FEATURE-BASED AUGMENTATION FOR SEMI-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose a feature-based augmentation, a simple and efficient method for semi-supervised learning, where only a small part of the data is labeled. In semi-supervised learning, input image augmentation is typically known to be a technique for ensuring generalization of unlabeled data. However, unlike general input augmentation(translation, filp, Gaussian noise, etc.), our method adds noise to features that have the most contribution on prediction, generating an augmented features. We call this method "Feature-based augmentation" because the noise is determined by the network weight itself and augmentation is carried out at the feature level. A prediction by augmented features is used as a target for unlabeled data. The target is stable because it is augmented by the noise based on its extracted features. Feature-based augmentation is applied to semi-supervised learning on SVHN, CIFAR-10 datasets. This method achieved a state-of-the-art error rate. In particular, performance differences from other methods were more pronounced with the smaller the number of labeled data.

1 INTRODUCTION

"Deep Learning" has recently achieved tremendous results in areas such as image recognition and speech recognition. A large number of labeled data is an essential element for these tasks. If the architecture of the network is deep and has a large number of parameters, more labeled data is needed because it is likely to be over-fitted easily (Bishop, 2006). However, the labeled data is limited and labeling the data is expensive and requires human-effort. To avoid problems such as over-fitting, noise can be added around the input data to make the model more robust (Goodfellow et al., 2016). Data augmentation is to create new data that has the same input data distribution by transforming the existing data in various way (DeVries & Taylor, 2017). In the previous works of Laine & Aila (2016), Tarvainen & Valpola (2017), and Miyato et al. (2018), they showed impressive results, achieving very low error rates mainly with input data augmentation. Therefore, efficient data augmentation is a good regularization technique that can prevent the model from over-fitting in semi-supervised learning.

To overcome the lack of labeled data, we apply augmentation to higher-level representations, which are derived hierarchically from lower-level representations rather than augmenting directly to lower-level representations. Bengio et al. (2013) claimed that higher-level representations potentially capture relatively higher-level abstractions. Transformation in the manifold around the data point tends to exponentially unfold when represented at higher levels. Indeed, adding such higher-level noise around unlabeled data makes the decision boundaries smooth. Augmentation in the latent space is suitable for regularization as it can produce new data point that is more plausible and comprehensive.

Furthermore, similar phenomenon(like augmentation in the latent space) can be found in biological neurons, which are neuronal activity in the brain has stochastic character at the microscopic level, it called noise (Destexhe, 2012). The neuronal response varies depending on the distance or position it is viewed at the same distance. A single neuron responds differently to a particular input signal. This is the variability of neurons, specifically the variability of the neurotransmitter released from the axon terminal fiber into the synapse (Richard B. Stein, 2005). In the early days, the variability in neurons circuits was thought to have a negative effect on signal transmission, but a recent study found that the variability of neurons improve information processing in complex and non-linear sys-

