Marginalized Stacked Denoising Autoencoders

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Q. Weinberger
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO 63130
chenm, zhixiang.xu, kilian@wustl.edu

Fei Sha
Computer Science Department
University of Southern California
Los Angeles, CA 90089
feisha@usc.edu

Stacked Denoising Autoencoders (SDAs) [4] have been used successfully in many learning scenarios and application domains. In short, denoising autoencoders (DAs) train one-layer neural networks to reconstruct input data from partial random corruption. The denoisers are then stacked into deep learning architectures where the weights are fine-tuned with back-propagation. Alternatively, the outputs of intermediate layers can be used as input features to other learning algorithms. These learned feature representations are known to improve classification accuracies in many cases. For example, Glorot et. al. [3] applied SDAs to domain adaptation and demonstrated that these learned features , when used with a simple linear SVM classifier, yield record performance in benchmark sentiment analysis tasks [1].

One downside of SDAs is the arguably long training time, which often entails specialized computing supports such as GPUs, especially for large-scale tasks. In this abstract we propose a variation to SDAs, in which the random corruption is marginalized out. This crucial step yields the optimal reconstruction weights computed in closed-form and eliminates the use of back-propagation for tuning. We show that the features learned with our approach lead to comparable classification accuracy as SDAs'. The training time, however, reduces by *orders of magnitude* – from up to 19 hours for SDAs to mere 3 minutes with our approach.

Linear Denoiser. The basic building block of our framework is a one-layer *linear* denoising autoencoder. From a given set of inputs D, we sample inputs $\mathbf{x}_1, \dots \mathbf{x}_m$ with replacement, where typically $m > |D|$. We corrupt these inputs by random feature removal — each feature is set to 0 with probability p. Let us denote the corrupted version of $\mathbf{x}_i$ as $\tilde{\mathbf{x}}_i$. As opposed to the nonlinear encoder in SDAs, we reconstruct the corrupted inputs with a *linear* mapping $\mathbf{W} : \mathcal{R}^d \rightarrow \mathcal{R}^d$, that minimizes the squared reconstruction loss

$$\mathcal{L}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i\|^2. \tag{1}$$

To simplify notation, we assume that a constant feature is added to the input, $\mathbf{x}_i = [\mathbf{x}_i; 1]$, and a corresponding bias is incorporated within the mapping $[\mathbf{W}, \mathbf{b}]$. The constant feature is never corrupted. Let us define the design matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]$ and $\widetilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m]$ to be its corrupted version. Then the solution of (1) can be expressed as the well-known closed-form solution for ordinary least squares

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1}, \text{ where } \mathbf{Q} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top \text{ and } \mathbf{P} = \mathbf{X}\widetilde{\mathbf{X}}^\top. \tag{2}$$

Noise Marginalization. The solution to (2) depends on the re-sampling of the inputs $\mathbf{x}_1, \dots, \mathbf{x}_m$ and which features are randomly corrupted. Ideally, we would like to consider all possible corruptions of all possible inputs when the denoising transformation $\mathbf{W}$ is computed, *i.e.* letting $m \rightarrow \infty$. By the weak law of large numbers, the matrices $\mathbf{P}$ and $\mathbf{Q}$ converge to their expected values $E[\mathbf{Q}], E[\mathbf{P}]$ as we create more copies of the corrupted data. In the limit, we can derive their expectations and express the corresponding mapping for $\mathbf{W}$ in closed form as

$$\mathbf{W} = E[\mathbf{P}]E[\mathbf{Q}]^{-1}, \text{ where: } E[\mathbf{Q}]_{\alpha,\beta} = \begin{cases} \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha\mathbf{q}_\beta & \text{if} \quad \alpha \neq \beta \\ \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha & \text{if} \quad \alpha = \beta \end{cases}, \text{ and } E[\mathbf{P}]_{\alpha\beta} = \mathbf{S}_{\alpha\beta}\mathbf{q}_\beta, \tag{3}$$

where $\mathbf{q} = [1-p, \dots, 1-p, 1]^\top \in \mathcal{R}^{d+1}$ and for notational convenience, $\mathbf{S} = \mathbf{X}\mathbf{X}^\top$ denotes the covariance matrix of the uncorrupted data. We refer to this closed-form denoising layer as marginalized Denoising Autoencoder (mDA).

