] and [A2] are,

$$\begin{aligned} [\text{A1}] \quad &\|\mathcal{L}(\eta; \mathbf{W}_{ij}) - \mathcal{L}(\eta; \hat{\mathbf{W}}_{ij})\| \leq L \|\mathbf{W}_{ij} - \hat{\mathbf{W}}_{ij}\| \quad \forall i, j \\ [\text{A2}] \quad &\|\nabla_{\mathbf{W}_{ij}} f(\mathbf{W}_{ij}) - \nabla_{\mathbf{W}_{ij}} f(\hat{\mathbf{W}}_{ij})\| \leq L' \|\mathbf{W}_{ij} - \hat{\mathbf{W}}_{ij}\| \quad \forall i, j \end{aligned}$$

The noisy gradient is defined as  $G(\eta^k; \mathbf{W}^k) = \nabla_{\mathbf{W}} \mathcal{L}(\eta^k; \mathbf{W}^k)$ . Using the mean value theorem and [A1], we have  $|G(\eta^k; \mathbf{W}_{ij}^k)| \leq L$ . This implies that the maximum variance of  $G(\eta^k; \mathbf{W}_{ij}^k)$  is  $L^2$ . We can then obtain the following upper bound on the variance of  $G(\eta^k; \mathbf{W}^k)$ ,

$$\begin{aligned} \mathbb{E}_{\eta^k} (\|G(\eta^k; \mathbf{W}^k) - \nabla_{\mathbf{W}} f(\mathbf{W}^k)\|^2) &= \mathbb{E}_{\eta^k} \left( \sum_{ij} (G(\eta^k; \mathbf{W}_{ij}^k) - \nabla_{\mathbf{W}_{ij}} f(\mathbf{W}_{ij}^k))^2 \right) \\ &= \sum_{ij} \text{Var}(G(\eta^k; \mathbf{W}_{ij}^k)) \leq d_h d_v L^2 \end{aligned} \quad (5)$$

Using [A2], we have

$$\begin{aligned} \|\nabla_{\mathbf{W}} f(\mathbf{W}) - \nabla_{\mathbf{W}} f(\hat{\mathbf{W}})\|^2 &= \sum_{i,j} \|\nabla_{\mathbf{W}_{ij}} f(\mathbf{W}_{ij}) - \nabla_{\mathbf{W}_{ij}} f(\hat{\mathbf{W}}_{ij})\|^2 \\ &\leq \sum_{i,j} (L'_{ij})^2 \|\mathbf{W}_{ij} - \hat{\mathbf{W}}_{ij}\|^2 \leq \sum_{i,j} (L'_{ij})^2 \sum_{i',j'} \|\mathbf{W}_{i'j'} - \hat{\mathbf{W}}_{i'j'}\|^2 \\ &\leq d_h d_v L'^2 \|\mathbf{W} - \hat{\mathbf{W}}\|^2 \end{aligned} \quad (6)$$

where the equality follows from the definition of  $\ell_2$ -norm. The second inequality is from [A2]. The last two inequalities use the definition of  $\ell_2$ -norm and that  $L'$  is the maximum of all  $L'_{ij}$ s.  $\square$

Whenever the inputs  $\mathbf{x}$  are bounded between 0 and 1,  $f(\mathbf{W})$  is finite-valued everywhere and there exists a minimum due to the bounded range of sigmoid in (40). Also,  $f(\cdot)$  is analytic with respect to  $\mathbf{W}_{ij} \forall i, j$ . Now, if one adopts the RSG scheme for the optimization, using Lemma 3.1, we have the following upper bound on the expected gradients for the one-layer DA pre-training in (2).

**Lemma 3.2 (Expected gradients of one-layer DA).** Let  $N \geq 1$  be the maximum number of RSG iterations with step sizes  $\gamma^k < \frac{2}{L' \sqrt{d_h d_v}}$ . Let  $\mathbb{P}_R(\cdot)$  be given as

$$\begin{aligned} \mathbb{P}_R(k) &:= \text{Pr}(R = k) \\ &= \frac{2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2}{\sum_{k=1}^N (2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2)} \end{aligned} \quad (7)$$

where  $k = 1, \dots, N$ . If  $D_f = 2(f(\mathbf{W}^1) - f^*)$ , we have

$$\begin{aligned} \mathbb{E} (\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) &\leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2}{\sum_{k=1}^N (2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2)} \end{aligned} \quad (8)$$

*Proof.* Broadly, this proof emulates the proof of Theorem 2.1 in Ghadimi & Lan (2013) with several adjustments. The Lipschitz continuity assumptions (refer to [A1] and [A2]) give the following bounds on the variance of  $G(\eta^k; \mathbf{W}^k)$  and the Lipschitz continuity of  $\nabla_{\mathbf{W}} f(\mathbf{W})$  (refer to Lemma 3.1),

$$\begin{aligned} \text{Var}(G(\eta^k; \mathbf{W}^k)) &\leq d_h d_v L^2 \\ \|\nabla_{\mathbf{W}} f(\mathbf{W}) - \nabla_{\mathbf{W}} f(\hat{\mathbf{W}})\| &\leq \sqrt{d_h d_v} L' \|\mathbf{W} - \hat{\mathbf{W}}\| \end{aligned} \quad (9)$$

Using the properties of Lipschitz continuity we have,

$$f(\mathbf{W}^{k+1}) \leq f(\mathbf{W}^k) + \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \mathbf{W}^{k+1} - \mathbf{W}^k \rangle + \frac{\sqrt{d_h d_v} L'}{2} \|\mathbf{W}^{k+1} - \mathbf{W}^k\|^2$$

Since the update of  $\mathbf{W}^k$  using the noisy gradient is  $\mathbf{W}^{k+1} \leftarrow \mathbf{W}^k - \gamma^k G(\eta^k; \mathbf{W}^k)$ , where  $\gamma^k$  is the step-size, we then have,

$$f(\mathbf{W}^{k+1}) \leq f(\mathbf{W}^k) - \gamma^k \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), G(\eta^k; \mathbf{W}^k) \rangle + \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \|G(\eta^k; \mathbf{W}^k)\|^2$$

By denoting  $\delta^k := G(\eta^k; \mathbf{W}^k) - \nabla_{\mathbf{W}} f(\mathbf{W}^k)$ ,

$$\begin{aligned} f(\mathbf{W}^{k+1}) &\leq f(\mathbf{W}^k) - \gamma^k \|\nabla_{\mathbf{W}} f(\mathbf{W}^k)\|^2 - \gamma^k \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle \\ &\quad + \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \left( \|\nabla_{\mathbf{W}} f(\mathbf{W}^k)\|^2 + 2 \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle + \|\delta^k\|^2 \right) \end{aligned}$$

Rearranging terms on the right hand side above,

$$\begin{aligned} f(\mathbf{W}^{k+1}) &\leq f(\mathbf{W}^k) - \left( \gamma^k - \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \right) \|\nabla_{\mathbf{W}} f(\mathbf{W}^k)\|^2 \\ &\quad - \left( \gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle + \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \|\delta^k\|^2 \end{aligned}$$

Summing the above inequality for  $k = 1, \dots, N$ ,

$$\begin{aligned} \sum_{k=1}^N \left( \gamma^k - \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \right) \|f(\mathbf{W}^k)\|^2 &\leq f(\mathbf{W}^1) - f(\mathbf{W}^{N+1}) \\ &\quad - \sum_{k=1}^N \left( \gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle + \frac{\sqrt{d_h d_v} L'}{2} \sum_{k=1}^N (\gamma^k)^2 \|\delta^k\|^2 \end{aligned} \tag{10}$$

where  $\mathbf{W}^0$  is the initial estimate. Using  $f^* \leq f(\mathbf{W}^{N+1})$ , we have,

$$\begin{aligned} \sum_{k=1}^N \left( \gamma^k - \frac{\sqrt{d_h d_v} L'}{2} (\gamma^k)^2 \right) \|f(\mathbf{W}^k)\|^2 &\leq f(\mathbf{W}^1) - f^* \\ &\quad - \sum_{k=1}^N \left( \gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle + \frac{\sqrt{d_h d_v} L'}{2} \sum_{k=1}^N (\gamma^k)^2 \|\delta^k\|^2 \end{aligned} \tag{11}$$