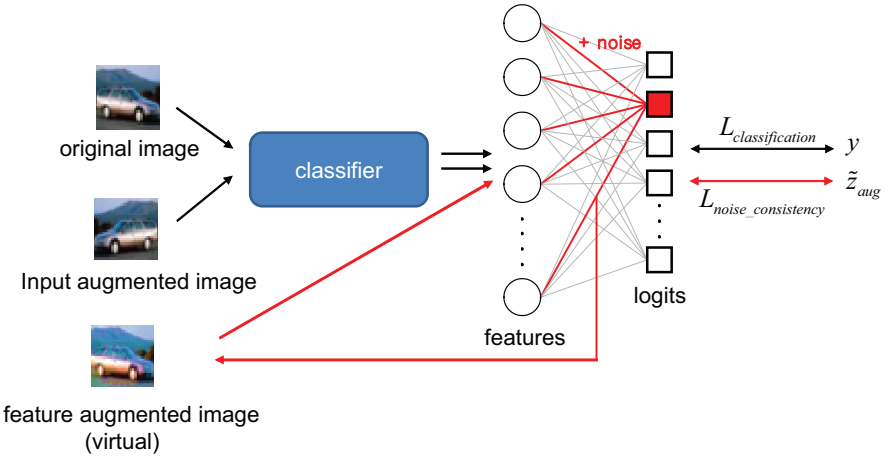

Figure 1: Feature-based augmentation for semi-supervised learning. The figure depicts the flow of labeled data and unlabeled data. We employ the output of the last convolution layer as the features, i.e. the input of the fully connected layer. Random noise is not applied to the layer but to the weights corresponding to label logit, similar to the synaptic variability of biological neurons. Our method adds noise to the weights that contribute the most to classification, so the feature-based augmented image has implicitly the regularization.

tems (Ward, 2011). The variability of neurons has shown that population of neurons automatically represent the probabilistic distribution over the stimulus (Wei Ji Ma, 2006).

Goodfellow et al. (2014b) generated adversarial examples through adversarial training, which assign a label to an input data that is similar to the labels of their neighbors in the adversarial direction. Choosing noise in the latent space should further be very careful and it has a huge impact on the performance of the model because it is so implicit. For that reason, the neural network uses the parameters to determine the amount and direction of noise on its own.

Whereas the Temporal ensembling (Laine & Aila, 2016) and Mean Teacher (Tarvainen & Valpola, 2017) produce stable targets using an exponential moving average (EMA) prediction or model, our method can yield quite stable targets with real-time prediction alone compared with that, so that target update is not necessary. This method is also available for on-line training. VAT (Miyato et al., 2018) adds adversarial perturbations to the input data in a gradient direction so that it can exhibit better generalization than simple random perturbations. And the adversarial perturbations play a role as a regularization term in VAT, but feature-based augmentation has the benefit of reducing the computation costs because it is based on network parameters without having to compute a gradient. It can also generate abstract and implicit augmented images because it is higher-level perturbations unlike input space augmentation.

Our goal is to improve target quality of unlabeled data and to obtain a robust model even if the number of labeled data is smaller. And our method is inspired by biological synaptic noise which improve information processing in complex and non-linear systems.

2 METHOD

2.1 FEATURE-BASED AUGMENTATION

We propose feature-based augmentation for semi-supervised learning. Let N be the number of training data, $\mathcal{L}$ and $\mathcal{U}$ denote the set of labeled data and the set of unlabeled data, respectively. The input images are denoted by x_i, where $i \in \{1, 2, 3...N\}$. For x_i, $y_{i\in\mathcal{L}} \in \{1, 2, 3...C\}$ is a label and $z_{i\in\mathcal{U}} \in \{1, 2, 3...C\}$ is a target of unlabeled data, where C is the number of classes. Also, we employ $f(x)$, $g(x)$ and $h(x)$ as the prediction of neural network with parameter θ, the stochastic input augmentation function (random translation, horizontal flips, Gaussian noise) and the feature of neural network.

Here we select the higher-level feature-vector $\overline{\mathbf{f}} \in R^{128}$ (i.e. the output of the last convolution layer and the input of the fully connected layer) to be applied to augmentation. The parameter of the fully connected layer is θ_{fc}, of size $128 \times C$. Since perturbations at higher-feature level is applied, we call this method feature-based augmentation.

Figure 1 shows the architecture of feature-based augmentation and Algorithm 1 shows the pseudocode. We describe how we can use our method to create a robust target for unlabeled data. The first step is to generate the noise to the features.

$$noise = \theta_{fc} \cdot Randomuniform[-k, k]$$
$$z_{i_{aug}} = f(\overline{\mathbf{f}} + noise),$$

where $Randomuniform[-k, k]$ is randomly uniform sampling from hyperparmeter $-k$ to $k \in [0, 1]$, of same size as $\overline{\mathbf{f}}$ and $z_{i_{aug}}$ is the target(the prediction of the model) with feature-based augmentation. At the equation above, noise is a slightly scaled value up to k times the θ_{fc}. The higher the parameter value, the more likely the noise will be. However, because θ_{fc} is about the relationship between the feature and the logit, the greater the value of the parameter corresponding to the feature element, the greater the feature value contributes to the logit.

