WEIGHT-AVERAGED CONSISTENCY TARGETS IMPROVE SEMI-SUPERVISED DEEP LEARNING RESULTS

Antti Tarvainen* **& Harri Valpola**
The Curious AI Company
{tarvaina,harri}@cai.fi

ABSTRACT

The recently proposed temporal ensembling has achieved state-of-the-art results in several semi-supervised learning benchmarks. It maintains an exponential moving average of label predictions on each training example, and penalizes predictions that are inconsistent with this target. However, because the targets change only once per epoch, temporal ensembling becomes unwieldy when using large datasets. To overcome this problem, we propose a method that averages model weights instead of label predictions. As an additional benefit, the method improves test accuracy and enables training with fewer labels than earlier methods. We report state-of-the-art results on semi-supervised SVHN, reducing the error rate from 5.12% to 4.41% with 500 labels, and achieving 5.39% error rate with 250 labels.

1 INTRODUCTION

Deep learning is a powerful approach for machine learning in areas such as image and speech recognition. In order to learn useful abstractions, deep learning models require a large number of parameters, thus making them prone to over-fitting. Moreover, adding high-quality labels to training data manually is often expensive. Therefore, it is desirable to use regularization methods that exploit unlabeled data effectively to reduce over-fitting in semi-supervised learning.

When a percept is changed slightly, a human typically still considers it to be the same object. Correspondingly, a classification model should favor functions which give a consistent output for similar data points. One approach for achieving this is to add noise to the input of the model. To enable the model to learn more abstract invariances, the noise may be added to intermediate representations, an insight that has motivated many regularization techniques, such as Dropout (Srivastava et al., 2014) and Adversarial Training (Goodfellow et al., 2014). Rather than minimizing the classification cost at the zero-dimensional data points of the input space, the regularized model minimizes the cost on a manifold around each data point, thus pushing decision boundaries away from the labeled data points (Figure 1a).

Since classification cost is undefined for unlabeled examples, the noise regularization by itself does not aid in semi-supervised learning. To overcome this, techniques such as Virtual Adversarial Training (Miyato et al., 2015) and Γ model (Rasmus et al., 2015) evaluate each data point with noise and without noise, and then apply a *consistency cost* between the two predictions. In this case, the model assumes a dual role as a *teacher* and a *student*. As a student, it learns as before; as a teacher, it generates targets, which are then used by itself as a student for learning. Since the targets are inevitably biased towards the model itself, it is important to balance these two roles carefully. If too much weight is given to the generated targets, the cost of inconsistency outweighs the cost of misclassification, preventing the learning of new information. In effect, the model suffers from confirmation bias (Figure 1b). This hazard can be mitigated by improving the quality of the targets.

In general, the softmax output of a model does not provide a good Bayesian approximation outside training data. This can be alleviated by adding noise to the model (Gal & Ghahramani, 2016). Consequently, a noisy teacher can yield targets with smaller bias (Figure 1c). This approach has been used recently in Π model (Laine & Aila, 2016) and a similar technique by (Sajjadi et al., 2016).

*also Aalto University

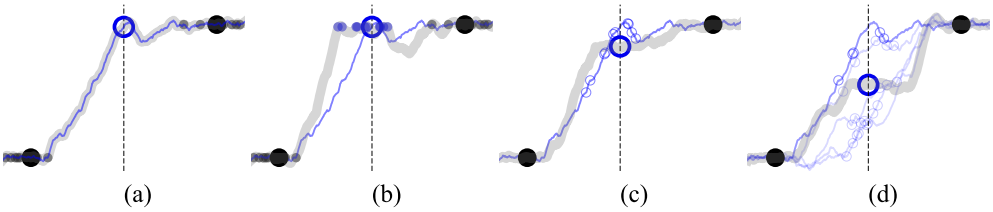

Figure 1: A sketch with two labeled examples (large black dots) and one unlabeled example (vertical line), demonstrating how the choice of unlabeled targets (blue circles) affects the fitted function (gray curve). For the clarity of illustration, we consider a regression task, although consistency regularization may be more suited for classification tasks. **(a)** A model trained with noisy labeled data (small dots) learns to give consistent predictions around labeled data points. **(b)** Consistency to noise around unlabeled examples provides additional smoothing. For the clarity of illustration, the teacher model (blue curve) is first fitted to the labeled examples, and then left unchanged during the training of the student model. Also for clarity, we will omit the small dots in figures c and d. **(c)** Noise on the teacher model reduces the bias and increases the variance of the targets. The expected direction of stochastic gradient descent is towards the mean (large blue circle) of individual noisy targets (small blue circles). **(d)** An ensemble of models gives an even better expected target.

They have reported significant improvements over previous state-of-the-art methods on several semi-supervised benchmarks. However, this approach increases the variance of the generated targets, limiting its usefulness.

To reduce the variance, Laine & Aila (2016) have proposed temporal ensembling. With this method, targets are computed from exponential moving average (EMA) of model predictions over epochs. Since the targets are now formed with an implicit ensemble of models, they should have smaller variance (Figure 1d). However, since temporal ensembling updates targets only once per epoch, the learned information is fed back to the training process at a slow pace. The larger the dataset, the longer the span of the updates, and in the case of on-line learning, it is unclear how temporal ensembling can be used at all.

