UNDERSTANDING REGULARIZATION BY VIRTUAL ADVERSARIAL TRAINING, LADDER NETWORKS AND OTHERS

Mudassar Abbas[*], **Jyri Kivinen**[*], **Tapani Raiko**[*]
Department of Computer Science, School of Science, Aalto University,
Espoo, Finland.
firstname.lastname@aalto.fi

ABSTRACT

We study a regularization framework where we feed an original clean data point and a nearby point through a mapping, which is then penalized by the Euclidian distance between the corresponding outputs. The nearby point may be chosen randomly or adversarially. A more general form of this framework has been presented in (Bachman et al., 2014). We relate this framework to many existing regularization methods: It is a stochastic estimate of penalizing the Frobenius norm of the Jacobian of the mapping as in Poggio & Girosi (1990), it generalizes noise regularization (Sietsma & Dow, 1991), and it is a simplification of the canonical regularization term by the ladder networks in Rasmus et al. (2015). We also investigate the connection to virtual adversarial training (VAT) (Miyato et al., 2016) and show how VAT can be interpreted as penalizing the largest eigenvalue of a Fisher information matrix. Our contribution is discovering connections between the studied and other existing regularization methods.

1 INTRODUCTION

Regularization is a commonly used meta-level technique in training neural networks. This paper studies a regularization method, which is an instance of the Pseudo-Ensemble Agreement regularizer (PEA) presented in (Bachman et al., 2014), investigating theoretical connections of the approach to three regularization techniques proposed in the literature. These regularization techniques are the canonical regularizations by penalizing the Jacobian (Poggio & Girosi, 1990), by noise injection (Sietsma & Dow, 1991), by the Ladder network (Rasmus et al., 2015), and by the virtual adversarial training (Miyato et al., 2016).

The structure of the rest of the paper is as follows: In the following section we describe the regularization criteria studied. In section 3 we investigate connections of these regularization techniques theoretically, providing novel theoretical results together connecting all of the regularization schemes. Section 4 concludes and discusses future work.

2 REGULARIZATION FRAMEWORK

Given a set of (N) input examples $\mathbf{X} = \left\{\{\mathbf{x}^{(n)}\}_{n=1}^{N}\right\}$, and mapping $f(\mathbf{x};\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, training happens by minimizing a training criterion $\mathcal{L}\left(\mathbf{X},\boldsymbol{\theta}\right) = \mathbb{E}_{p(\mathbf{x})}(\cdot)$ iteratively, where the expectation $\mathbb{E}_{p(\mathbf{x})}$ is taken over the training data set $\mathbf{X}$.[1] Often a regularization term $\mathcal{R}$ is added to the training criterion in order to help generalize better to unseen data.

The canonical regularization term by the studied approach is given as follows:

$$\mathcal{R}\left(\mathbf{x},\boldsymbol{\theta};\alpha,\epsilon\right) = \alpha\mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\|f(\mathbf{x}) - f(\mathbf{x}+\Delta\mathbf{x})\|^2 = \alpha\mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\|\mathbf{h} - \tilde{\mathbf{h}}\|^2, \tag{1}$$

[*] The authors declare equal contributions.

[1] We leave open the particular task and rest of the neural architecture.

where $\mathbf{h} = f(\mathbf{x};\boldsymbol{\theta})$ denotes the network output given an example $\mathbf{x}$, $\tilde{\mathbf{h}} = f(\mathbf{x}+\Delta\mathbf{x};\boldsymbol{\theta})$ denotes the network output given a perturbed version of the example $\mathbf{x}+\Delta\mathbf{x}$, and α is a positive scalar hyperparameter.

By default, the perturbation is implemented as additive white Gaussian noise $\Delta\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \epsilon^2 \mathrm{I}\right)$, with I denoting the identity-matrix and ϵ is a positive scalar that is used to control the amount of perturbation. We can also consider an adversarial variant where $\Delta\mathbf{x} = \Delta\mathbf{x}_{\text{adv.}} = \arg\max_{\Delta\mathbf{x}:\|\mathbf{x}\|\leq\epsilon} \|\mathbf{h}-\tilde{\mathbf{h}}\|^2$.

3 Connections to other regularization methods

3.1 Pseudo-Ensemble Agreement

The approach can be seen as an instance of the Pseudo-Ensemble Agreement regularization by Bachman et al. (2014), assuming $\mathcal{V}(\mathbf{h},\tilde{\mathbf{h}}) = \|\mathbf{h}-\tilde{\mathbf{h}}\|^2$ in (Bachman et al., 2014, Equation (2)).