In other words, noise for feature-based augmentation is generated by adding the most likely value to the element that contributes the most to the prediction among the features being learned. It is natural to think that applying appropriate manipulation to the feature that contributes the most is effective in learning. This feature-based augmentation is also similar to the variability of neurotransmitters released by biological neurons into the synapses for the specific input signal. Conversely inducing other similar input signals(little different) to be recognized in the same class. Thus, without having to calculate a gradient, the feature that has the greatest effect on the error rates can be identified, and this operation can have the effect of pushing the labeling data from the decision boundaries.It adds noise based on weight value, it has the effect of sharpening logits more. The feature-augmented image has implicitly the regularization effect to generate a robust target $z_{i_{aug}}$. Depending on the k value, the degree of augmentation can be adjusted, which greatly affects network performance.

The second step is to average the target over input images with and without augmentation to make the target more stable. Temporal ensembling generates the mean target value generated according to the time axis, but and our mean target value $\widetilde{z}_i$ is generated by various kinds of augmentation at the same time. In order words, $\widetilde{z}_i$ is mean target of images without input augmentation and images with feature augmentation. In addition to generating targets for unlabeled data, the same can be done for labeled data to lead to more stable learning. Various feature-based augmentations can be made to improve the quality of the mean target, $\widetilde{z}_i$. Unlike Temporal ensembling, it can be called "parallel ensembling".

$$\widetilde{z}_i = Average(z_i, z_{i_{aug}})$$
$$where\ z_i = f_\theta(x_{i \in \mathcal{L} \cap \mathcal{U}})$$

2.2 Objective function

We define the objective function L_{total}, which utilizes the target $\widetilde{z}_i$ for unlabeled data with the feature-based augmentation method described in Section 2.1. For semi-supervised learning, Our method produces the "guess" labels for unlabeled data $\mathcal{U}$ with feature augmentation and compute them to the objective function L_{total}, which consists of

$$L_{classification} = -\sum_{x \in \mathcal{L}} y \log f_\theta(g(x))$$
$$L_{consistency} = d(\sum_{x_i \in \mathcal{L}} h_\theta(g(x_i)) - \sum_{x_j \in \mathcal{U}} h_\theta(g(x_j)))$$
$$L_{noise\ consistency} = D_{KL}[p(f_\theta(g(x)) \mid x \in \mathcal{U}, \theta) \parallel p(\widetilde{z}_i \mid x \in \mathcal{U}, \theta)]$$

$$L_{total} = L_{classification} + \lambda_1 L_{consistency} + \lambda_2 L_{noise\ consistency}$$

where $L_{classification}$ is cross-entropy loss among the labeled dataset $\mathcal{L}$, $L_{consistency}$ is mean squared error(MSE) loss among the features that correspond to labeled $\mathcal{L}$ and unlabeled dataset $\mathcal{U}$

Algorithm 1: Feature-based Augmentation pseudocode.