2 WEIGHT-AVERAGED CONSISTENCY TARGETS

To overcome the limitations of temporal ensembling, we propose *weight-averaged consistency targets*. Averaging model weights over training steps tends to give better validation cost than using final weights directly (Polyak & Juditsky, 1992). We can take advantage of this during training to form better targets. Instead of sharing the weights with the student model, the teacher model uses the EMA weights of the student model. Now it can aggregate information after every step instead of every epoch. In addition, since the weight averages improve all layer outputs, not just the top output, the target model has better intermediate representations. These aspects lead to two practical advantages over temporal ensembling: First, the more accurate target labels lead to a positive feedback loop between learning and target models, resulting in better test accuracy. Second, the approach scales to large datasets and online learning.

We define consistency cost J as the expected distance between the prediction of the student model with weights θ and the expected prediction of the teacher model with weights θ', where noise η is applied to the models separately.

$$J(\theta) = \mathbb{E}_{x,\eta}\left[\left\|\mathbb{E}_{\eta'}\left[f(x,\theta',\eta')\right] - f(x,\theta,\eta)\right\|^2\right]$$

With regards to optimization, the teacher model parameters θ' are treated as constants. Similarly to Laine & Aila (2016), we use mean square error as the distance metric.[1] Noise can take many forms. In our experiments, we will apply three types of noise: random translations of input images, Gaussian noise on the input layer, and dropout applied within the network.

[1] We also ran experiments with cross-entropy but saw no improvement in results.

We can approximate the cost in stochastic gradient descent by sampling noise η at each training step. Whereas the Π model uses $\theta' = \theta$, and temporal ensembling approximates good $f(x, \theta', \eta)$ with a weighted average of successive predictions, we define θ'_t at training step t as the EMA of successive θ weights:

$$\theta'_t = \alpha\theta'_{t-1} + (1 - \alpha)\theta_t$$

where α is a smoothing coefficient hyperparameter.

3 EXPERIMENTS

To test our hypothesis, we first replicated the Π model of Laine & Aila (2016) in TensorFlow (Abadi et al., 2015), except we used batch normalization instead of weight normalization. Probably because of this change, our results on the original Π model were slightly worse than the reported results. We then changed the model to use weight-averaged consistency targets[2]. We retained the Π model hyperparameters and applied $\alpha = 0.999$ as the EMA coefficient for teacher weight updates.

We ran experiments on SVHN (Netzer et al., 2011) dataset, using the 73257 examples of its primary training set, and excluding the additional 531131 training examples of the extra training set. In the development phase of our work, we separated 10% of the training data into a validation set. We removed most of the labels from the remaining training data and used that for training; we retained labels in the validation set to enable exploration of the results. In the final evaluation phase we used the entire training set, including the validation set but with labels removed.[3]

From Table 1 we can see that the use of weight-averaged consistency targets improves the test accuracy over earlier methods on semi-supervised tasks.

Table 1: Error rate percentage on SVHN over 10 runs. Unlike temporal ensembling (TE), weight-averaged consistency (WAC) is applicable to on-line learning.

Model	On-line compatible	250 labels	500 labels	1000 labels	All labels[a]
Supervised-only	yes	42.65 ± 2.68	22.08 ± 0.73	14.46 ± 0.71	2.81 ± 0.07
Improved GAN[b]	yes		18.44 ± 4.8	8.11 ± 1.3	
Π model[c]	yes		6.65 ± 0.53	4.82 ± 0.17	$\mathbf{2.54 \pm 0.04}$
TE[c]	no		5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
WAC	yes	$\mathbf{5.39 \pm 0.39}$	$\mathbf{4.41 \pm 0.26}$	$\mathbf{4.02 \pm 0.19}$	2.64 ± 0.05

[a] 4 runs [b] Salimans et al. (2016) [c] Laine & Aila (2016)

4 ACKNOWLEDGEMENTS

We thank Samuli Laine and Timo Aila for interesting and fruitful discussions about their work.

[2] We will publish the TensorFlow source code of our model in the near future.

[3] On a real-world use case we would not possess a fully-labeled validation set. However, this setup is useful in a research setting, since it enables a more thorough analysis of the results. To the best of our knowledge, this is the common practice when carrying out research on semi-supervised learning. By retaining the hyper-parameters from previous work where possible we decreased the chance of over-fitting our results to validation labels.

REFERENCES

Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. December 2014. arXiv: 1412.6572.

Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. *arXiv:1610.02242 [cs]*, October 2016. arXiv: 1610.02242.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

B. T. Polyak and A. B. Juditsky. Acceleration of Stochastic Approximation by Averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992. ISSN 0363-0129. doi: 10.1137/0330046.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised Learning with Ladder Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3546–3554. Curran Associates, Inc., 2015.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1163–1171. Curran Associates, Inc., 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2226–2234, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15 (1):1929–1958, January 2014. ISSN 1532-4435.