Topic: Learning algorithms **Presentation Preference**: Oral

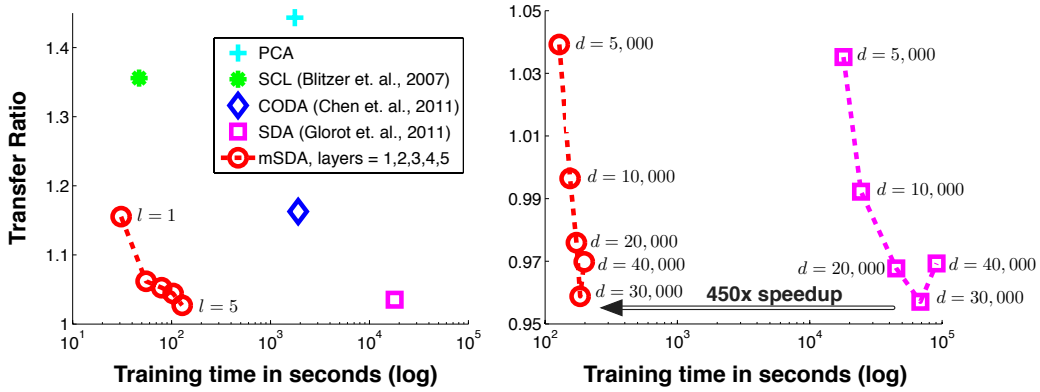

Figure 1: Transfer ratio and training times across 12 domain adaptation tasks, see texts for details.

marginalized Stacked Denoising Autoencoder (mSDA). A key component of the success of SDAs is the fact that they consist of multiple stacked layers of denoising autoencoders, which creates a "deep" learning architecture. Our framework has the same capability. We stack several mDA layers together by feeding the representations of the t^{th} denoising layer as the input to the $(t+1)^{th}$ layer. Each transformation $\mathbf{W}^t$ is learned to reconstruct the previous mDA output $\mathbf{h}^{t-1}$ from its corrupted equivalent. In order to extend our mapping beyond a linear transformation, we apply a non-linear "squashing"-function between layers. We obtain each layer's representation from its pre-ceeding layer through $\mathbf{h}^t = \tanh\left(\mathbf{W}^t \mathbf{h}^{t-1}\right)$, with $\mathbf{h}^0 = \mathbf{x}$ denoting the input.

High Dimensional Data. Many data sets (*e.g.* bag-of-words text documents) are naturally high dimensional. For the feature reconstruction to be successful, SDAs typically learn $O(d^2)$ parameters. This is costly in training time and prevents SDAs from extracting information from rarer but important features. High dimensionality also poses a challenge to mSDA, as the inversion of the outer-product matrix $\mathbf{Q}, E[\mathbf{Q}] \in \mathcal{R}^{d \times d}$ in (2) and (3) would become prohibitively expensive. To overcome this challenge, we leverage the concept of "pivot features" from [1]. Instead of aiming to reconstruct all the corrupted features at once, we reconstruct a subset of pivot features only (here the 5000 most common features). We divide all input features into K subsets and learn K rectangular matrices for pivot-feature reconstruction. K is chosen so that each subset is of a manageable size. The K resulting pivot reconstructions are summed. Subsequent layers are in the pivot-space only and require no special treatment. We use this approach to scale-up the dimensionality of both SDAs, and mSDA.

Results. Figure 1 shows the sentiment analysis results on Amazon review benchmark [1]. The left plot compares mSDA (with $1, 2, 3, 4, 5$ layers respectively) with SDAs [3], Co-training for domain adaptation (CODA) [2], Structural Correspondence Learning (SCL) [1] and simple PCA feature transformation. The transfer ratio denotes the ratio of the classification error of the adapted classifier (trained on source) over by the error of a classifier trained on true target-domain data of 5,000 pivot features. Two trends can be observed: 1. the transfer ratio of mSDA keeps improving with additional layers; and 2. the training time of mSDA is two orders of magnitude below that of SDAs with comparable transfer ratio. The right plot shows the SDA and mSDA performance as the input dimensionality of the data increases (words are picked in decreasing order of their frequency). Clearly, algorithms benefit from having more features up to $30,000$. mSDA matches the performance of SDA consistently and is up to 450 times faster in training.

Acknowledgements. We thank the authors of [3] for sharing their code and providing helpful advice.

References

[1] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. Association for Computational Linguistics, 2006.

[2] M. Chen, K.Q. Weinberger, and Y. Chen. Automatic Feature Decomposition for Single View Co-training. In *International Conference on Machine Learning*, 2011.

[3] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*, 2011.

[4] P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.