Require: $\mathcal{L}$ = set of training labeled data
Require: $\mathcal{U}$ = set of training unlabeled data
Require: x_i = training input image
Require: y_i = label of labeled data $i \in \mathcal{L}$
Require: $f_\theta(x)$ = prediction of neural network with parameter θ
Require: $h_\theta(x)$ = features of neural network with parameter θ
Require: $g(x)$ = stochastic input augmentation function
Require: $Aug(x)$ = feature-based augmentation function
for $e \leftarrow 1$ **to** $total\ epochs$ **do**
  for $b \leftarrow 1$ **to** $total\ minibatch, B$ **do**
    $z_i \leftarrow f_\theta(x_{i \in \mathcal{U} \cap B})$
    $z_{i_{aug}} \leftarrow f_\theta(Aug(g(x_{i \in \mathcal{U} \cap B})))$ ▷ with feature-based augmentation
    $\widetilde{z}_i \leftarrow Average(z_i, z_{i_{aug}})$ ▷ prediction target for unlabeled data
    $loss \leftarrow H(y_i, f_\theta(g(x_{i \in \mathcal{L} \cap B})))$ ▷ cross-entropy classification loss
      $+ \lambda_1 \cdot d(h_\theta(g(x_{i \in \mathcal{L} \cap B})), h_\theta(g(x_{i \in \mathcal{U} \cap B})))$ ▷ consistency loss
      $+ \lambda_2 \cdot D[f_\theta(g(x_{i \in \mathcal{U} \cap B})), z_{i_{faug}}]$ ▷ noise consistency loss
    update θ
  end
end
return θ

respectively and $L_{noise\ consistency}$ represents the expected distance between the targets(with input augmentation) and the augmented targets(with feature-based augmentation) using KL-divergence in the unlabeled dataset $\mathcal{U}$. And λ_1 and λ_2 are scaling factors of $L_{consistency}$ and $L_{noise\ consistency}$, respectively.

In our works, MSE is used for d and D_{KL} is KL-divergence[1]. For semi-supervised learning, the amount of labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$ within a mini-batch is not usually the same(sometimes the same depending on the model's design) because the number of unlabeled data is much larger than the number of labeled data. For this reason, when $L_{consistency}$ is calculated, the distance can be obtained between the average feature value of labeled $\mathcal{L}$ and unlabeled data $\mathcal{U}$ with the same class value.

3 EXPERIMENTS

We conducted an experiment using two datasets to evaluate the performance of our methods. We tested the semi-supervised learning for SVHN and CIFAR-10 benchmarks and used the model with a 13-layer convolutional neural network (ConvNet) just like the previous works (Tarvainen & Valpola, 2017; Laine & Aila, 2016; Miyato et al., 2018) with three types of input noise: random translation, horizontal flips, Gaussian noise. Also we applied feature-based augmentation. Our model was trained under TensorFlow framework(Abadi et al., 2015) environment.

We ran 10 experiments on each case. We used dropout (Srivastava et al., 2014; Gal & Ghahramani, 2016) and mean-only batch normalization (Salimans & Kingma, 2016) as regularization, and updated the network parameters using Adam optimizer (Kingma & Ba, 2014). In addition, we did not use the ramp-up, ramp-down function applied to the scaling factor $\lambda 1$, λ_2 of L and learning rate. More details about model architectures, hyperparameters, and etc., are described in the Appendix.

3.1 BASELINE

As the baseline of comparison, the methods of supervised-only learning, Π-model (Tarvainen & Valpola, 2017), Temporal Ensembling (Laine & Aila, 2016), Mean Teacher (Tarvainen & Valpola, 2017), and Virtual Adversarial Training (Miyato et al., 2018) were considered. No techniques for

[1]KL-divergence is empirically more stable than cross-entropy.

further regularization (SNTG, SWA, etc.) were used (Luo et al., 2018; Athiwaratkun et al., 2018; Li et al., 2019). The other methods as the baseline are used the dropout probability $p = 0.5$, but our method is set to dropout probability $p = 0.8$ because feature-based augmentation inherently has its own regularization.