We now take the expectation of the above inequality over all the random variables in the RSG updating process – which include the randomization  $\eta$  used for constructing noisy gradients, and the stopping iteration  $R \sim \mathbb{P}_R(\cdot)$ . First, note that the stopping criterion is chosen at random with some given probability  $\mathbb{P}_R(\cdot)$  and is independent of  $\eta$ . Second, recall that the random process  $\eta$  is such that the random variable  $\eta^k$  is independent of  $\eta^{k+1}$  for some iteration number  $k$ , because  $\mathcal{SFO}$  selects then randomly. However, the update point  $\mathbf{W}^{k+1}$  depends on  $G(\eta^k; \mathbf{W}^k)$  (which are functions of the random variables  $\eta^k$ ) from the first to the  $k^{th}$  iteration. That is,  $\mathbf{W}^{k+1}$  is not independent of  $\mathbf{W}^k$ , and in fact the updates  $\mathbf{W}^k$  form a Markov process. So, we can take the expectation with respect to the joint probability  $p(\eta^{[N]}, R) = p(\eta^{[N]})p(R)$  where  $\eta^{[N]}$  denotes the random process from  $\eta^1$  until  $\eta^N$ . We analyze each of the last two terms on the right hand side of (11) by first taking expectation with respect to  $\eta^{[N]}$ . The second last term becomes,

$$\begin{aligned} \mathbb{E}_{\eta^{[N]}} \left[ \sum_{k=1}^N \left( \gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle \right] \\ = \sum_{k=1}^N \left( \gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \mathbb{E}_{\eta^{[k]}} (\langle \nabla_{\mathbf{W}} f(\mathbf{W}^k), \delta^k \rangle | \eta^1, \dots, \eta^k) = 0 \end{aligned} \tag{12}$$

where the last equality follows from the definition of  $\delta^k = G(\eta^k; \mathbf{W}^k) - \nabla_{\mathbf{W}} f(\mathbf{W}^k)$  and  $\mathbb{E}_{\eta^k}(G(\eta^k; \mathbf{W}^k)) = \nabla_{\mathbf{W}} f(\mathbf{W}^k)$ . Further, from Equation 9 we have  $\mathbb{E}_{\eta^k} \|\delta^k\|^2 = \text{Var}(G(\eta^k; \mathbf{W}^k)) \leq d_h d_v L^2$ . So, the expectation of the last term in (11) becomes,

$$\mathbb{E}_{\eta^{[N]}} \left[ \frac{\sqrt{d_h d_v} L'}{2} \sum_{k=1}^N (\gamma^k)^2 \|\delta^k\|^2 \right] = \frac{\sqrt{d_h d_v} L'}{2} \sum_{k=1}^N (\gamma^k)^2 \mathbb{E}_{\eta^{[N]}} (\|\delta^k\|^2) \leq \frac{(\sqrt{d_h d_v})^3 L' L^2}{2} \sum_{k=1}^N (\gamma^k)^2 \quad (13)$$

Using (12) and (13) and the inequality in (11) we have,

$$\sum_{k=1}^N \left( 2\gamma^k - \sqrt{d_h d_v} L' (\gamma^k)^2 \right) \mathbb{E}_{\eta^{[N]}} \|f(\mathbf{W}^k)\|^2 \leq 2(f(\mathbf{W}^1) - f^*) + (\sqrt{d_h d_v})^3 L' L^2 \sum_{k=1}^N (\gamma^k)^2 \quad (14)$$

Using the definition of  $\mathbb{P}_R(k)$  from Equation 7 and denoting  $D_f = 2(f(\mathbf{W}^1) - f^*)$ , we finally obtain

$$\begin{aligned} \mathbb{E}_{R, \eta^{[N]}} (\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) &= \sum_{k=1}^N \frac{(2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2) \mathbb{E}_{\eta^{[N]}} (\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2)}{\sum_{k=1}^N (2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2)} \\ &\leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2}{\sum_{k=1}^N (2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2)} \end{aligned} \quad (15)$$

□

The expectation in (8) is over  $\eta$  and  $R \sim \mathbb{P}_R(\cdot)$ . Here,  $\gamma^k < \frac{2}{L' \sqrt{d_h d_v}}$  ensures that the summations in the denominators of  $\mathbb{P}_R(\cdot)$  in (7) and the bound in (8) are positive.  $D_f$  represents a quantity which is twice the deviation of the objective  $f(\mathbf{W})$  at the RSG starting point ( $\mathbf{W}^1$ ) from the optimum. Observe that the bound in (8) is a function of  $D_f$  and network parameters, and we will analyze it shortly.

As stated, there are a few caveats that are useful to point out. Since no convexity assumptions are imposed on the loss function, Lemma 3.2 on its own offers no guarantee that the function values decrease as  $N$  increases. In particular, in the worst case, the bound may be loose. For instance, when  $D_f \approx 0$  (i.e., the initial point is already a good estimate of the stationary point), the upper bound in (8) is non-zero. Further, the bound contains summations involving the stepsizes, both in the numerator and denominator, indicating that the limiting behavior may be sensitive to the choices of  $\gamma^k$ . The following result gives a remedy — by choosing  $\gamma^k$  to be small enough, the upper bound in (8) will decrease monotonically as  $N$  increases.

**Lemma 3.3 (Monotonicity and convergence of expected gradients).** *By choosing  $\gamma^k$  such that*

$$\gamma^{k+1} \leq \gamma^k \text{ with } \gamma^1 < \frac{1}{L' \sqrt{d_h d_v}} \quad (16)$$

*the upper bound of expected gradients in (8) decreases monotonically. Further, if the sequence for  $\gamma^k$  satisfies*

$$\begin{aligned} \lim_{N \rightarrow \infty} \sum_{k=1}^N \gamma^k \rightarrow \infty, \quad \lim_{N \rightarrow \infty} \sum_{k=1}^N (\gamma^k)^2 < \infty \\ \text{then } \lim_{N \rightarrow \infty} \mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \rightarrow 0 \end{aligned} \quad (17)$$

*Proof.* We first show the monotonicity of the expected gradients followed by its limiting behavior. Observe that whenever  $\gamma^k < \frac{1}{L' \sqrt{d_h d_v}}$ , we have

$$\left( 2 - L' \sqrt{d_h d_v} \gamma^k \right) > 1 \quad \forall k$$

Then the upper bound in (8) reduces to

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2}{\sum_{k=1}^N \gamma^k} \quad (18)$$

To show that right hand side in the above inequality decreases as  $N$  increases, we need to show the following

$$\frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^{N+1} (\gamma^k)^2}{\sum_{k=1}^{N+1} \gamma^k} \leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2}{\sum_{k=1}^N \gamma^k} \quad (19)$$

By denoting the terms in the above inequality as follows,

$$\begin{aligned} a &= D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2 \\ b &= (\sqrt{d_h d_v})^3 L^2 L' (\gamma^{N+1})^2 \quad c = \sum_{k=1}^N \gamma^k \quad d = \gamma^{N+1} \end{aligned} \quad (20)$$

To show that the inequality in Equation 19 holds,

$$\begin{aligned} \frac{a+b}{c+d} \leq \frac{a}{c} &\iff b \leq \frac{a}{c} d \\ &\iff (\sqrt{d_h d_v})^3 L^2 L' (\gamma^{N+1})^2 \leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2}{\sum_{k=1}^N \gamma^k} \gamma^{N+1} \end{aligned} \quad (21)$$

Rearranging the terms in the last inequality above, we have

$$\begin{aligned} (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N \gamma^{N+1} \gamma^k &\leq D_f + (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N (\gamma^k)^2 \\ &\iff (\sqrt{d_h d_v})^3 L^2 L' \sum_{k=1}^N \gamma^k (\gamma^{N+1} - \gamma^k) \leq D_f \end{aligned} \quad (22)$$

Recall that  $D_f = 2(f(\mathbf{W}^0) - f^*)$ ; so without loss of generality we always have  $D_f \geq 0$ . With this result, the last inequality in (22) is always satisfied whenever  $\gamma^{N+1} \leq \gamma^k$  for  $k = 1, \dots, N$ . Since this needs to be true for all  $N$ , require  $\gamma^{k+1} \leq \gamma^k$  for  $k = 1, \dots, N-1$ . This proves the monotonicity of expected gradients. For the limiting case, recall the relaxed upper bound from (18). Whenever  $\lim_{N \rightarrow \infty} \sum_{k=1}^N \gamma^k \rightarrow \infty$ ,  $\lim_{N \rightarrow \infty} \sum_{k=1}^N (\gamma^k)^2 < \infty$ , the right hand side in (18) converges to 0.  $\square$

The second part of the lemma is easy to ensure by choosing diminishing step-sizes (as a function of  $k$ ). This result ensures the convergence of expected gradients, provides an easy way to construct  $\mathbb{P}_R(\cdot)$  based on (7) and (17), and to decide the stopping iteration based on  $\mathbb{P}_R(\cdot)$  ahead of time.

*Remarks.* Note that the maximum  $\gamma^k$  in (17) needed to ensure the monotonic decrease of expected gradients depends on  $L'$ . Whenever the estimate of  $L'$  is too loose, the corresponding  $\gamma^k$  might be too small to be practically useful. An alternative in such cases is to compute the RSG updates for some  $N$  (fixed a priori) iterations using a reasonably small stepsize, and select  $R$  to be the iteration with the smallest possible gradient  $\|\nabla_{\mathbf{W}} f(\mathbf{W}^k)\|^2$  or the cumulative gradient  $\sum_{i=1}^k \|\nabla_{\mathbf{W}} f(\mathbf{W}^i)\|^2$  among some last  $N_1 < N$  iterations. While a diminishing stepsize following (17) is ideal, the next result gives the best possible *constant* stepsize  $\gamma = \gamma^k, \forall k$ , and the corresponding rate of convergence.

