Virtual Adversarial Ladder Networks for Semi-Supervised Learning

Saki Shinoda[1], **Daniel E. Worrall**[2] **& Gabriel J. Brostow**[2]
Computer Science Department
University College London
United Kingdom
[1]saki.shinoda.16@ucl.ac.uk
[2]{d.worrall, g.brostow}@cs.ucl.ac.uk

Abstract

Semi-supervised learning (SSL) partially circumvents the high cost of labeling data by augmenting a small labeled dataset with a large and relatively cheap unlabeled dataset drawn from the same distribution. This paper offers a novel interpretation of two deep learning-based SSL approaches, ladder networks and virtual adversarial training (VAT), as applying distributional smoothing to their respective latent spaces. We propose a class of models that fuse these approaches. We achieve near-supervised accuracy with high consistency on the MNIST dataset using just 5 labels per class: our best model, ladder with layer–wise virtual adversarial noise (LVAN-LW), achieves $1.42\% \pm 0.12$ average error rate on the MNIST test set, in comparison with $1.62\% \pm 0.65$ reported for the ladder network. On adversarial examples generated with L_2-normalized fast gradient method, LVAN-LW trained with 5 examples per class achieves average error rate $2.4\% \pm 0.3$ compared to $68.6\% \pm 6.5$ for the ladder network and $9.9\% \pm 7.5$ for VAT.

1 Introduction

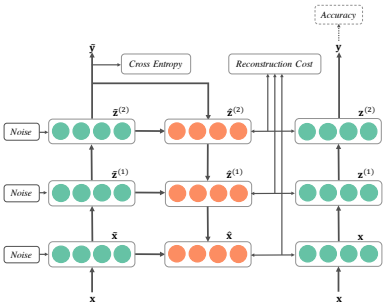
Ladder networks [15, 11] and virtual adversarial training [9] are two seemingly unrelated deep learning methods that have been successfully applied to semi-supervised learning (SSL). Below we present the view that both methods share statistical information between labeled and unlabeled examples by smoothing the probability distributions over their respective latent spaces. With this interpretation, we propose a new class of deep models for SSL that apply spatially-varying, anisotropic smoothing to latent spaces in the direction of greatest curvature of the unsupervised loss function.

In two models we investigate, we apply a virtual adversarial training cost in addition to ladder classification and denoising costs; in the other two models, we inject virtual adversarial noise into the encoder path. We train our models with 5, 10, or 100 labeled examples per class from the MNIST dataset, evaluating performance on both the standard test set and adversarial examples generated using the fast gradient method [3]. We found our models achieve state-of-the-art accuracy with high stability in the 5- or 10- labels per class setting on both normal and adversarial examples for MNIST.

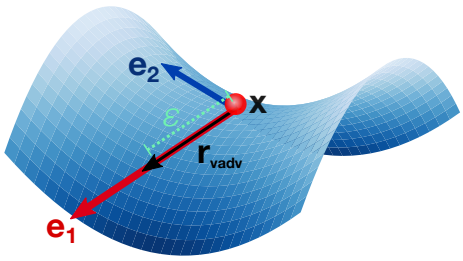
2 Background

In this section, we outline the ladder network [15, 11] and Virtual Adversarial Training (VAT) [9]. We present the ladder network as representing data in a hierarchy of nested latent spaces. SSL is performed by smoothing the labeled and unlabeled data distributions over this hierarchy, thus sharing distributional information between labeled and unlabeled distributions in a coarse-to-fine regime. In VAT, there is no latent space hierarchy, but instead a particularly clever choice of smoothing operator.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



(a) Illustration of a ladder network architecture.
Left: Noisy encoder with activations $\tilde{\mathbf{z}}^{(l)}$, additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ at each layer, and outputs $\tilde{\mathbf{y}}$.
Centre: Decoder; input from layer above and corresponding layer in noisy encoder are combined by a denoising function $g^{(l)}(\cdot, \cdot)$ to form reconstructions $\hat{\mathbf{z}}^{(l)}$.
Right: Clean encoder, weights shared with noisy encoder; activations $\mathbf{z}^{(l)}$ are denoising targets.



(b) Conceptual illustration of virtual adversarial perturbation $\mathbf{r}_{\text{vadv}}$ on a surface representing the divergence $D\left[\Pr(y|\mathbf{x}, \boldsymbol{\theta}), \Pr(y|\mathbf{x} + \mathbf{r}, \boldsymbol{\theta})\right]$. $\mathbf{e}_1$, $\mathbf{e}_2$ indicate eigenvectors of the Hessian $\nabla\nabla_r D|_{\mathbf{r}=\mathbf{0}}$. $\mathbf{r}_{\text{vadv}}$ lies parallel to the dominant eigenvector $\mathbf{e}_1$ and has magnitude equal to the hyperparameter ϵ.