3.2 SVHN

Table 1: SVHN semi-supervised error rates with 13-layer CNN architecture over 10 runs(4 runs when using all labels). See table Appendix

	250 labels 73257 images	500 labels 73257 images	1000 labels 73257 images
Supervised-only (Tarvainen & Valpola, 2017)	27.77 ± 3.18	16.88 ± 1.30	12.32 ± 0.95
Π-model (Tarvainen & Valpola, 2017)	9.69 ± 0.92	6.83 ± 0.66	4.95 ± 0.26
Temporal Ensembling (Laine & Aila, 2016)		5.12 ± 0.13	4.42 ± 0.16
Mean Teacher (Tarvainen & Valpola, 2017)	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19
VAT (Miyato et al., 2018)			5.42 ± 0.22
VAT + EntMin (Miyato et al., 2018)			$\mathbf{3.86 \pm 0.11}$
This work	$\mathbf{4.25 \pm 0.25}$	$\mathbf{4.12 \pm 0.16}$	3.92 ± 0.28

The Street View House Numbers (SVHN) (Netzer et al., 2011) dataset is a $32 \times 32 \times 3$ RGB real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. The dataset consists of 73257 training images and 10 classes for each digits and 26032 test images. SVHN is obtained from house numbers in Google Street View images. Using SVHN dataset, semi-supervised learning was conducted and Table 1 shows the results compared to recent state-of-the-art-methods. We evaluate error rates with a varying number of labels from 250 to 1000. Our method obtains slightly better error rates for both 250 and 500 labels (4.25% and 4.12%, respectively) compared to the 4.18% reported by Tarvainen & Valpola (2017) for 500 labels. Because our method is feature-based augmentation, which can hold more abstract meaning at the higher-level, we can see that the smaller the number of data labels, the more effective it is. Interestingly, when the number of data is smaller, we confirm that our method is more efficient for data augmentation.

3.3 CIFAR-10

Table 2: CIFAR-10 semi-supervised error rates over 10 runs. The whole hyperparameter and experimental setup is in Appendix.

	1000 labels 50000 images	2000 labels 50000 images	4000 labels 50000 images
Supervised-only (Tarvainen & Valpola, 2017)	46.43 ± 1.21	33.94 ± 0.73	20.66 ± 0.57
Π-model (Tarvainen & Valpola, 2017)	27.36 ± 1.20	18.02 ± 0.60	13.20 ± 0.27
Temporal Ensembling (Laine & Aila, 2016)			12.16 ± 0.31
Mean Teacher (Tarvainen & Valpola, 2017)	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.28
VAT (Miyato et al., 2018)			11.36 ± 0.34
VAT + EntMin (Miyato et al., 2018)			$\mathbf{10.55 \pm 0.05}$
This work	$\mathbf{19.45 \pm 1.02}$	$\mathbf{14.69 \pm 0.52}$	11.34 ± 0.27

The CIFAR-10 dataset consists of $32 \times 32 \times 3$ RGB 60000 color images in 10 classes, with 6000 images per class. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 50000 training images and 10000 test images (Krizhevsky, 2009). In CIFAR-10, similarly, we used our method to perform tasks, and input augmentation was applied. We evaluate error rates with a varying number of labels from 1000 to 4000. The results from Table 2 show that although the recently released VAT + EntMin (Miyato et al., 2018) shows somewhat better performance than our method in 4000 labels, our method shows an error rate of 11.34%, which is better than the 11.36% error rate when only the VAT method without

additional regularization term such as entropy minimization. Our method obtains better error rates for both 1000 and 2000 labels (19.45% and 14.69%, respectively) compared to the 15.73% reported by Tarvainen & Valpola (2017) for 2000 labels. In addition, as with SVHN, we could see that the smaller the number of labeling data, the higher the performance improvement. We achieved the state-of-the-art with no techniques for further regularization terms.