**Corollary 3.4 (Convergence of one-layer DA).** *The optimal constant step sizes  $\gamma^k$  are given by*

$$\gamma^k = \frac{D}{\sqrt{N}(d_h d_v)^{3/4}} \quad \forall k; \quad 0 < D \leq \frac{\sqrt{N}}{L'} (d_h d_v)^{1/4} \quad (23)$$

If we denote  $\bar{D} = \frac{D_f}{D} + DL^2 L'$ , then we have

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \leq \bar{D} \frac{(d_h d_v)^{3/4}}{\sqrt{N}} \quad (24)$$

*Proof.* Using constant stepsizes  $\gamma^k = \gamma, k = 1, \dots, N$ , the convergence bound in (8) reduces to

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \leq \frac{D_f + (\sqrt{d_h d_v})^3 L^2 L' N(\gamma)^2}{N\gamma(2 - L'\sqrt{d_h d_v}\gamma)} \quad (25)$$

To achieve monotonic decrease of expected gradients, we require  $\gamma^k < \frac{1}{L'\sqrt{d_h d_v}}$  (from (17) in Lemma 3.3). For such  $\gamma^k$ s,

$$\left(2 - L'\sqrt{d_h d_v}\gamma^k\right) > 1 \quad \forall \quad k$$

which when used in (25) gives,

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \leq \frac{D_f}{N\gamma} + (\sqrt{d_h d_v})^3 L^2 L' \gamma \quad (26)$$

Observe that as  $\gamma$  increases (resp. decreases), the two terms on the right hand side of above inequality decreases (resp. increases) and increase (resp. decreases). Therefore, the optimal  $\gamma = \gamma^k$  for all  $k$ , is obtained by balancing these two terms, as in

$$\frac{D_f}{N\gamma} = (\sqrt{d_h d_v})^3 L^2 L' \gamma \implies \gamma = \gamma^k = \frac{\sqrt{D_f}}{\sqrt{N} L^2 L' (d_h d_v)^{3/4}} \quad (27)$$

However, the above choice of  $\gamma^k$  has the unknowns  $D_f$ ,  $L'$  and  $L^2$  (although note that the later two constants can be empirically estimated by sampling the loss functions  $\mathcal{L}(\cdot)$  for different choices of  $x$  and  $\mathbf{W}$ ). Replacing  $\sqrt{\frac{D_f}{L' L^2}}$  by some  $D$ , the best possible choice constant stepsize is

$$\gamma = \gamma^k = \frac{D}{\sqrt{N} (d_h d_v)^{3/4}} \quad \forall \quad k \quad (28)$$

Since  $\gamma^k$  needs to be smaller than  $\frac{1}{L'\sqrt{d_h d_v}}$  as discussed at the start of the proof, we have

$$\frac{D}{\sqrt{N} (d_h d_v)^{3/4}} < \frac{2}{L'\sqrt{d_h d_v}} \implies D \leq \frac{2\sqrt{N}}{L'} (d_h d_v)^{1/4} \quad (29)$$

Now substituting this optimal constant stepsize from (28) into the upper bound in (26) we get

$$\begin{aligned} \mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) &\leq \frac{D_f}{N\gamma} + (d_h d_v)^{3/2} L^2 L' \gamma \\ &= \frac{D_f (d_h d_v)^{3/4}}{\sqrt{N} D} + \frac{D L^2 L' (d_h d_v)^{3/4}}{\sqrt{N}} \end{aligned} \quad (30)$$

and by denoting  $\bar{D} = \frac{D_f}{D} + D L^2 L'$ , we finally have

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^R)\|^2) \leq \bar{D} \frac{(d_h d_v)^{3/4}}{\sqrt{N}} \quad (31)$$

□

The upper bound in (8) can be written as a summation of two terms, one of which involves  $D_f$ . The optimal stepsize in (23) is calculated by balancing these terms as  $N$  increases (refer to the supplement). The ideal choice for  $D$  is  $\sqrt{\frac{D_f}{L^2 L'}}$  in which case  $\bar{D}$  reduces to  $2\sqrt{D_f L' L^2}$ . For a fixed network size ( $d_h$  and  $d_v$ ), Corollary 3.4 shows that the rate of convergence for one-layer DA pre-training using RSG is  $\mathcal{O}(1/\sqrt{N})$ . It is interesting to see that the convergence rate is proportional to  $(d_h d_v)^{3/4}$  where the number of parameters of our bipartite network (of which DA is one example) is  $d_h d_v$ .

Corollary 3.4 gives the convergence properties of a single RSG run over some  $R$  iterations. However, in practice one is interested in a large deviation bound, where the best possible solution is selected from multiple independent runs of RSG. Such a large deviation estimate is indeed more meaningful than one RSG run because of the randomization over  $\eta$  in (2). Consider a  $C$ -fold RSG with  $C \geq 1$  independent RSG estimates of  $\mathbf{W}$  denoted by  $\mathbf{W}^{R_1}, \dots, \mathbf{W}^{R_C}$ . Using the expected convergence from (24), we can compute a  $(\epsilon, \delta)$ -solution defined as,



**Definition (( $\epsilon, \delta$ )-solution).** For some given  $\epsilon > 0$  and  $\delta \in (0, 1)$ , an  $(\epsilon, \delta)$ -solution of one-layer DA is given by  $\{\mathbf{W}^{R_c}\}, c = 1, \dots, C$  such that

$$Pr \left( \min_{1, \dots, C} \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) \leq \delta \quad (32)$$

$\epsilon$  governs the goodness of the estimate  $\mathbf{W}$ , and  $\delta$  bounds the probability of good estimates over multiple independent RSG runs. Since  $N$  is the maximum iteration count (i.e., maximum number of  $\mathcal{SFO}$  calls), the number of data instances required is  $S = N/t$ , where  $t$  denotes the average number of times each instance is used by the oracle. Although in practice there is no control over  $t$  (in which case, we simply have  $S \leq N$ ), we estimate the required sample size and the minimum number of folds ( $C$ ) in terms of  $t$ , as shown by the following result.

**Corollary 3.5 (Sample size estimates of one-layer DA).** *The number of independent RSG runs ( $C$ ) and the number of data instances ( $S$ ) required to compute a  $(\epsilon, \delta)$ -solution are given by*

$$C(r, \delta) \geq \left\lceil \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} \right\rceil ; \quad S(r, \epsilon) \geq \frac{r(d_h d_v)^{3/2}}{t\epsilon^2} \quad (33)$$

where  $r > 1$  is a given constant,  $\lceil \cdot \rceil$  denotes ceiling operation and  $t$  denotes the average number of times each data instance is used.

*Proof.* Recall that a  $(\epsilon, \delta)$ -solution is defined such that

$$Pr \left( \min_{1, \dots, C} \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) \leq \delta \quad (34)$$

for some given  $\epsilon > 0$  and  $\delta \in (0, 1)$ . Using basic probability properties,

$$\begin{aligned} Pr \left( \min_{1, \dots, C} \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) &= Pr \left( \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \quad \forall \quad c = 1, \dots, C \right) \\ &= \prod_{c=1}^C Pr \left( \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) \end{aligned} \quad (35)$$

Using Markov inequality and (24),

$$\begin{aligned} Pr \left( \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) &\leq \frac{\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2)}{\epsilon \bar{D}} \\ &\leq \frac{(d_h d_v)^{3/4}}{\epsilon \sqrt{N}} \end{aligned} \quad (36)$$

Hence, the number of  $\mathcal{SFO}$  calls per RSG is at least  $N > \frac{(d_h d_v)^{3/2}}{\epsilon^2}$  for the above probability to make sense. If  $r > 1$  is a constant, then the number of calls per RSG is  $N = \frac{r(d_h d_v)^{3/2}}{\epsilon^2}$ . Using this identity, and (35) and (36), we get

$$Pr \left( \min_{1, \dots, C} \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D} \right) \leq \prod_{c=1}^C \frac{1}{\sqrt{r}} = \frac{1}{r^{C/2}} \quad (37)$$

To ensure that this probability is smaller than a given  $\delta$  and noting that  $C$  is a positive integer, we have

$$\frac{1}{r^{C/2}} \leq \delta \implies C(r, \delta) \geq \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} := \left\lceil \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} \right\rceil \quad (38)$$

where  $\lceil \cdot \rceil$  denotes ceiling operation. Note that there is no randomization over the data instances among multiple instances of RSG ( $c = 1, \dots, C$ ). That is, each RSG is going to use all the available data instances. Hence, we can just look at one RSG to derive the sample size required. Let  $S$  be the number of data instances,  $t_s$  be the number of times  $s^{th}$  instance is used in one RSG and  $t = \mathbb{E}(t_s)$  be the average number of times each instance/example is used. We then have

$$N = \sum_{s=1}^S t_s \implies N \approx \mathbb{E}(N) = S\mathbb{E}(t_s) \implies S = \frac{N}{t} \geq \frac{r(d_h d_v)^{3/2}}{t\epsilon^2} \quad (39)$$

□

The above result shows that the required sample size is  $\mathcal{O}(1/\epsilon^2)$ , which is easy to see from the convergence rate of  $\mathcal{O}(1/\sqrt{N})$  in (24). The constant  $r$  in Corollary 3.5 acts like a trade-off parameter between the number of folds  $C(r, \delta)$  and the sample size  $S(r, \epsilon)$ . Hence, not surprisingly, more folds are needed to guarantee a  $(\epsilon, \delta)$ -solution for a smaller  $S$ . Note that the minimum possible  $S$  is  $\frac{(d_h d_v)^{3/2}}{t \epsilon^2}$ , below this quantity the idea of computing an  $(\epsilon, \delta)$ -solution in a large deviation sense is not meaningful (refer to proof of Corollary 3.5 in the supplement).