3.2 Penalizing the Jacobian

A central method connecting different regularizers here is the penalization of the Jacobian. Poggio & Girosi (1990) proposed to penalize a neural network by the Frobenius norm of the Jacobian $\mathbf{J}$:

$$\mathcal{R}_{\text{Jacobian}} = \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[\|\mathbf{J}\|_F^2\right] = \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[\sum_{k,i}\left(\frac{\partial h_k}{\partial x_i}\right)^2\right]. \tag{2}$$

In the special case of linear f, the Jacobian is constant and the Jacobian penalty can be seen as a generalization of weight decay. Note that the Frobenius norm of the Jacobian penalizes the mapping f directly, rather than through its parameters.

Matsuoka (1992) and Reed et al. (1992) found a connection between noise regularization and penalizing the Jacobian, assuming a regression setting with a quadratic loss. The connection is applicable in our setting, too, as is demonstrated in the following: Assuming that a small perturbation of the form $\Delta\mathbf{x} \sim \mathcal{N}(\mathbf{0},\epsilon^2\mathrm{I})$ is added to every example $\mathbf{x}$, such that every $\tilde{\mathbf{h}} = f(\mathbf{x}+\Delta\mathbf{x};\boldsymbol{\theta})$, approximating $\tilde{\mathbf{h}}$ near $\mathbf{h}$ using a *component-wise* Taylor series expansion yields that for every output dimension index k

$$\tilde{h}_k \approx h_k + \mathbf{J}_{k,:}\Delta\mathbf{x} + \frac{1}{2}(\Delta\mathbf{x})^\top \mathbf{H}^{(k)}\,\Delta\mathbf{x}\,,$$

where $J_{k,i} = \frac{\partial h_k(\mathbf{x})}{\partial x_i}$ is the Jacobian matrix of partial derivatives of h_k w.r.t. x_i, $\mathbf{J}_{k,:}$ denotes its k^{th} row vector, and $\mathbf{H}^{(k)} = \frac{\partial^2 h_k(\mathbf{x})}{\partial x_i \partial x_j}$ is the Hessian matrix. Plugging this into eq. (1) yields that

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[(\tilde{h}_k - h_k)^2\right] \approx \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[(h_k + \mathbf{J}_{k,:}\Delta\mathbf{x} + \frac{1}{2}(\Delta\mathbf{x})^\top\mathbf{H}^{(k)}\Delta\mathbf{x} - h_k)^2\right]\\
&= \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[(\mathbf{J}_{k,:}\Delta\mathbf{x})^2\right] + \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[\mathbf{J}_{k,:}\Delta\mathbf{x}(\Delta\mathbf{x})^\top\mathbf{H}^{(k)}\Delta\mathbf{x}\right] + \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[(\frac{1}{2}(\Delta\mathbf{x})^\top\mathbf{H}^{(k)}\Delta\mathbf{x})^2\right]\\
&\approx \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[(\mathbf{J}_{k,:}\Delta\mathbf{x})^2\right] = \epsilon^2\,\mathbb{E}_{p(\mathbf{x})}\left[\|\mathbf{J}_{k,:}\|^2\right] = \sum_i\left(\frac{\partial h_k}{\partial x_i}\right)^2\epsilon^2,
\end{aligned}$$

where we have used the fact that $\mathbb{E}_{p(\Delta\mathbf{x})}[\Delta\mathbf{x}(\Delta\mathbf{x})^\top] = \epsilon^2\mathrm{I}$ and the assumption that the terms involving the Hessians are very small. Therefore we can obtain that

$$\mathcal{R} = \alpha\mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})}\left[\|\mathbf{h}-\tilde{\mathbf{h}}\|^2\right] \approx \alpha\epsilon^2\mathcal{R}_{\text{Jacobian}}\,,$$

and thus conclude that the studied regularization can be seen as a stochastic approximation of the Jacobian penalty in the limit of low noise.

An example of a recent method penalizing the Jacobian is the contractive auto-encoder (Rifai et al., 2011).

3.3 NOISE REGULARIZATION

Sietsma & Dow (1991) proposed to regularize neural networks by injecting noise to the inputs $\mathbf{x}$ while training. This method does not use a separate penalty function. It can be interpreted either as keeping the solution insensitive to random perturbations in $\mathbf{x}$, or as a data augmentation method, where a number of noisy copies of each data point are used in training.