3.4 THE EFFECT OF FEATURE-BASED AUGMENTATION ON THE INPUT IMAGES

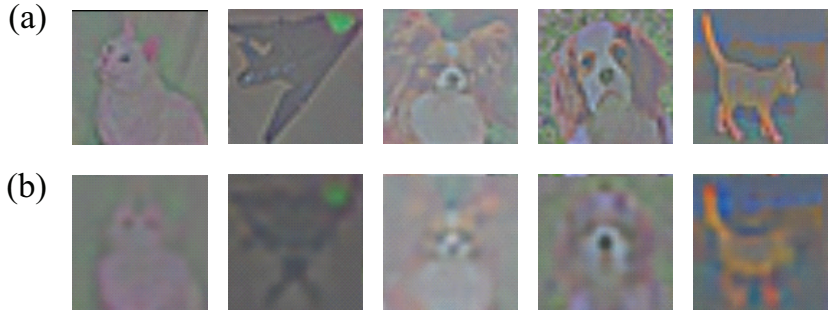

Figure 2: The effect of feature-based augmentation on the input images. We reconstruct the noise-added image using autoencoder(Appendix). It was verified that higher-level augmentation occurred, such as a change in direction of the animal's head, creating a new wing shape of the airplane, etc. All of input images were preprocessed by ZCA. Although the restored images are somewhat blurred because the decoder of the autoencoder is fairly shallow compared with the encoder, the augmentation can be clearly observed. (a) Reconstructed original image without any augmentation. (b) Reconstructed image with augmentation at the feature level.

To verify how feature-based augmentation actually affects the input images, we used the autoencoder. Generative adversarial networks (Goodfellow et al., 2014a), variational autoencoders (Kingma & Welling, 2013) are also used in this context to extract useful high-level features. The encoder part of the autoencoder is the same as the 13-layer CNN structure used in Section 3, and the input of decoder that restores the image is the feature we add noise to, and the decoder structure is the 4-layer transposed convolution architecture. (See Appendix for more details)

We could find interesting facts through images restored by the autoencoder. Figure 2 shows an abstract, high-dimensional augmentation that cannot be obtained by standard input augmentation, such as changing the head direction of an animal, creating a new type of airplane wing, or bending the tail.

Based on these experimental observations, we can infer why our methods are effective when the number of labeled data is smaller. Variations in the manifolds around the data point allow the exponentially diverse forms of images to be represented, thus yielding a more stable and accurate target for unlabeled data.

3.5 ABLATION STUDY

To verify that feature-based augmentation actually shows a regularization such as dropout, the following experiments were carried out by varying dropout probability p, feature-based augmentation scaling factor k on 4000 CIFAR-10 labels. Figure 3(a) shows the relation between dropout probability p and test error when the rest of the hyperparameters are fixed. When the dropout p was 0.5, the performance was rather decreased. Thus, the experimental results also showed that feature-based augmentation had a regularization, and 0.8 was the optimal value. Figure 3(b), we can reveal that the feature-based augmentation scaling factor k is beneficial to semi-supervised learning and converging better results. There is an optimal value for k and 0.15 is that value.

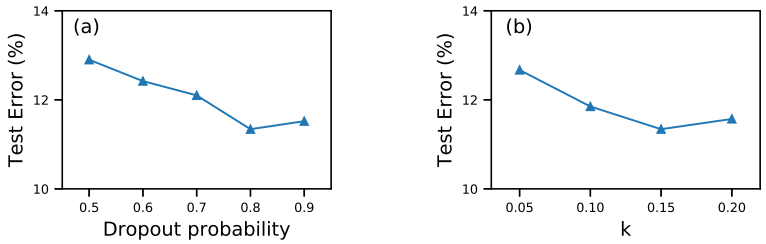

Figure 3: Test error comparison on 4000-label CIFAR-10 for varying dropout probability p, feature-based augmentation scaling factor k. Feature-based augmentation has the regularization effect as traditional regularization, which is useful for semi-supervised learning. (a) feature-based augmentation scaling factor $k = 0.15$. (b) dropout probability $p = 0.8$

4 Related work

Zhu (2005) say that unlabeled data can be easily obtained compared to labeled data, but there are not many ways to use it. Therefore, semi-supervised learning can solve this problem by designing better classifiers using large amounts of unlabeled data, together with labeled data. Semi-supervised learning is receiving great attention because it can achieve high accuracy with little effort.