The Ladder Network For both supervised and unsupervised tasks, the ladder network [15, 11] uses a single autoencoder-like architecture with added skip connections from encoder to decoder. For labeled examples, the encoder is used as a feed-forward classifier, and for the unsupervised task, the full architecture is used as a denoising autoencoder, with extra reconstruction costs on intermediate representations (see Figure 1a). For the denoising autoencoder, additive spherical Gaussian noise is applied to encoder activations $\mathbf{z}^{(\ell)}$, which we interpret as applying isotropic smoothing to the hierarchy of latent spaces modelled by the ladder network. We denoted the smoothed activations as $\tilde{\mathbf{z}}^{(\ell)}$. Mathematically, we have

$$\Pr(\{\tilde{\mathbf{z}}^{(1)}, .., \tilde{\mathbf{z}}^{(L)}\}|\mathbf{x}) = \prod_{\ell=1}^{L} \int \mathcal{N}(\tilde{\mathbf{z}}^{(\ell)}; \mathbf{z}^{(\ell)}, \sigma^2_{\text{Ladder}} I) \delta(\mathbf{z}^{(\ell)} - \mathbf{y}^{(\ell-1)}) \, \mathrm{d}\tilde{\mathbf{z}}^{(\ell)} \tag{1}$$

$$\mathbf{y}^{(\ell)} = f_\ell(\mathbf{W}^{(\ell)} \tilde{\mathbf{z}}^{(\ell)} + \mathbf{b}^{(\ell-1)}) \tag{2}$$

where $\mathbf{y}^{(\ell-1)}$ is the output of the previous layer for $\ell > 0$ and $\mathbf{y}^{(0)} = \mathbf{x}$, f_ℓ is the nonlinearity, $\mathbf{W}^{(\ell)}$ is the weights, $\mathbf{b}^{(\ell)}$ is the bias. $\mathcal{N}(\tilde{\mathbf{z}}^{(\ell)}; \mathbf{z}^{(\ell)}, \sigma^2_{\text{Ladder}} I)$ is a normal distribution over injected noise given mean $\mathbf{z}^{(\ell)}$ and variance $\sigma^2_{\text{Ladder}}\mathbf{I}$, $\mathbf{I}$ being the identity matrix.

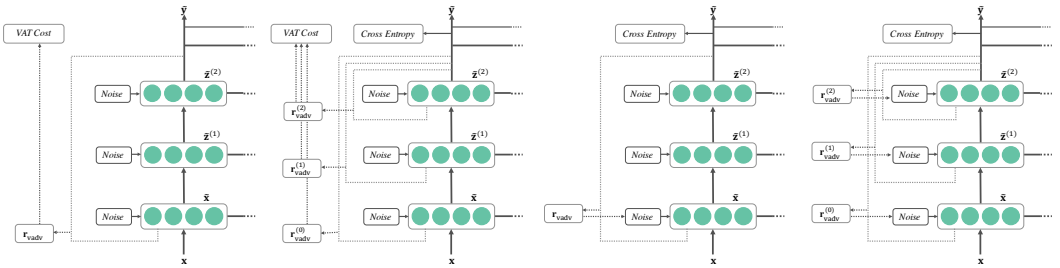
The ladder network architecture is illustrated in Figure 1a for a ladder network with $L = 2$ layers. The ladder network is trained to simultaneously minimize a negative log-likelihood on labeled examples and a denoising reconstruction cost at each layer on unlabeled examples. Specifically, the reconstruction cost is a squared error between the decoder activation $\hat{\mathbf{z}}^{(\ell)}$ and the noiseless encoder activation $\mathbf{z}^{(\ell)}$. Forward passes at training time through the model can be seen as Monte Carlo sampling from equation 1.