*Remarks.* To get a practical sense of (33), consider a DA with  $d_v = 100, d_h = 20$ . According to Corollary 3.5, the number of data instances for computing a  $(0.05, 0.05)$ -solution with  $t = 10^3$  is at least 0.3 million. Depending on the structural characteristics of the data (variance of each dimension, correlations across multiple dimensions etc.), which we do not exploit, the bound from Corollary 3.5 will overestimate the required number of samples, as expected. Overall, the convergence and sample size bounds in (24) and (33) provide some justification of a behavior which is routinely observed in practice — large number of unsupervised data instances are required for efficient pre-training of deep architectures (Chapter 4, Bengio (2009), Erhan et al. (2010)). Note that the results in the convergence bound in (24) do not differentiate between the visible and hidden layers, implying that the bound is symmetric with respect to  $d_h$  and  $d_v$ . However, there is empirical evidence that the choice of  $d_h$  would affect the reconstruction error with oversized networks giving better generalization in general Lawrence et al. (1998); Paugam-Moisy (1997). This can be seen by recalling that until  $d_h$  is more than the dimensionality of the low-dimensional manifold on which the input data lies, the DA setup may not be able to compute good estimates of  $\mathbf{W}$ . We discuss this issue in more detail when presenting our experiments in Section 5.

Recall that pre-training is done layer-wise in deep architectures with multiple hidden layers. Hence, the bounds presented above in (24) and (33) directly apply to stacked DAs with no changes. For stacked DAs the total number of  $\mathcal{SFO}$  calls would simply be the sum of the calls across all the layers. The results also provide insights regarding convolutional neural networks where one-to-two layer neural nets are learned from small regions (e.g., local neighborhoods in imaging data), whose outputs are then combined using some nonlinear pooling operation Lee et al. (2009). Observe that the sub-linear dependence of convergence rate on the network size  $(d_h, d_v)$  from (24) implies that whenever  $S$  (and hence  $N$ ) is reasonable large, small networks are learned efficiently. This partially supports the evidence that deep convolutional networks with multiple levels of pooling over large number of small networks are successful in learning complex concepts Lee et al. (2009); Krizhevsky et al. (2012). With these results in hand, we now consider the case of distributed synchronous pre-training where small parts of the whole network are learned at-a-time.

## 4 Distributed DA pre-training

The results in the previous section show that the convergence rate has polynomial dependence on the size of the network  $(d_h d_v)$ , where the number of  $\mathcal{SFO}$  calls increases as  $(d_h d_v)^{3/2}$ . Although this is unlikely to happen in practice because of the redundancies across the input data dimensions (for example, sufficiently strong correlations across multiple input dimensions, presence of invariant dimensions etc.), the results in Corollaries 3.4 and 3.5 show that pre-training *very large* DAs is impractical with smaller sample sizes (and thereby fewer iterations). There is empirical evidence supporting that this is indeed the case in practice Erhan et al. (2009); Raina et al. (2009). Several authors have suggested learning parts of the network instead. Recently, Dean et al. (2012) showed empirical results on how distributed learning substantially improves convergence while not sacrificing test-time performance. Motivated by these ideas, we extend the results presented in Section 3 to the distributed pre-training setting. We first show that the objective in (40) lends itself to be distributed in a simple way where the whole network is broken down into multiple parts, and each such *sub-network* is learned in a synchronous manner. By relating the corruption probabilities of these sub-networks to that of the parent DA, we compute a lower bound on the number of sub-networks required. Later, we present the convergence and sample size results for this distributed DA pre-training setting.

Recall that the objective of DA in (40) involves an expectation over corruptions  $\tilde{\mathbf{x}}$  where certain visible units are nullified (set to 0). This implies that the corrupted dimension does not provide any information to the hidden layer. Since the DA network is bipartite, the objective can then be separated into sub-networks (referred to as sub-DAs) — while the hidden layer remains unchanged, we use only a *subset* of all available  $d_v$  visible units. For each such sub-DA of size  $(\lceil \tau d_v \rceil, d_h)$ , where  $0 < \tau < 1$  is the fraction of the visible layer used, the inputs from all the left out  $(d_v - \lceil \tau d_v \rceil)$  visible units is zero i.e., their corruption probability is 1. Now, consider the setting where  $B$  such sub-DAs constructed by sampling  $\lceil \tau d_v \rceil$  number of visible units with replacement. The following result shows the equivalence of learning these  $B$  sub-DAs to learning one large DA of size  $(d_v, d_h)$ .

**Lemma 4.1 (Distributed learning of one-layer DA).** Consider a DA network of size  $(d_v, d_h)$  with corruption probability  $\zeta$  and some  $1 - \zeta < \tau < 1$  and  $0 < \phi \ll 1$ . Learning this DA is equivalent to learning  $B > \frac{\log(\phi)}{\log(1-\tau)}$  number of DAs of size  $(\lceil \tau d_v \rceil, d_h)$  with corruption probability  $1 - \frac{1-\zeta}{\tau}$ , whose visible units are a fraction  $\tau$  (with replacement) of the total available  $d_v$  units, where  $\lceil \cdot \rceil$  denotes the ceiling operation.

*Proof.* Recall that the DA objective is

$$\min_{\mathbf{W}} \mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}})} \|\mathbf{x} - \sigma(\mathbf{W}^T \sigma(\mathbf{W} \tilde{\mathbf{x}}))\|^2 \quad (40)$$

By considering one term from this expectation, we show that it is equivalent to learning two disjoint DAs of sizes  $(\lceil \tau d_v \rceil, d_h)$  and  $(d_v - \lceil \tau d_v \rceil, d_h)$  synchronously. Without loss of generality, let this term correspond to the last  $d_v - \lceil \tau d_v \rceil$  visible units be corrupted with probability 1 i.e., are set to 0. For the rest of the proof, any visible unit that is set to 0 via corruption will be referred to as a ‘clamped’ unit.

Let  $W_1$  (of size  $d_h \times \lceil \tau d_v \rceil$ ) and  $W_2$  (of size  $d_h \times d_v - \lceil \tau d_v \rceil$ ) be the matrices of edge weights (i.e., unknown parameters) from the un-clamped and clamped visible units to all  $d_h$  hidden units respectively. For some inputs  $\mathbf{x}$ , let  $\mathbf{x}_1$  (of length  $\lceil \tau d_v \rceil \times 1$ ) and  $\mathbf{x}_2$  ( $d_v - \lceil \tau d_v \rceil \times 1$ ) be the un-clamped and clamped parts. Hence  $\bar{\mathbf{x}}_1 = \mathbf{x}_1$  and  $\bar{\mathbf{x}}_2 = 0$ . Then the hidden activation  $\mathbf{h}$ , and the corresponding un-clamped and clamped reconstructions,  $\hat{\mathbf{x}}_1$  and  $\hat{\mathbf{x}}_2$  have the following structure,

$$\begin{aligned} \mathbf{h} &= \sigma(W_1 \bar{\mathbf{x}}_1 + W_2 0) = \sigma(W_1 \bar{\mathbf{x}}_1) & \hat{\mathbf{x}}_1 &= \sigma(W_1^T \sigma(W_1 \bar{\mathbf{x}}_1)) \\ \hat{\mathbf{x}}_2 &= \sigma(W_2^T \sigma(W_1 \bar{\mathbf{x}}_1)) = \sigma(W_2^T \sigma(W_1 \bar{\mathbf{x}}_1 + W_2 \bar{\mathbf{x}}_2)) \end{aligned} \quad (41)$$

The objective for the term considered then simplifies to

$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \|\mathbf{x}_1 - \sigma(W_1^T \sigma(W_1 \bar{\mathbf{x}}_1))\|^2 + \|\mathbf{x}_2 - \sigma(W_2^T \sigma(W_1 \bar{\mathbf{x}}_1 + W_2 \bar{\mathbf{x}}_2))\|^2 \quad (42)$$

It is easy to see that the first term from the above summation is exactly minimizing the recovery of  $\mathbf{x}_1$  with no corruption applied to it. That is to say, it corresponds to one of the terms in the objective of a smaller DA of size  $(\lceil \tau d_v \rceil, d_h)$ . The second term in the summation has similar structure however with an extra  $W_1 \bar{\mathbf{x}}_1$  within the inner sigmoid. If  $W_1$  is fixed, then this the second term is minimizing the recovery of  $\mathbf{x}_2$  with ‘complete’ corruption applied to all the  $1 - \lceil \tau d_v \rceil$  dimensions. Hence we can first pre-train the  $(\lceil \tau d_v \rceil, d_h)$  sized sub-DA, and use the learned  $W_1$  as a constant bias, and then learn the  $(d_v - \lceil \tau d_v \rceil, d_h)$  sized sub-DA. This strategy can be shown for all the terms in the objective in (40). With this, we can begin with set of sub-DAs of size  $(\lceil \tau d_v \rceil, d_h)$  each and pre-train then one at-a-time in a synchronous manner, thereby justifying the distributed setting for DA pre-training.