In the semi-supervised setting, noise regularization alone does not make unlabelled data useful. However, our regularization method is applicable even when the output is not available: We can compute the regularization penalty $\mathcal{R}$ based on the input alone.

In noise regularization, the strength of regularization can only be adjusted by changing the noise level. This affects the locality of the regularization at the same time. The regularizer studied gives two hyperparameters to adjust: Noise level ϵ^2 and the regularization strength α. Using small noise level ϵ^2 emphasizes the local (linear) properties of the mapping f, while a larger noise level emphasizes more global (nonlinear) properties.

3.4 VIRTUAL ADVERSARIAL TRAINING

In this section we will show the following: (i) VAT penalizes the largest eigenvalue of a Fisher information matrix, (ii) in the special case of $\mathbf{h}$ parameterizing the mean of a Gaussian output, the VAT loss which the adversarial perturbation maximization is operating on is equivalent to the studied loss $\mathcal{R}$, and (iii) the adversarial variant of the studied regularization penalizes the largest eigenvalue of $\mathbf{J}^\top\mathbf{J}$, whereas the sum of its eigenvalues equals the Frobenius norm of $\mathbf{J}$.

The virtual adversarial training of Miyato et al. (2016) regularizes training under minimization by

$$\mathcal{R}_{\text{VAT}} = \alpha\left\{-\text{LDS}(\mathbf{x},\boldsymbol{\theta};\epsilon)\right\} = \alpha\left\{\max_{\Delta\mathbf{x}:\|\mathbf{x}\|\le\epsilon} \text{D}_{\text{KL}}\left(p(\mathbf{y}\mid\mathbf{x},\Phi)\|p(\mathbf{y}\mid\mathbf{x}+\Delta\mathbf{x},\Phi)\right)\right\},$$

a positive scalar α times the maximal Kullback-Leibler divergence between encoding distributions for an example $\mathbf{x}$ and its perturbed version $\mathbf{x}+\Delta\mathbf{x}$, denoted $p(\mathbf{y}\mid\mathbf{x},\Phi)$ and $p(\mathbf{y}\mid\mathbf{x}+\Delta\mathbf{x},\Phi)$, respectively; the distributions are assumed to be of the same parametric form, and that the perturbation affects the values of the parameters. Using a note in Kaski et al. (2001, eq. (1)-(2)), and a result in Kullback (1959, Sec. 6, page 28, eq. (6.4)), we can state:

$$\text{D}_{\text{KL}}(p(\mathbf{y}\mid\mathbf{x},\Phi)\|p(\mathbf{y}\mid\mathbf{x}+\Delta\mathbf{x},\Phi)) \approx \frac{1}{2}(\Delta\mathbf{x})^\top\mathcal{I}(\mathbf{x})\Delta\mathbf{x}, \tag{3}$$

with $\mathcal{I}(\mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x},\Phi)}\left(\frac{\partial\log p(\mathbf{y}|\mathbf{x},\Phi)}{\partial\mathbf{x}}\right)\left(\frac{\partial\log p(\mathbf{y}|\mathbf{x},\Phi)}{\partial\mathbf{x}}\right)^\top$, a Fisher-information matrix. Let $\widehat{\Delta\mathbf{x}} = \frac{\Delta\mathbf{x}}{\|\Delta\mathbf{x}\|}$, a unit-length input-perturbation vector. Using the approximation above we can then write that

$$\mathcal{R}_{\text{VAT}} \approx \frac{\alpha}{2}\max_{\Delta\mathbf{x}:\|\Delta\mathbf{x}\|\le\epsilon}(\Delta\text{x})^\top\mathcal{I}(\mathbf{x})\Delta\mathbf{x} = \frac{\alpha}{2}\max_{\widehat{\Delta\mathbf{x}}\|\Delta\mathbf{x}\|:\|\Delta\mathbf{x}\|\le\epsilon}\|\Delta\mathbf{x}\|^2\left[\frac{\left(\widehat{\Delta\mathbf{x}}\right)^\top\mathcal{I}(\mathbf{x})\widehat{\Delta\mathbf{x}}}{\|\widehat{\Delta\mathbf{x}}\|^2}\right] \tag{4}$$

$$=\frac{\alpha}{2}\underbrace{\max_{\|\Delta\mathbf{x}\|:\|\Delta\mathbf{x}\|\le\epsilon}\|\Delta\mathbf{x}\|^2}_{\epsilon^2}\max_{\widehat{\Delta\mathbf{x}}}\left[\frac{\left(\widehat{\Delta\mathbf{x}}\right)^\top\mathcal{I}(\mathbf{x})\widehat{\Delta\mathbf{x}}}{\|\widehat{\Delta\mathbf{x}}\|^2}\right] =^* \frac{\alpha\epsilon^2}{2}\max\left\{\lambda_k\right\}_{k=1}^K = \frac{\alpha\epsilon^2}{2}\lambda_{\text{max}},$$