Virtual Adversarial Training Virtual adversarial perturbations (VAP) were first presented in [9] extending adversarial perturbations from [3] to the case where there are no labels. Adversarial perturbations are computed as

$$\mathbf{r}_{\text{adv}} := \underset{\mathbf{r};\|\mathbf{r}\|\leq\epsilon}{\arg\max} \, D\left[h(y), \Pr(y|\mathbf{x} + \mathbf{r}, \boldsymbol{\theta})\right], \tag{3}$$

where ϵ is a parameter dictating the size of the perturbation; $h(y)$ is the target distribution, *i.e.,* a one-hot vector of the true labels; $\Pr(y|\mathbf{x} + \mathbf{r}, \boldsymbol{\theta})$ is the output probabilities of the model with parameters $\boldsymbol{\theta}$; and $D[p, q]$ is a statistical divergence between P and Q (*i.e.,* a positive definite functional, only equal to zero when $P = Q$), such as the Kullback–Leibler (KL) divergence [7]. VAPs are generated by approximating $h(y)$ with $\Pr(y|\mathbf{x}, \boldsymbol{\theta})$. The perturbation is defined as

$$\mathbf{r}_{\text{vadv}} := \underset{\mathbf{r};\|\mathbf{r}\|_2\leq\epsilon}{\arg\max} \, D\left[\Pr(y|\mathbf{x}, \boldsymbol{\theta}), \Pr(y|\mathbf{x} + \mathbf{r}, \boldsymbol{\theta})\right], \tag{4}$$



(a) LVAC (b) LVAC-LW (c) LVAN (d) LVAN-LW

Figure 2: Conceptual illustrations of our proposed models. All show corrupted encoder path only. Decoder and clean encoder (not shown) are identical to that of standard ladder in Figure 1a.

In practice, as further detailed in [8], the direction of $\mathbf{r}_{\text{vadv}}$ is approximated via second-order Taylor expansion as the dominant eigenvector of the Hessian matrix of the divergence D with respect to the perturbation $\mathbf{r}$; the eigenvector is computed with a finite difference power method [2]. At training time, the virtual adversarial perturbed examples $\mathbf{x} + \mathbf{r}_{\text{vadv}}$ are added to the training set and a virtual adversarial training loss is added to the current loss. The virtual adversarial training loss is defined as the average over all input data points of

$$L_{\text{vadv}}(\mathbf{x}, \boldsymbol{\theta}) := D\left[\Pr(y|\mathbf{x}, \boldsymbol{\theta}), \Pr(y|\mathbf{x} + \mathbf{r}_{\text{vadv}}, \boldsymbol{\theta})\right]. \tag{5}$$

Minimization of this term can be seen as a smoothing operation, since it penalizes differences in $\Pr(y|\mathbf{x}, \boldsymbol{\theta})$, in its direction of greatest curvature. The range of smoothing and hence strength of regularization is controlled by the parameter ϵ, the magnitude of the VAP. [8] found that fixing the coefficient of L_{vadv} relative to the supervised cost to 1 and tuning ϵ was sufficient to achieve good results, rather than tuning both.

3 Methods

Ladder with virtual adversarial costs (LVAC and LVAC-LW) One approach for applying the anisotropic smoothing in output space of VAT to the ladder network is to add a virtual adversarial cost term (as in Equation 5) to the supervised cross-entropy cost and unsupervised activation reconstruction cost which is optimized to train the ladder network. In the most general formulation of VAT on a ladder, the loss term can be written

$$C_{\text{vadv}} = \sum_l \alpha^{(l)} D_{KL}\left[\Pr(\tilde{y}|\tilde{\mathbf{z}}^{(l)}, \boldsymbol{\theta}), \Pr(\tilde{y}|\tilde{\mathbf{z}}^{(l)} + \mathbf{r}^{(l)}_{\text{vadv}}, \boldsymbol{\theta})\right], \tag{6}$$

$$\mathbf{r}^{(l)}_{\text{vadv}} = \underset{\mathbf{r};\, \|\mathbf{r}\|_2 \le \epsilon^{(l)}}{\arg\max}\ D_{KL}\left[\Pr(\tilde{y}|\tilde{\mathbf{z}}^{(l)}, \boldsymbol{\theta}), \Pr(\tilde{y}|\tilde{\mathbf{z}}^{(l)} + \mathbf{r}, \boldsymbol{\theta})\right] \tag{7}$$

where $\tilde{\mathbf{z}}^{(l)}$ is the activation in layer l of the corrupted encoder path, with $\tilde{\mathbf{z}}^{(0)} = \tilde{\mathbf{x}}$. This gives the *ladder with virtual adversarial costs, layer-wise (LVAC-LW)*, illustrated conceptually in 2b. Applying VAP to only the input images rather than the intermediate activations in the encoder gives us the *ladder with virtual adversarial cost (LVAC)*, illustrated in Figure 2a.