Now consider such a setup where many such  $(\lceil \tau d_v \rceil, d_h)$  sub-DAs are learned synchronously by randomly sampling different subsets  $\lceil \tau d_v \rceil$  of the total available visible units. It is easy to see that, in expectation this sequential distributed learning is equivalent to minimizing all the terms inside the expectation in (40). Hence learning the big  $(d_v, d_h)$  DA is the same as sequentially learning small DAs of size  $(\lceil \tau d_v \rceil, d_h)$  where the units  $\lceil \tau d_v \rceil$  are chosen at random. In practice, this is achieved only if each of the visible unit is included in at least one of the sub-DAs (i.e., all unknown parameters are updated at least once). Let  $B$  be the number of sub-DAs that are learned sequentially. If  $0 < \phi \ll 1$  denotes the probability that a given unit is not in all the  $B$  sub-DAs (ideally,  $\phi$  should be small in practice). Then, it is easy to see that this probability is given by  $(1 - \tau)^B$  because the probability that a particular unit is sampled to be included in one sub-DA is  $\tau$ . Since  $1 - \tau < 1$ , we then have

$$(1 - \tau)^B \ll \phi \implies B \gg \frac{\log(\phi)}{\log(1 - \tau)} \quad (43)$$

We now relate the corruption probabilities of the sub-DAs (denoted by  $q$ ) to that of the mother DA ( $\zeta$ ). Recall that clamping is the same as corrupting (i.e., setting the input from that unit to be 0). Given the sampling fraction  $\tau$ , the probability that a given visible unit  $(1, \dots, d_v)$  belongs to one sub-DA is  $\tau$ . Further, if  $q$  is the corruption probability of this sub-DA, then the un-clamping probability of a given unit is  $(1 - \tau) + \tau q$ . If the  $B$  sub-DAs are constructed independently by sampling the visible units with replacement, then the overall corrupting (un-clamping) probability is  $\frac{(B - \tau B) + \tau q B}{B} = 1 - \tau + \tau q$ . We require this to be equal to  $\zeta$ , which then gives  $q = 1 - \frac{1 - \zeta}{\tau}$  (with  $1 - \zeta < \tau < 1$ ).  $\square$

The above statement (proof in supplement) establishes the equivalence of distributed DA (dDA) pre-training to the non-distributed case by explicitly considering the DA’s property of using nullified/corrupted inputs (which then provide no new information to the objective). We remark that Lemma 4.1 is specific for the case of DAs and (unlike

many other results in this paper) may not be directly applicable for other types of auto-encoders that do not involve an explicit corruption function. Also,  $\tau$  and  $\zeta$  should be chosen carefully so that  $1 - \zeta < \tau$  and  $1 - \frac{1-\zeta}{\tau}$  does not end up too close to 1. Specifically, whenever  $\zeta$  is very small, according to Lemma 4.1, there is very little room for distribution because  $\tau$  will be close to 1. This is not surprising because, with small  $\zeta$ , the DA is allowed to discard visible units very rarely, pushing  $\tau$  closer to 1, where the distributed setup tends to behave like the non-distributed case. Although these requirements seem too restrictive, we show in Section 5 that they can be fairly relaxed in practice. Overall, Lemma 4.1 provides some justification (from the perspective of the autoencoder design itself) for distributing the learning process. The lower bound on  $B$  in Lemma 4.1 ensures that all the unknown parameters are updated in at least one of the  $B$  sub-DAs. Hence, in practice,  $\phi$  can be chosen to be very small and the sub-DAs can be explicitly sampled to be “non-overlapping” (i.e., disjoint with respect to the parameters). Once the hyper-parameters  $\tau$ ,  $\zeta$  and  $B$  are fixed, the recipe is simple. The dDA pre-training setup will involve running  $B$  individual RSGs on randomly sampled disjoint sub-DAs. The  $B$  sub-DAs share a common parameter set which holds the latest estimates of  $\mathbf{W}$ .

Similar to the multi-fold RSG setup in Section 3, we perform  $M$  meta-iterations of the dDA pre-training, where each meta-iteration involves learning  $B$  number of sub-DAs. Because sub-DAs are constructed randomly, different meta-iterations end up with different set of sub-DAs, ensuring low variance in the estimate of  $\mathbf{W}$  corresponding to the  $(\epsilon, \delta)$ -solution (34). It is clear that due to the reduction in the size of the network by a factor of  $\tau$ , the convergence rate and required sample sizes (see (24) and (33)) will improve in this distributed case. This observation is formalized in the two results below. Here,  $\gamma_b^k$  denotes the step size in  $k^{th}$  meta-iteration for  $b^{th}$  RSG (corresponding to  $b^{th}$  sub-DA) and  $N$  is the number of  $\mathcal{SFO}$  calls for each of the  $B$  RSGs. The subscript  $b$  in  $\mathbf{W}_b^{R_b}$  represents the updates of  $b^{th}$  RSG where  $R_b$  is its stopping iteration.

**Corollary 4.2 (Convergence of one-layer dDA).** *The optimal constant step size  $\gamma_b^k$  is given by*

$$\gamma_b^k = \frac{D}{\sqrt{N}(\tau d_h d_v)^{3/4}} \quad \forall b, k; \quad 0 < D \leq \frac{\sqrt{N}}{L'} (\tau d_h d_v)^{1/4} \quad (44)$$

By selecting  $B$  according to Lemma 4.1, and denoting  $\bar{D} = \frac{D_f}{BD} + DL^2L'$ , we have,

$$\mathbb{E}(\|\nabla_{\mathbf{W}} f(\mathbf{W}_b^{R_b})\|^2) \leq \bar{D} \frac{(\tau d_h d_v)^{3/4}}{\sqrt{N}} \quad (45)$$

*Proof.* The proof for this theorem emulates the proofs of Lemma 3.2 and Corollary 3.4. First we derive an upper bound on the expected gradients similar to the one in (8) of Lemma 3.2. Using this bound, we then compute the optimal stepsizes and the rate of convergence.

In the distributed setting, we have  $B$  number of RSGs running synchronously (or sequentially) and the size of each of the  $B$  sub-DAs is  $(\lceil \tau d_v \rceil, d_h)$ . This is the same as a  $(d_v, d_h)$  DA with  $\tau d_h d_v$  unknowns (for notational convenience the ceil operator  $\lceil \cdot \rceil$  is dropped in the analysis). So, the bounds on the variance of noisy gradients ( $G(\eta^k; \mathbf{W}^k)$ ) and the Lipschitz continuity of  $f(\mathbf{W})$  change as follows,

$$\begin{aligned} \text{Var}(G(\eta^k; \mathbf{W}^k)) &\leq \tau d_h d_v L^2 \\ \|\nabla_{\mathbf{W}} f(\mathbf{W}) - \nabla_{\mathbf{W}} f(\hat{\mathbf{W}})\| &\leq \sqrt{\tau d_h d_v} L' \|\mathbf{W} - \hat{\mathbf{W}}\| \end{aligned} \quad (46)$$

We then have the following inequality for each of the  $B$  RSGs based on the analysis in the proof of Lemma 3.2 until (10)

$$\begin{aligned} \sum_{k=1}^{N_b} \left( \gamma_b^k - \frac{\sqrt{\tau d_h d_v} L'}{2} (\gamma_b^k)^2 \right) \|f(\mathbf{W}_b^k)\|^2 &\leq f(\mathbf{W}_b^1) - f(\mathbf{W}_b^{N_b+1}) \\ &\quad - \sum_{k=1}^{N_b} \left( \gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle + \frac{\sqrt{\tau d_h d_v} L'}{2} \sum_{k=1}^{N_b} (\gamma_b^k)^2 \|\delta_b^k\|^2 \end{aligned} \quad (47)$$

It should be noted that the subscript  $b$  indicates  $b^{th}$  RSG i.e.,  $\mathbf{W}_b^k$  is the  $k^{th}$  update from  $b^{th}$  RSG.  $N_b$  denotes the maximum number of iterations of  $b^{th}$  RSG. Although the size of  $\mathbf{W}$  is  $d_h \times d_v$ , only  $\tau d_h d_v$  of the total  $d_h d_v$  are being updated within a single RSG.

Now recall that the sequential nature of the  $B$  RSGs implies that the estimate of  $\mathbf{W}$  at the end of  $b^{th}$  RSG will be the starting point for the  $(b+1)^{th}$  RSG. This implies that  $f(\mathbf{W}_b^{N_b+1}) = f(\mathbf{W}_{b+1}^1)$  for all  $b = 1, \dots, B$ . Using this fact, we can then sum up all the  $B$  inequalities of the form in (47) to get,

$$\begin{aligned} \sum_{b=1}^B \sum_{k=1}^{N_b} \left( \gamma_b^k - \frac{\sqrt{\tau d_h d_v} L'}{2} (\gamma_b^k)^2 \right) \|f(\mathbf{W}_b^k)\|^2 &\leq f(\mathbf{W}_1^1) - f(\mathbf{W}_B^{N_B+1}) \\ &\quad - \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2) \langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle + \frac{\sqrt{\tau d_h d_v} L'}{2} \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k)^2 \|\delta_b^k\|^2 \end{aligned} \quad (48)$$

Using the fact that  $f^* \leq f(\mathbf{W}_B^{N_B+1})$ , we then have

$$\begin{aligned} \sum_{b=1}^B \sum_{k=1}^{N_b} \left( \gamma_b^k - \frac{\sqrt{\tau d_h d_v} L'}{2} (\gamma_b^k)^2 \right) \|f(\mathbf{W}_b^k)\|^2 &\leq f(\mathbf{W}_1^1) - f^* \\ &\quad - \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2) \langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle + \frac{\sqrt{\tau d_h d_v} L'}{2} \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k)^2 \|\delta_b^k\|^2 \end{aligned} \quad (49)$$