There are various semi-supervised learning methods. The key idea is to improve the quality of the target of unlabeled data, and there are two main approaches. The first is to add noise to representations. The other approach is to select a teacher model that can generate consistent target values.

From the perspective of the first approach, Sajjadi et al. (2016) analyzed that the stochastic transformations and perturbations can achieve better generalization and stability. and Miyato et al. (2018) adds adversarial perturbations to input data in a gradient direction. Any slight change in the input should be recognized as the same (Goodfellow et al., 2015). This method also confirmed that it had better generalization performance than the random perturbation and showed impressive performance at the CIFAR-10 dataset 4000 labels.

Rasmus et al. (2015) implemented DDS(Sarela & Valpola, 2005), which produces noise-added student predictions and noise-free teacher predictions. This method uses a denoising layer to make teacher predictions from student predictions. Both Temporal ensembling(Laine & Aila (2016)) and Mean Teacher(Tarvainen & Valpola (2017)) use exponential moving average(EMA) to generate stable and accurate targets. However, Laine & Aila (2016) applies EMA to the prediction value itself, and Tarvainen & Valpola (2017) applies EMA to the network parameters of the model. Therefore, Tarvainen & Valpola (2017) shows better performance than Laine & Aila (2016) because it allows more frequent target updates.

Another methods for semi-supervised learning is the use of a generative models. Kingma et al. (2014) employed multiple probabilistic models and solves semi-supervised problem as if it were a classification problem with specialized missing data.

Data belonging to the same class resemble each other. Thus, the label propagation of Zhu & Ghahramani (2002) has the advantage of pushing the labeled data out of the decision boundaries away. Weston et al. (2008) is applied the label propagation method to semi-supervised learning. Using the kernel, the embedded features are shown to have a regularization effect.

Grandvalet & Bengio (2005) claimed one of the regularization techniques, which is the entropy minimization method. This method has the effect of exaggerating the prediction of the model at each data point. Therefore, higher performance was achieved when this method was applied in VAT (Miyato et al., 2018), which is suitable for semi-supervised learning tasks.

5 CONCLUSION

Semi-supervised learning, both in theory and in practice, is a big concern. There were several attempts to overcome the lack of labeled data. In this respect, data augmentation is considered as a effective way especially for semi-supervised learning. In this work, we proposed a simple and efficient feature-based augmentation method, which is a way to generate new data that is more realistic and plausible because it adds noise based on the parameters of the network in the latent space, which is the intermediate stage of representation. To verify how feature-based augmentation actually affects on input images, we used an autoencoder to check the variations in input images. In fact, we have seen more abstract augmentation taking place in higher dimensions. Also, it is expected to improve the quality of the target, if we perform parallel ensembling with augmented features. Our method has confirmed that it has a greater effect, especially when the number of labeling data is smaller, and achieved the state-of-the-art results.

ACKNOWLEDGMENTS

anonymous acknowledgments

REFERENCES

Martín 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, Dandelion Mané, 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 Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average, 2018.

Yoshua Bengio, Gregoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 552–560. PMLR, Atlanta, Georgia, USA, 17–19 Jun 2013. URL `http://proceedings.mlr.press/v28/bengio13.html`.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Rudolph-Lilith Michelle Destexhe, Alain. *Neuronal Noise*. Springer, 2012. ISBN 978-0-387-79020-6.

Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space. arXiv:1702.05538, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, 2016. URL `http://dl.acm.org/citation.cfm?id=3045390.3045502`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014a. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618, 9780262035613.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2014b.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems 17*, pp. 529–536. MIT Press, 2005. URL http://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv:1312.6114, 2013.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3581–3589. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. arXiv:1610.02242, 2016.

Yiting Li, Lu Liu, and Robby T. Tan. Decoupled certainty-driven consistency loss for semi-supervised learning, 2019.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

T. Miyato, S. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2018.2858821.

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*. 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

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. URL http://papers.nips.cc/paper/5947-semi-supervised-learning-with-ladder-networks.pdf.