Ladder with virtual adversarial noise (LVAN and LVAN-LW) Alternatively, VAT smoothing can be applied to the ladder network by injecting virtual adversarial perturbations into the activations at each layer of the corrupted encoder in addition to isotropic Gaussian noise. VAP for each layer can be computed with the same form of Equation 7 for any layer l. As with the models proposed above, the addition of noise can be on the input images only, giving the *ladder with virtual adversarial noise (LVAN)* which is illustrated in Figure 2c. The alternative case of adding noise to each layer of the encoder, *ladder with virtual adversarial noise, layer-wise (LVAN-LW)*, is shown in Figure 2d.

4 Experiments

The structure of the classifier encoder for all of our models was a fully-connected network with layers of 1000, 500, 250, 250, 250, 10 units respectively. In all ladder implementations the decoder was

symmetric to the encoder. All models were trained for 250 epochs using the Adam optimizer [4] with initial learning rate 0.002 and linear learning rate decay from 200 epochs. All models were trained with unlabeled batch size 100; labeled batch size was 50 for training with 50 labeled examples and 100 otherwise. VAP's were generated with L_∞-norm. Hyperparameters were very roughly tuned using Bayesian optimization [13]; values used are given in Appendix A.

Low labeled data Table 1 shows average error rates on MNIST with 50, 100 and 1000 labeled examples for each of our proposed models and benchmark implementations of the ladder network and VAT. Mean and standard deviation for 50 labels is computed across ten training runs with different random seeds (fixed between models) for selecting labeled data and initializing weights; mean and standard deviations on 100 and 1000 labels are computed over five training runs. We expect high variability between training runs for very few labeled examples as performance depends significantly on the particular examples chosen. In this setting LVAN and LVAN-LW are notably highly stable in achieving very good performance.

Table 1: Average Error Rate (%) and standard error on MNIST

Model	**50 labels**		**100 labels**		**1000 labels**	
VAT (ours)	5.38	$\pm$ 2.92	2.14	$\pm$ 0.64	1.11	$\pm$ 0.05
Ladder (ours)	1.86	$\pm$ 0.43	1.45	$\pm$ 0.36	**1.10**	$\pm$ **0.05**
LVAC	2.42	$\pm$ 1.05	1.65	$\pm$ 0.12	1.28	$\pm$ 0.07
LVAC-LW	4.08	$\pm$ 3.55	1.39	$\pm$ 0.06	1.11	$\pm$ 0.12
LVAN	1.52	$\pm$ 0.20	1.30	$\pm$ 0.09	1.48	$\pm$ 0.03
LVAN-LW	**1.42**	$\pm$ **0.12**	**1.25**	$\pm$ **0.06**	1.51	$\pm$ 0.06

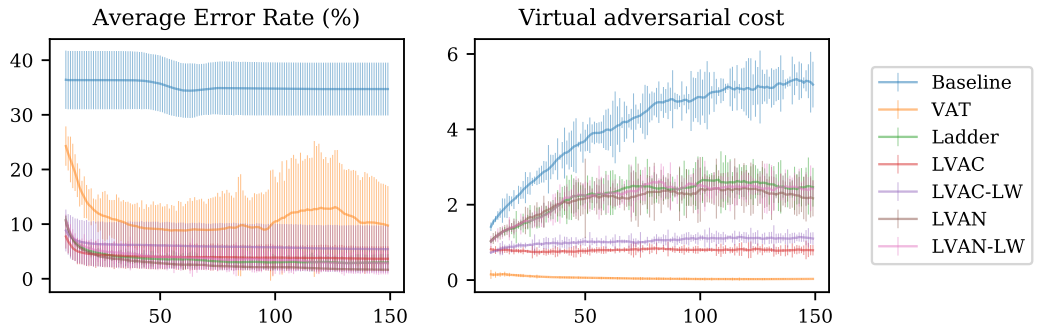
Adversarial examples We tested the performance of our models on adversarial examples generated using the CleverHans implementation of the fast gradient method with L_1, L_2 and L_∞ norms [3, 10]. For adversarial examples generated with L_∞ norm, VAT outperformed the ladder network and all of our proposed models with 50, 100 and 1000 labeled examples. VAT and all of our models showed improvement in AER as the number of labeled examples increased, while the ladder network did not appear to improve substantially. On L_1 and L_2 adversarial examples, LVAN-LW followed closely by LVAN outperformed all models including VAT and ladder for 50, 100 labels, while LVAC and LVAC-LW performed better than LVAN, LVAN-LW, VAT and ladder for 1000 labels. The ladder network performed very poorly on 50 and 100 labels, while VAT AER for 1000 labels appears to have been limited by the relatively high AER on non-adversarial examples. The relative strength of the LVAC models on L_∞ examples compared to L_2 or L_1 where the LVAN models performed better could be due to the relative strengths of regularization by VAT cost, which is based on L_∞ perturbations, and the L_2 denoising cost. Full results are presented in Table 2.