We now take the expectation of the above inequality over all the random variables involved in the  $B$  RSGs, which include, the  $B$  number of stopping criterions  $R_b, b = 1, \dots, B$  and the random processes  $\eta_b^{[N_b]}, b = 1, \dots, B$  ( $\eta_b^{[N_b]}$  is the random process of  $\eta$  within  $b^{th}$  RSG). First, note the following observations about  $\langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle$  and  $\|\delta_b^k\|^2$

$$\begin{aligned} \mathbb{E}_{\eta_b^{[N_b]}}(G(\eta_b^k; \mathbf{W}_b^k)) &= \nabla_{\mathbf{W}} f(\mathbf{W}_b^k) \implies \langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle = 0 \\ \mathbb{E}_{\eta_b^{[N_b]}} \|\delta_b^k\|^2 &= \text{Var}(G(\eta_b^k; \mathbf{W}_b^k)) \leq \tau d_h d_v L^2 \end{aligned} \quad (50)$$

which follow from (46). This implies that after taking the expectation of the inequality in (49), the last two terms on the right hand side will be,

$$\mathbb{E}_{\eta^{[N]}} \left[ \sum_{b=1}^B \sum_{k=1}^{N_b} \left( \gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2 \right) \langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle \right] \quad (51)$$

$$= \sum_{b=1}^B \sum_{k=1}^{N_b} \left( \gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2 \right) \mathbb{E}_{\eta^{[N_b]}} (\langle \nabla_{\mathbf{W}} f(\mathbf{W}_b^k), \delta_b^k \rangle | \eta_b^1, \dots, \eta_b^k) = 0$$

$$\begin{aligned} \mathbb{E}_{\eta^{[N]}} \left[ \frac{\sqrt{\tau d_h d_v} L'}{2} \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k)^2 \|\delta_b^k\|^2 \right] &= \frac{\sqrt{\tau d_h d_v} L'}{2} \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k)^2 \mathbb{E}_{\eta^{[N_b]}} (\|\delta_b^k\|^2) \\ &\leq \frac{(\sqrt{\tau d_h d_v})^3 L' L^2}{2} \sum_{k=1}^{N_b} (\gamma_b^k)^2 \end{aligned} \quad (52)$$

where  $\eta^{[N]}$  denotes the composition of the  $B$  random processes  $\eta^{[N_b]}, b = 1, \dots, B$ . Using (51) and (52) and (49), we get

$$\begin{aligned} \sum_{b=1}^B \sum_{k=1}^{N_b} \left( 2\gamma_b^k - \sqrt{\tau d_h d_v} L' (\gamma_b^k)^2 \right) \mathbb{E}_{\eta^{[N]}} \|f(\mathbf{W}_b^k)\|^2 \\ \leq 2(f(\mathbf{W}_1^1) - f^*) + (\sqrt{\tau d_h d_v})^3 L' L^2 \sum_{b=1}^B \sum_{k=1}^{N_b} (\gamma_b^k)^2 \end{aligned} \quad (53)$$

Recall the definition of  $\mathbb{P}_R(k)$  from (7) in Lemma 3.2, which is

$$\mathbb{P}_R(k) = \text{Pr}(R = k) = \frac{2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2}{\sum_{k=1}^N 2\gamma^k - L' \sqrt{d_h d_v} (\gamma^k)^2} \quad k = 1, \dots, N \quad (54)$$

Adapting this to the current case of  $B$  sequential RSGs, we get

$$\mathbb{P}_{R_b}(k) = Pr(R_b = k) := \frac{2\gamma_b^k - L'\sqrt{\tau d_h d_v}(\gamma_b^k)^2}{\sum_{b=1}^B \sum_{k=1}^{N_b} 2\gamma_b^k - L'\sqrt{\tau d_h d_v}(\gamma_b^k)^2} \quad k = 1, \dots, N \quad b = 1, \dots, B \quad (55)$$

Using this distribution of stopping criterion and taking the expectation of (53) with respect the set of random variables to  $R_b, b = 1, \dots, B$ , we get

$$\begin{aligned} \mathbb{E}(\|\nabla \mathbf{w} f(\mathbf{W}_b^R)\|^2) &= \sum_{b=1}^B \sum_{k=1}^{N_b} \frac{(2\gamma_b^k - L'\sqrt{\tau d_h d_v}(\gamma_b^k)^2) \mathbb{E}_{\eta[N]}(\|\nabla \mathbf{w} f(\mathbf{W}_b^R)\|^2)}{\sum_{b=1}^B \sum_{k=1}^N (2\gamma_b^k - L'\sqrt{\tau d_h d_v}(\gamma_b^k)^2)} \\ &\leq \frac{D_f + (\sqrt{\tau d_h d_v})^3 L^2 L' \sum_{b=1}^B \sum_{k=1}^N (\gamma_b^k)^2}{\sum_{b=1}^B \sum_{k=1}^N (2\gamma_b^k - L'\sqrt{\tau d_h d_v}(\gamma_b^k)^2)} \end{aligned} \quad (56)$$

Observe that whenever  $B$  is selected as in Lemma 4.1 with sufficiently high  $\phi$ , each of the  $d_h d_v$  unknowns is updated in at least one of the  $B$  RSGs. Hence, all the unknowns are covered in the left hand side above, see 56.

We now compute the optimal stepsizes and the corresponding convergence rate using the upper bound in (56). At any given point of time only one of the  $B$  RSGs will be running. So, using Lemma 3.4, the optimal constant stepsize for  $b^{th}$  RSG is then given by

$$\gamma_b^k = \gamma_b = \frac{D}{\sqrt{N_b}(\tau d_h d_v)^{3/4}} \quad \text{where } D \leq \frac{\sqrt{N_b}}{L'}(\tau d_h d_v)^{1/4}$$

Assuming  $N_b = N$  for all  $b = 1, \dots, B$ , we then have

$$\gamma_b^k = \gamma = \frac{D}{\sqrt{N}(\tau d_h d_v)^{3/4}} \quad \forall k, b \quad \text{where } D \leq \frac{\sqrt{N}}{L'}(\tau d_h d_v)^{1/4} \quad (57)$$

With this in hand, we now derive the convergence rate. Using some constant stepsizes  $\gamma_b^k = \gamma$  for all  $k, b$  and assumption that  $N_b = N$  for all  $b$ , the upper bound in (56) becomes

$$\begin{aligned} \mathbb{E}(\|\nabla \mathbf{w} f(\mathbf{W}_b^{R_b})\|^2) &\leq \frac{D_f + (\sqrt{\tau d_h d_v})^3 L^2 L' N B \gamma^2}{N B (2\gamma - L'\sqrt{\tau d_h d_v} \gamma^2)} \\ &\leq \frac{D_f + (\sqrt{\tau d_h d_v})^3 L^2 L' N B \gamma^2}{N B \gamma} \end{aligned} \quad (58)$$

where the last inequality uses the fact that  $(2 - L'\sqrt{\tau d_h d_v} \gamma) > 1$  (which follows from Lemma 3.3 and was used in deriving the stepsizes in Lemma 3.4). Substituting for  $\gamma$  from (57) in the above inequality gives,

$$\begin{aligned} \mathbb{E}(\|\nabla \mathbf{w} f(\mathbf{W}_b^{R_b})\|^2) &\leq \frac{D_f}{N B \gamma} + (\tau d_h d_v)^{3/2} L^2 L' \gamma \\ &= \frac{D_f (\tau d_h d_v)^{3/4}}{\sqrt{N} D B} + \frac{D L^2 L' (\tau d_h d_v)^{3/4}}{\sqrt{N}} \end{aligned} \quad (59)$$

By denoting  $\bar{D} = \frac{D_f}{B D} + D L^2 L'$ , we finally have

$$\mathbb{E}(\|\nabla \mathbf{w} f(\mathbf{W}_b^{R_b})\|^2) \leq \bar{D} \frac{(\tau d_h d_v)^{3/4}}{\sqrt{N}} \quad (60)$$

□

**Corollary 4.3 (Sample size estimates of one-layer dDA).** *The number of meta-iterations ( $M$ ) and the number of data instances ( $S$ ) required to compute a  $(\epsilon, \delta)$ -solution in the distributed setting are*

$$M(r, \delta) \geq \left\lceil \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} \right\rceil \quad ; \quad S(r, \epsilon) \geq \frac{r(\tau d_h d_v)^{3/2}}{t \epsilon^2} \quad (61)$$

where  $r > 1$  is a given constant and  $t$  denotes the average number of times each data instance is used within each sub-DA.

*Proof.* First observe that there is no randomization of data instances across the  $B$  sub-DAs. Hence we can compute the sample sizes  $S$  from a single sub-DA. Secondly, since  $B \geq 1$ ,  $\bar{D}$  in Corollary 4.2 is such that  $\bar{D} \leq \frac{D_f}{D} + DL^2L'$ . Using these two facts, the computation for  $S$  then follows the steps in Lemma 3.5 with  $d_h d_v$  replaced by  $\tau d_h d_v$ . Hence we have,

$$S(r, \epsilon) \geq \frac{r(\tau d_h d_v)^{3/2}}{t\epsilon^2} \quad (62)$$

To compute the bound for  $M$  we follow the same steps in the proof of Lemma 3.5, and end up with the following inequality

$$Pr(\|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D}) \leq \frac{(\tau d_h d_v)^{3/4}}{\epsilon \sqrt{N}} \quad (63)$$

Since  $N$  is the number of calls for each of the  $B$  RSGs, we have  $N = \frac{r(\tau d_h d_v)^{3/2}}{\epsilon^2}$  using (62). Then we have,

$$Pr\left(\min_{1, \dots, M} \|\nabla_{\mathbf{W}} f(\mathbf{W}^{R_c})\|^2 \geq \epsilon \bar{D}\right) \leq \prod_{c=1}^M \frac{1}{\sqrt{r}} = \frac{1}{r^{C/2}} \quad (64)$$

and hence  $M(r, \delta) \geq \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} = \left\lceil \frac{\log(\frac{1}{\delta})}{\log(\sqrt{r})} \right\rceil$ .  $\square$

These results show that, whenever  $B$  is chosen as in Lemma 4.1, the convergence rate and sample sizes will improve by  $\tau^{3/4}$  and  $\tau^{3/2}$  respectively, if the stepsize is appropriate. The improvements may be much larger whenever  $\zeta$  is not unreasonably small (or  $\tau$  is not too close to 1).