where λ_{max} denotes the maximal eigenvalue of the (K) eigenvalues $\{\lambda_k\}_{k=1}^K$ of $\mathcal{I}(\mathbf{x})$, with the equality marked $=^*$ suggested by a spectral theory for symmetric matrices printed in Råde & Westergren (1998, Sec. 4.5, page 96); $\text{diag}\left(\{\lambda_k\}_{k=1}^K\right) = \mathbf{C}^\top\mathcal{I}(\mathbf{x})\mathbf{C}$, where $\mathbf{C}$ denotes the matrix of eigenvectors with $\mathbf{C}_{\cdot,k}$ denoting its k^{th} column and k^{th} eigenvector; the adversarial perturbation vector $\Delta\mathbf{x}_{\text{adv.}} = \frac{\epsilon^2}{2}\mathbf{C}_{\cdot,\arg\max_k\{\lambda_k\}_{k=1}^K}$.

Let us assume $\mathbf{y} = f(\mathbf{x};\boldsymbol{\theta}) + \mathbf{n}$, where $\mathbf{n}\sim\mathcal{N}(\mathbf{0},\sigma^2\text{I})$. We can then write that

$$p(\mathbf{y}\mid\mathbf{x},\Phi) = \mathcal{N}\left(\mathbf{y};\mathbf{h},\sigma^2\text{I}\right), \qquad p(\mathbf{y}\mid\mathbf{x}+\Delta\mathbf{x},\Phi) = \mathcal{N}\left(\mathbf{y};\tilde{\mathbf{h}},\sigma^2\text{I}\right),$$

where $\mathbf{h} = f(\mathbf{x}; \boldsymbol{\theta})$, $\tilde{\mathbf{h}} = f(\mathbf{x} + \Delta\mathbf{x}; \boldsymbol{\theta})$, $\Phi = \{\boldsymbol{\theta}, \sigma\}$, and I denotes the identity-matrix. Using Kullback (1959, Chapter 9, Sec. 1, page 190, eq. (1.4)) (and the derivation in the Appendix), we have that

$$\mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\mathbf{y}; \mathbf{h}, \sigma^2 \mathrm{I}\right) \mid\mid \mathcal{N}\left(\mathbf{y}; \tilde{\mathbf{h}}, \sigma^2 \mathrm{I}\right)\right) = \frac{1}{2\sigma^2} \sum_{k=1}^{K} (h_k - \tilde{h_k})^2 = \frac{1}{2\sigma^2} \left(\mathbf{h} - \tilde{\mathbf{h}}\right)^\top \left(\mathbf{h} - \tilde{\mathbf{h}}\right).$$

For small perturbations $\Delta\mathbf{x}$ the following linearization approximation is expected to be effective for non-linear functions f[2]:

$$\mathbf{h} - \tilde{\mathbf{h}} = f(\mathbf{x}; \boldsymbol{\theta}) - f(\mathbf{x} + \Delta\mathbf{x}; \boldsymbol{\theta}) \simeq \mathbf{J}\Delta\mathbf{x},$$

where $\mathbf{J}$ denotes a Jacobian-matrix, with element $J_{k,i} = \frac{\partial f_k(\mathbf{x})}{\partial x_i}$, where $k \in [1, K]$, and $i \in [1, I]$. Then given the prior analysis, we have for the Gaussian models that

$$\mathcal{R}_{\text{VAT}} \simeq \frac{\alpha}{2\sigma^2} \max_{\Delta\mathbf{x}: \|\Delta\mathbf{x}\| \leq \epsilon} (\Delta\mathbf{x})^\top \mathbf{J}^\top \mathbf{J} \Delta\mathbf{x} = \frac{\alpha\epsilon^2}{2\sigma^2} \lambda_{\text{max}},$$

where λ_{max} denotes the maximal eigenvalue of the (K) eigenvalues $\{\lambda_k\}_{k=1}^K$ of $\mathbf{J}^\top\mathbf{J}$; we have used the fact that the optimization problem is exactly the same as in eq. (4), with the replacement of $\mathcal{I}(\mathbf{x})$ with $\mathbf{J}^\top\mathbf{J}$.