Table 2: Mean average error rate (%) and standard error on adversarial examples from MNIST

Norm	**Labels**	VAT	Ladder	LVAC	LVAC-LW	LVAN	LVAN-LW
$\boldsymbol{L_\infty}$	50	**22 $\pm$ 5.3**	54 $\pm$ 6	49 $\pm$ 2	34 $\pm$ 6	59 $\pm$ 5	56 $\pm$ 3
	100	**16 $\pm$ 1**	58 $\pm$ 2	48 $\pm$ 1	31 $\pm$ 2	60 $\pm$ 3	60 $\pm$ 2
	1000	**10.6 $\pm$ 0.4**	53 $\pm$ 2	36 $\pm$ 1	27 $\pm$ 3	40 $\pm$ 1	43 $\pm$ 2
$\boldsymbol{L_2}$	50	10 $\pm$ 8	26 $\pm$ 5	3 $\pm$ 2	5 $\pm$ 4	1.8 $\pm$ 0.4	**1.6 $\pm$ 0.2**
	100	3 $\pm$ 2	1.6 $\pm$ 0.4	1.7 $\pm$ 0.1	1.4 $\pm$ 0.1	1.4 $\pm$ 0.1	**1.3 $\pm$ 0.1**
	1000	2.4 $\pm$ 0.2	1.5 $\pm$ 0.1	1.4 $\pm$ 0.1	**1.2 $\pm$ 0.2**	1.6 $\pm$ 0.1	1.7 $\pm$ 0.1
$\boldsymbol{L_1}$	50	10 $\pm$ 8	69 $\pm$ 7	3 $\pm$ 2	4 $\pm$ 4	2.5 $\pm$ 0.4	**2.4 $\pm$ 0.3**
	100	4 $\pm$ 2	2.3 $\pm$ 0.4	2.2 $\pm$ 0.1	**1.9 $\pm$ 0.1**	2.0 $\pm$ 0.2	**1.9 $\pm$ 0.2**
	1000	2.6 $\pm$ 0.2	2.3 $\pm$ 0.2	**1.8 $\pm$ 0.2**	1.6 $\pm$ 0.1	2.2 $\pm$ 0.1	2.3 $\pm$ 0.2

Introspection: measuring smoothness The virtual adversarial loss given in Equation 5 is a measure of *lack of* local smoothness of the output distribution with respect to the input images. We expect that the ladder and LVAN models, though they do not explicitly minimize this cost, still perform

Figure 3: Comparison of average error rate and virtual adversarial cost (3 runs of 150 epochs each).



smoothing that should be reflected in this metric. This cost was computed with $\epsilon = 5.0$ for all models over 150 epochs of training (Figure 3). As expected, VAT, which directly minimizes this cost, is most smooth by this metric. The benchmark ladder network is significantly smoother than the fully supervised baseline despite not explicitly minimising the virtual adversarial cost. We measure LVAN and LVAC to be smoother than LVAN-LW and LVAC-LW respectively. This suggests smoothing with respect to the input image, which this metric measures, is traded off in the layer-wise models with smoothing in the intermediate latent spaces.

5 Discussion and Conclusions

In this work, we conducted an analysis of the ladder network from [11] and virtual adversarial training (VAT) from [9, 8] for semi-supervised learning and proposed four variants of a model applying virtual adversarial training to the ladder network: ladder with virtual adversarial cost (LVAC), ladder with layer-wise virtual adversarial cost (LVAC-LW), ladder with virtual adversarial noise (LVAN), and ladder with layer-wise virtual adversarial noise (LVAN-LW).

Based on the manifold and cluster assumptions of semi-supervised learning [1], we hypothesised that virtual adversarial training could improve the classification accuracy of the ladder network trained in a semi-supervised context. We tested this hypothesis on the MNIST dataset [6], by training models on training sets consisting of 50, 100 or 1000 labeled examples augmented by the full 60,000 images in the MNIST training set as unlabeled examples. We measured performance as error rate on the held-out test set of 10,000 examples.