## 5 Experiments

To evaluate the bounds presented above, we pre-trained a one-layer DA on two computer vision and one neuroimaging datasets – MNIST digits, Magnetic Resonance Images from Alzheimer’s Disease Neuroimaging Initiative (ADNI) and ImageNet. These will be referred to as *mnist*, *neuro* and *imagenet*. See supplement for complete details about these datasets, including the number of instances, features and other attributes. Briefly, *neuro* dataset has stronger correlations across its dimensions compared to others, and *imagenet* includes natural images and is very diverse/versatile.

Our experiments are two-fold. We first evaluate the non-distributed setting (Corollary 3.4, (24)) by computing the expected gradients vs. the number of  $\mathcal{SFO}$  calls ( $N$ ) and the network structure ( $d_v, d_h$ ). We then evaluate the distributed setup (Corollary 4.2, (45)) by varying the number of disjoint sub-DAs ( $B$ ) that constitute the network. The expectations in (24) and (45) are approximated by the empirical average of gradient norm (last 100 iterations). Since we are interested in the trends of convergence rates, all plots are normalized/scaled by the corresponding maximum value of expected gradients. Figure 1 shows these results: the first and second columns correspond to the non-distributed setting and the last column corresponds to the distributed setting. Each row represents one of the three datasets considered.

*Expected gradients vs  $N$ .* Figure 1(a,d,g) show that the expected gradients decrease as the number of  $\mathcal{SFO}$  calls ( $N$ ) increases. The three curves (red, black, blue) in each plot correspond to different stepsizes. The expected gradients decrease monotonically for all the curves in Figure 1(a,d,g), and their hyperbolic trend as  $N$  increases supports the  $\mathcal{O}(\frac{1}{\sqrt{N}})$  decay rate presented in (24). Unlike *mnist* and *imagenet*, *neuro* has stronger correlations across its features, and so shows a decay rate seems to be stronger than  $\frac{1}{\sqrt{N}}$  (the red curve in Figure 1(d)). The gradients, in general, also seems to be smaller for larger stepsizes (blue and black curves), which is expected because the local minima are attained faster with reasonably large stepsizes, until the minima are overshoot. Supplement shows a plot indicative of this well-known behavior.

*Expected gradients vs  $d_v, d_h$ .* The second column (Figure 1(b,e,h)) shows the influence of increasing the length of the visible layer ( $d_v$ ) for multiple  $d_h$ ’s and fixed  $N$ . As suggested by (24), the expected gradients increase as  $d_v$  increases. This rate of increase (vs. increasing  $d_v$  on  $x$ -axis) seems to be stronger for smaller values of  $d_h$  (black and green curves vs. red and blue curves). Recall that  $d_h$  should be “sufficiently” large to encode the underlying input data dependencies Paugam-Moisy (1997); Lawrence et al. (1998); Bianchini & Scarselli (2014). Hence the network

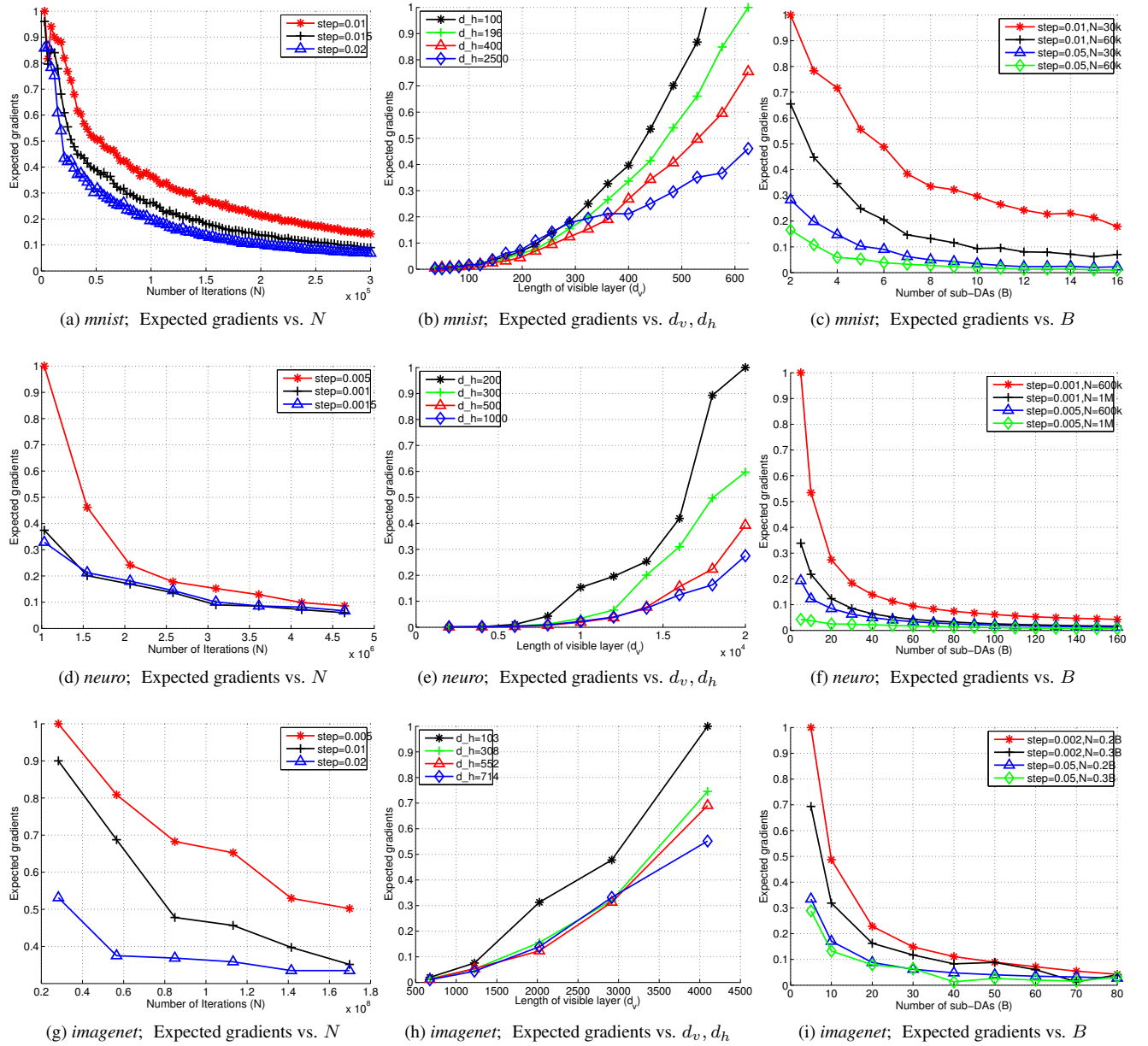


Figure 1: Expected gradients. First (a,d,g) column shows the expected gradients vs the number of *SFO* calls  $N$ , for multiple stepsizes  $\gamma$  (corresponding to red, black and blue colors). Second (b,e,h) column shows the expected gradients vs. the size of visible layer  $d_v$  for multiple  $d_h$ s (corresponding to red, blue and black colors). Third column (c,f,i) presents expected gradients vs. the number of sub-DAs ( $B > 1$ ) used in a distributed asynchronous setting (for a fixed iterations  $N$  and network size  $d_h \cdot d_v$ ). For the results in first and last columns,  $d_v$  equals the inherent input data dimensionality (see supplement), and  $d_h$  is one-tenth of  $d_v$ . Top row corresponds to *mnist*, second to *neuro* and third to *imagenet*. All the expected gradients are normalized with the maximum value in the respective plot.

may under-fit for small  $d_h$ , and not recover inputs with small error. This behavior is seen in Figure 1(b,e,h) where initially the expected gradients (across all  $d_v$ s) gradually decrease as  $d_h$  increases (black, green and red curves). Once  $d_h$  is reasonably large, increasing it further tends to increase the expected gradients (as shown by the blue curve which overlaps the others).  $d_h$  may hence be chosen empirically (e.g. using cross validation), so that the network still generalizes to test instances but is not massive (avoids unnecessary computational burden).



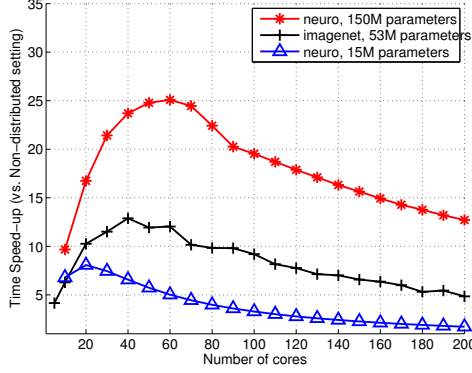


Figure 2: The relative time speed-up achieved by distributed pre-training (vs. non-distributed) as a function of the number of cores ( $x$ -axis), which in this experiment is equal to the number  $B$  of sub-DAs (see Lemma 4.1) i.e. each of core works on one sub-DA. Curves correspond to different number of parameters. Step-sizes are scaled according to 17, while  $N$  is fixed for each curve.