Since

$$\sum_{k=1}^{K} \lambda_k = \mathrm{Trace}\left(\mathbf{J}^\top \mathbf{J}\right) = \sum_k \sum_i J_{k,i}^2 = \sum_k \sum_i \left(\frac{\partial f_k(\mathbf{x})}{\partial x_i}\right)^2, \tag{5}$$

where the first equality is suggested by a fact in Råde & Westergren (1998, Sec. 4.5, page 96), we find that $\mathcal{R}_{\text{Jacobian}} = \alpha \sum_k \sum_i J_{k,i}^2 = \alpha \sum_k \sum_i \left(\frac{\partial f_k(\mathbf{x})}{\partial x_i}\right)^2 = \alpha \sum_{k=1}^{K} \lambda_k$, is also penalizing the eigenvalues of $\mathbf{J}^\top\mathbf{J}$, penalizing their sum as opposed to their maximum value by the VAT.

3.5 Ladder networks

The Γ variant of the ladder network (Rasmus et al., 2015) is closely related to the studied regularization approach. It uses an auxiliary denoised $\hat{\mathbf{h}} = g(\tilde{\mathbf{h}})$ and an auxiliary cost $\mathcal{R}_\Gamma = \mathbb{E}_{p(\mathbf{x},\Delta\mathbf{x})} \|\mathbf{h} - \hat{\mathbf{h}}\|^2$ to support another task such as classification. It means that if we would train a Γ-ladder network and further restrict the g-function to identity, we would recover the studied regularization method.

4 Future work

We would like to investigate the studied regularization method and the found connections empirically. Miyato et al. (2016) used the D_{KL} in equation (3) as a regularization criterion with adversarial perturbations, but it could also be used for regularization using random perturbations.

Acknowledgments

We thank Casper Kaae Sønderby, Jaakko Peltonen, and Jelena Luketina for useful comments related to the manuscript/work. We also thank the anonymous reviewers for information about relevant literature, which we have used in improving the document.

A KL-divergence between two diagonal-covariance multivariate Gaussians

We first derive the KL-divergence between two diagonal-covariance multivariate Gaussians, with untied parameters. We then use this result for the case where the covariance matrices are tied and the elements in the diagonal have the same values. Starting with the first derivation: Assume

[2] It is exact for linear functions f.

that both $\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right)$ and $\mathcal{N}\left(\mathbf{y} \mid \Theta_2\right)$ denote (K-dimensional) multivariate Gaussians assuming parameters Θ_1 and Θ_2, respectively, and with independent dimensions as follows: $\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right) = \prod_{k=1}^{K} \mathcal{N}(y_k; \mu_k, \sigma_k^2), \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right) = \prod_{k=1}^{K} \mathcal{N}(y_k; m_k, \tau_k^2)$, where $\mathcal{N}(y_k; \mu_k, \sigma_k^2)$ denotes a univariate Gaussian with mean μ_k and variance σ_k^2, and $\mathcal{N}(y_k; m_k, \tau_k^2)$ denotes a univariate Gaussian with mean m_k and variance τ_k^2.

We use a ”divided-and-then-joined” calculation strategy to give the result, similar to as in the presentation of the result in Kingma & Welling (2014, Appendix B):

$$\mathrm{D_{KL}}\left(\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right) || \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right)\right) = \underbrace{\mathbb{E}_{\mathbf{y} \sim \mathcal{N}(\mathbf{y} \mid \Theta_1)} \log \mathcal{N}\left(\mathbf{y} \mid \Theta_1\right)}_{A} - \underbrace{\mathbb{E}_{\mathbf{y} \sim \mathcal{N}(\mathbf{y} \mid \Theta_1)} \log \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right)}_{B}.$$

Let us now derive the forms of A and B above:

$$\begin{aligned}
A &= \mathbb{E}_{\mathbf{y} \sim \mathcal{N}(\mathbf{y} \mid \Theta_1)} \left\{ -\frac{1}{2} \sum_{k=1}^{K} \left[\log 2\pi + 2 \log \sigma_k + (y_k - \mu_k)^2 / \sigma_k^2 \right] \right\} \\
&= -\frac{1}{2} \sum_{k=1}^{K} \left[\log 2\pi + 2 \log \sigma_k + \frac{1}{\sigma_k^2} \underbrace{\mathbb{E}_{y_k \sim \mathcal{N}(y_k \mid \Theta_1)} \left\{ (y_k - \mu_k)^2 \right\}}_{\sigma_k^2} \right], \\
B &= \mathbb{E}_{\mathbf{y} \sim \mathcal{N}(\mathbf{y} \mid \Theta_1)} \left\{ -\frac{1}{2} \sum_{k=1}^{K} \left[\log 2\pi + 2 \log \tau_k + (y_k - m_k)^2 / \tau_k^2 \right] \right\} \\
&= -\frac{1}{2} \sum_{k=1}^{K} \left[\log 2\pi + 2 \log \tau_k + \frac{1}{\tau_k^2} \mathbb{E}_{y_k \sim \mathcal{N}(y_k \mid \Theta_1)} \left\{ y_k^2 - 2 m_k y_k + m_k^2 \right\} \right] \\
&= -\frac{1}{2} \sum_{k=1}^{K} \left[\log 2\pi + 2 \log \tau_k + \frac{1}{\tau_k^2} \left\{ \sigma_k^2 + \mu_k^2 - 2 m_k \mu_k + m_k^2 \right\} \right],
\end{aligned}$$

where we have used the fact that $\mathbb{E}_{y_k \sim \mathcal{N}(y_k \mid \Theta_1)} \left\{ y_k^2 \right\} = \sigma_k^2 + \mu_k^2$; $\mathbb{E}(y_k^2) \triangleq \mathrm{Var}(y_k) + \left[\mathbb{E}(y_k)\right]^2$. Joining the results, we have that

$$\mathrm{D_{KL}}\left(\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right) || \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right)\right) = -\frac{1}{2} \sum_{k=1}^{K} \left[1 + 2 \log \frac{\sigma_k}{\tau_k} - \frac{1}{\tau_k^2} \left\{ \sigma_k^2 + (\mu_k - m_k)^2 \right\} \right]. \tag{6}$$

Let $\sigma_k = \tau_k = \sigma$, $\forall\, k$, $\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right) = \prod_{k=1}^{K} \mathcal{N}(y_k; \mu_k, \sigma^2), \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right) = \prod_{k=1}^{K} \mathcal{N}(y_k; m_k, \sigma^2)$. Then using the result in eq. (6), we have that

$$\mathrm{D_{KL}}\left(\mathcal{N}\left(\mathbf{y} \mid \Theta_1\right) || \mathcal{N}\left(\mathbf{y} \mid \Theta_2\right)\right) = \frac{1}{2\sigma^2} \sum_{k=1}^{K} (\mu_k - m_k)^2 = \frac{1}{2\sigma^2} \left(\mu - \mathbf{m}\right)^{\top} \left(\mu - \mathbf{m}\right). \tag{7}$$

The result can also be obtained by using the result in Kullback (1959, Chapter 9, Sec. 1, page 190, eq. (1.4)) which states that $\mathrm{D_{KL}}\left(\mathcal{N}\left(\mathbf{y}; \mu, \Lambda^{-1}\right) || \mathcal{N}\left(\mathbf{y}; \mathbf{m}, \Lambda^{-1}\right)\right) = \frac{1}{2} (\mu - \mathbf{m})^{\top} \Lambda \left(\mu - \mathbf{m}\right)$: plugging in $\Lambda = \left(\sigma^2 \mathrm{I}\right)^{-1} = \sigma^{-2} \mathrm{I}$ yields the result in eq. (7).

REFERENCES

P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. In *Proc., Advances in Neural Information Processing Systems (NIPS)*, pp. 3365–3373. 2014.

S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12(4):936–947, 2001.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. arXiv:1312.6114v10 [stat.ML], 2014.

S. Kullback. *Information Theory and Statistics*. John Wiley& Sons, Inc., 1959.

K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3):436–440, 1992.

T. Miyato, S.-i Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. arXiv:1507.00677v7 [stat.ML], 2016.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9): 1481–1497, 1990.

L. Råde and B. Westergren. *Mathematics Handbook for Science and Engineering*. Studentlitteratur, 4th edition, 1998.

A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-Supervised learning with Ladder Networks. In *Proc., Advances in Neural Information Processing Systems (NIPS)*, pp. 3532–3540. 2015.

R. Reed, S. Oh, and R. J. Marks II. Regularization using jittered training data. In *Proc., International Joint Conference on Neural Networks (IJNN)*, volume 3, pp. 147–152. IEEE, 1992.

S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive Auto-Encoders: Explicit invariance during feature extraction. In *Proc., International Conference on Machine Learning (ICML)*, 2011.

J. Sietsma and R. J. F. Dow. Creating artificial neural networks that generalize. *Neural networks*, 4 (1):67–79, 1991.