We found that our models, most significantly LVAN-LW, improved on the performance of the ladder for 50 labels and 100 labels, achieving state-of-the-art error rates. For 1000 labels, both VAT and ladder baselines outperformed our models. This leads us to believe that the additional regularization provided by VAP's to the ladder network are useful only when the task is sufficiently challenging, suggesting that we should test our models on more complex datasets such as SVHN or CIFAR-10.

Additionally we found that our models performed better than the ladder network on adversarial examples. VAT outperformed our models for L_∞ adversarial examples, but our models, again especially the LVAN-LW model, achieved best performance for the few-label cases (50 and 100 labels) on L_1 and L_2-normalized adversarial examples.

Our best-performing models overall were based on adding virtual adversarial noise to the corrupted encoder path of the ladder (LVAN and LVAN-LW). These have additional advantages over the LVAC models proposed: they are faster, as they require fewer passes through the network, and produce more stable, consistent results.

A natural extension of our work would be to extend our interpretation to deep SSL methods which have been more recently introduced such as temporal ensembling [5], random data augmentation [12], and the Mean Teacher method [14].

Acknowledgements

This work was carried out by Saki Shinoda while at UCL and completed while at Prediction Machines Pte Ltd. Daniel Worrall is funded by Fight For Sight, UK.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, Massachusetts, 2006.

[2] G. H. Golub and H. A. van der Vorst. Eigenvalue computation in the 20th century. *J. Comput. Appl. Math.*, 123(1-2):35–65, Nov. 2000.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, Dec. 2014. Presented at the 3rd International Conference on Learning Representations (San Diego, CA, USA, 7–9 May 2015).

[4] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, Dec. 2014. Presented at the 3rd International Conference on Learning Representations (San Diego, CA, USA, 7–9 May 2015).

[5] S. Laine and T. Aila. Temporal Ensembling for Semi-Supervised Learning. *ArXiv e-prints*, Oct. 2016. Presented at the 5th International Conference on Learning Representations (Toulon, FR, 24–26 April 2017).

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[7] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[8] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning. *arXiv preprint arXiv:1704.03976*, Apr. 2017.

[9] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional Smoothing with Virtual Adversarial Training. *arXiv preprint arXiv:1507.00677*, July 2015. Presented at the 4th International Conference on Learning Representations (San Juan, PR, USA, 2–4 May 2016).

[10] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel. cleverhans v1.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.

[11] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko. Semi-supervised learning with ladder networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3546–3554, Cambridge, MA, USA, 2015. MIT Press.

[12] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[13] scikit-optimize contributors. Scikit-Optimize: Sequential model-based optimization with a 'scipy.optimize' interface, 2017. Software used under BSD license.

[14] A. Tarvainen and H. Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017. Accepted as a conference paper at 2017 conference on Neural Information Processing Systems.

[15] H. Valpola. From neural PCA to deep unsupervised learning. *arXiv preprint arXiv:1411.7783*, Nov. 2014.

A Hyperparameters

Model	**Labels**	$\lambda^{(0)}$	$\lambda^{(1)}$	$\lambda^{(\geq 2)}$	$\epsilon^{(0)}$	$\epsilon^{(1)}$	$\epsilon^{(\geq 2)}$
LVAC	50	1504	16.15	0.0381	0.0733	-	-
	100	1966	14.20	0.1563	0.0731	-	-
	1000	3883	12.35	0.0539	2.5206	-	-
LVAC-LW	50	1000	10.00	0.1000	1.0000	0.1000	1.00×10^{-3}
	100	1966	14.20	0.1563	0.0731	0.4822	1.402×10^{-3}
	1000	3883	12.35	0.0539	2.5206	0.0143	6.002×10^{-4}
LVAN	50	1504	16.15	0.0381	0.0733	-	-
	100	1966	14.20	0.1563	0.0731	-	-
	1000	3883	12.35	0.0539	2.5206	-	-
LVAN-LW	50	1504	16.15	0.0381	0.0733	0.3897	8.372×10^{-2}
	100	1966	14.20	0.1563	0.0731	0.4822	1.402×10^{-3}
	1000	3883	12.35	0.0539	2.5206	0.0143	6.002×10^{-4}
Ladder	50	1504	16.15	0.0381	-	-	-
	100	1966	14.20	0.1563	-	-	-
	1000	3883	12.35	0.0539	-	-	-
VAT	50	-	-	-	5.0	-	-
	100	-	-	-	5.0	-	-
	1000	-	-	-	2.5	-	-