*Does distributed learning help?* The last column in Figure 1 shows the expected gradients in a distributed setting where  $x$ -axis represents the number of sub-DAs ( $B$ ) into which the whole network is divided. The number of  $B$ 's is chosen such that  $d_h$  is no larger than twice the size of  $d_v$ . Corollary 4.2 presents the bounds with respect to  $\tau$  which is the fraction of visible layers used in each of the sub-DAs. The results in Figure 1(c,f,i) are shown relative to the number of disjoint sub-DAs  $B$ , which is chosen to be at least  $1/\tau$  and follows the conditions in Lemma 4.1. Observe that, the expected gradients decay as  $B$  increases for all the three datasets considered. For a sufficiently large  $B$ , the decay rate settles down with no further improvement, see Figure 1(f,i). The bounds derived in Section 4 are based on a synchronous setup. In our experiments a central master holds the current updates of the parameters, and the  $B$  different sub-DAs pre-train independently on as many as 200 cores, communicating with the master via message passing. The sub-DAs are initialized by running the whole network (in a non-distributed way) for a few hundred iterations.

Figure 2 shows the time speed-up achieved by distributing the pre-training (relative to the non-distributed setting) on *neuro* and *imagenet*. Note that the number of sub-DAs used is equal to the number of cores used, which means one sub-DA is pre-trained per core. As the number of cores used increases, the speed-up relative to the non-distributed setting increases rapidly up to a certain limit, and then gradually falls back. This is because for large values of  $B$  the communication time between machines dominates the actual computation time. The speed-up is much higher for datasets with large number of parameters ( $> 50$ mil, red and black curves vs. 15mil, blue curve). Note that the distributed setting gives faster convergence and time speed-up, but does not lose out on generalization error (refer to the supplement for a plot confirming this behavior). Lastly, these computational (Figure 1(c,f,i)) and time speed-up (Figure 2) improvements of distributed setup are in agreement with existing observations Raina et al. (2009); Dean et al. (2012). Overall, the results in Figures 1 and 2, in tandem with existing observations Bengio (2009); Erhan et al. (2009, 2010); Vincent et al. (2010); Dean et al. (2012) provide strong empirical support to the convergence and sample size bounds constructed in Sections 3 and 4.

## 6 Conclusion

We analyzed the convergence rate and sample size estimates of gradient based learning of deep architectures. The only assumption we make is on the Lipschitz continuity of the loss function. We provided bounds for classical and distributed pre-training for Denoising Autoencoders, and the experiments support the suggested behavior. We believe that our results complement a sizable body of work showing the success of empirical pre-training in deep architectures and identifies a number of interesting directions for additional improvements – both on the theoretical side as well as the design of practical large scale pre-training.

## Appendix (Supplementary Material)

**Datasets Description** The three datasets that were used, *mnist*, *neuro*, *imagenet*, correspond to the small- $d$ -large- $n$ , large- $d$ -small- $n$  and large- $d$ -large- $n$  setups respectively ( $d$  is the number of data dimensions,  $n$  is the number of data

instances).

- *mnist*: This famous digit recognition dataset contains binary images of hand-written digits (0 – 9). We used  $10^4$  of these images which are part of the mnist training data set (<http://yann.lecun.com/exdb/mnist/>). The training data contains approximately equal number of instances for each of the ten classes. Each image is 784 pixels/dimensions, and the signal in each pixel is binary. No extra preprocessing was done to the data.
- *neuro*: This neuroimaging dataset is a prototypical example of dataset with very large number of features, but small number of instances. It comprises of Magnetic Resonance Imaging (MRI) data from Alzheimer’s Disease Neuroimaging Initiative study from a total of 534 subjects. Each image is three-dimensional of size  $256 \times 256 \times 176$ . Each voxel in this 3D space corresponds to water-level intensity in the brain, and the signal is positive scalar. Standard pre-processing is applied on all the images, which involves stripping out grey matter and normalizing to a template space (called MNI space). Refer to Statistical Parametric Mapping Tool (SPM8, <http://www.fil.ion.ucl.ac.uk/spm/doc/>) for this standardised procedure. The resulting processed images are sorted out according to the signal variance. For the experiments in this work, we picked out the top (most variant) 25% of the features/voxels, which amounted to  $3 \times 10^4$  features. Even within this setting the number of features is much larger than the number of instances available (534).
- *imagenet*: This well-known dataset comprises of natural images from various types of categories collected as apart of WordNet hierarchy. It comprises of more than 14 Million images, broadly categorized under more than 20 thousand synsets (<http://www.image-net.org/>). We used imaging data from five of the largest categories contained in the imagenet database. This amount to  $> 7000$  synsets/sub-categories and approximately 5 million images. As a pre-processing step, we resized all images to  $128 \times 128$  pixels, and centered each of the 16384 dimensions.

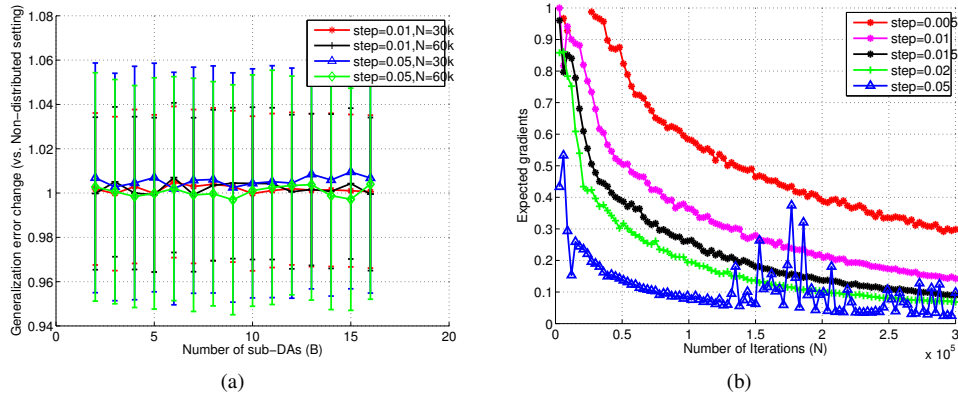


Figure 3: (a) Distributed setup does not lose on generalization error. The four curves correspond to the ratio of test-set reconstruction errors for distributed pre-training ( $B > 2$ ) to the non-distributed case. The error-bars correspond to 10 fold cv errors computed using 10 different test-sets. (b) Expected gradients vs the number of  $\mathcal{SFO}$  calls  $N$ , for multiple stepsizes  $\gamma$  (corresponding to the four different colors). The trends show that as stepsize increases the expected gradients decrease, and beyond a reasonably large stepsize (gree curve) the gradients overshoot local optima (blue curve).

## References

- Bengio, Y. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.
- Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al. Greedy layer-wise training of deep networks. *Advances in Neural information processing systems*, 19:153, 2007.
- Bianchini, M. and Scarselli, F. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. In *Neural Networks and Learning Systems, IEEE Transactions on*, pp. 1553–1565, 2014.

- Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.
- Dahl, G., Yu, D., Deng, L., and Acero, A. Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. In *Acoustics, Speech and Signal Processing (ICASSP), Proceedings of*, pp. 4688–4691, 2011.
- Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., and Ng, A. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, volume 1, pp. 1232–1240, 2012.
- Erhan, D., Manzagol, P., Bengio, Y., Bengio, S., and Vincent, P. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 153–160, 2009.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.
- Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Hamel, P. and Eck, D. Learning features from music audio with deep belief networks. In *International Society of Music Information Reterival (ISMIR) Conference*, pp. 339–344. Utrecht, The Netherlands, 2010.
- Hinton, G. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504–507, 2006.
- Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 1, pp. 4, 2012.
- Lawrence, S., Giles, C., and Tsoi, A. What size neural network gives optimal generalization? convergence properties of backpropagation. 1998.
- Lee, H., Grosse, R., Ranganath, R., and Ng, A. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine Learning (ICML-09)*, pp. 609–616, 2009.
- Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pp. 855–863, 2014.
- Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q., and Ng, A. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 265–272, 2011.
- Paugam-Moisy, A. Size of multilayer networks for exact learning: analytic approach. In *Advances in Neural Information Processing Systems*, volume 9, pp. 162, 1997.
- Plis, S., Hjelm, D., Salakhutdinov, R., and Calhoun, V. Deep learning for neuroimaging: a validation study. *arXiv preprint arXiv:1312.5847*, 2013.
- Raina, R., Madhavan, A., and Ng, A. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th International Conference on Machine Learning (ICML-09)*, volume 9, pp. 873–880, 2009.
- Rifai, S., Dauphin, Y., Vincent, P., Bengio, Y., and Muller, X. The manifold tangent classifier. In *Advances in Neural Information Processing Systems*, pp. 2294–2302, 2011a.
- Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 833–840, 2011b.
- Saxe, A., Koh, P., Chen, Z., Bhand, M., Suresh, B., and Ng, A. On random weights and unsupervised feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1089–1096, 2011.

- Socher, R., Huang, E., Pennington, J., Ng, A., and Manning, C. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, volume 24, pp. 801–809, 2011a.
- Socher, R., Pennington, J., Huang, E., Ng, A., and Manning, C. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing*, pp. 151–161, 2011b.
- Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1701–1708. IEEE, 2014.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 9999:3371–3408, 2010.