E. Roderich Gossen Kelvin E. Jones Richard B. Stein. Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6:389–397, 2005.

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 and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 901–909. Curran Associates, Inc., 2016.

Jaakko Sarela and Harri Valpola. Denoising source separation. *Journal of Machine Learning Research*, 6:233–272, 2005.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1195–1204. Curran Associates, Inc., 2017.

Mark D. McDonnell Lawrence M. Ward. The benefits of noise in neural systems: bridging theory and experiment. *Nature Reviews Neuroscience*, 12:415–425, 2011.

Peter E Latham Alexandre Pouget Wei Ji Ma, Jeffrey M Beck. Bayesian inference with probabilistic population codes. *Nature Reviews Neuroscience*, 9:1432–1438, 2006.

Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1168–1175, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390303. URL http://doi.acm.org/10.1145/1390156.1390303.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

A Appendix

A. network architecture

Table 3: The classifier network(the encoder of the autoencoder)

13-layer CNN
3×3 conv. 128 leaky ReLU
3×3 conv. 128 leaky ReLU
3×3 conv. 128 leaky ReLU
Maxpool 2×2, stride 2
dropout, $p = 0.8$
3×3 conv. 256 leaky ReLU
3×3 conv. 256 leaky ReLU
3×3 conv. 256 leaky ReLU
Maxpool 2×2, stride 2
dropout, $p = 0.8$
3×3 conv. 512 leaky ReLU
1×1 conv. 256 leaky ReLU
1×1 conv. 128 leaky ReLU
Avgpool $6 \times 6 \rightarrow 1 \times 1$
dense $128 \rightarrow 10$ (Feature-based augmentation in this layer.)
softmax

▷ The input of the dense layer is a feature that applies augmentation in this paper of size [batch size, 128].

▷ Decoder was used to see how augmentation on feature-level affects the input image. Although the restored images are somewhat blurred because the decoder of the autoencoder is fairly shallow compared to the encoder, the augmentation can be clearly observed in Figure. 3 of Section 3.3. We did not use batch normalization in decoder unlike encoder.

Table 4: The decoder of autoencoder

4-layer Transposed Convolutional network
4×4 Trasnposed conv. 512 leaky ReLU
3×3 Trasnposed conv. 256 leaky ReLU
3×3 Trasnposed conv. 128 leaky ReLU
3×3 Trasnposed conv. 3

B. SVHN

We normalized the input images to have zero mean and unit variance. We applied the stochastic input augmentation; translation(randomly 2×2 pixel translate), Gaussian noise(adding noise, $\sigma = 0.15$). We used feature-based augmentation scaling factor $k = 0.15$. We used leaky ReLU with $\alpha = 0.1$. We used dropout and mean-only batch normalization as regularization, and updated the network parameters using Adam optimizer with learning rate 0.001 and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. In addition, We used the scaling factor $\lambda 1 = 1$, $\lambda_2 = 2$ of objective function.

We trained the network with minibatches of size 100 and the number of unlabeled data is 50. We trained during total epoch 300.

C. CIFAR-10

We conducted ZCA preprocessing prior to the semi-supervised learning. We applied the stochastic input augmentation; translation(randomly 2×2 pixel translate), horizontal flip(randomly, $p = 0.5$), Gaussian noise(adding noise, $\sigma = 0.15$). We used feature-based augmentation scaling factor $k = 0.15$. We used leaky ReLU with $\alpha = 0.1$. We used dropout and mean-only batch normalization as regularization, and updated the network parameters using Adam optimizer with learning rate 0.001 and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. In addition, We used the scaling factor $\lambda 1 = 1$, $\lambda_2 = 2$ of objective function.

We trained the network with minibatches of size 100 and the number of unlabeled data is 50. We trained during total epoch 500.