

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Regularization in Deep Neural Networks

by

Guoliang Kang

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Sydney, Australia

2019

Copyright@Data to Decision CRC

Certificate of Authorship/Originality

I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis. This research is supported by the Australian Government Research Training Program.

Guoliang Kang

Feb. 2019

Production Note:
Signature removed
prior to publication.

ABSTRACT

Regularization in Deep Neural Networks

by

Guoliang Kang

Recent years have witnessed the great success of deep learning. As the deep architecture becomes larger and deeper, it is easy to overfit to relatively small amount of data. Regularization has proved to be an effective way to reduce overfitting in traditional statistical learning area. In the context of deep learning, some special design is required to regularize their training process. Generally, we firstly proposed a new regularization technique named “Shakeout” to improve the generalization ability of deep neural networks beyond Dropout, via introducing a combination of L_0 , L_1 , and L_2 regularization effect into the network training. Then we considered the unsupervised domain adaptation setting where the source domain data is labeled and the target domain data is unlabeled. We proposed “deep adversarial attention alignment” to regularize the behavior of the convolutional layers. Such regularization reduces the domain shift existing at the start in the convolutional layers which has been ignored by previous works and leads to superior adaptation results.

Dissertation directed by Professor Yi Yang

Center of AI, School of Software

Acknowledgements

First and foremost, I am tremendously grateful for my supervisor Yi Yang for his continuous support and guidance throughout my PhD, and for providing me the freedom to work on a variety of problems. I am grateful for Prof. Dacheng Tao, who has ever supervised me and provided me support. I am grateful for my co-supervisor Jun Li for his beneficial suggestions for my research.

I am happy to collaborate with the previous postdoc in our team Liang Zheng. Thanks for his creative guidance and suggestions for my research and academic writing. I am happy to collaborate with many creative students in our team. I am grateful for the creative discussions with them and I really appreciate the kind and useful suggestions given by them.

Thanks for all the people that ever helped me and encouraged me.

Finally, this thesis is dedicated to my parents Zhongwen Kang, Fenglan Zhang, and my wife Mingyue You, for all the years of love and support. They are always the source of my power and the reason I insist on pursuing my dream.

Guoliang Kang
Sydney, 2019.

List of Publications

Journal Papers

- J-1. **G. Kang**, J. Li, and D. Tao, “Shakeout: A new approach to regularized deep neural network training”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 12451258, 2018.

Conference Papers

- C-1. **G. Kang**, J. Li, and D. Tao, “Shakeout: A new regularized deep neural network training scheme,” in *AAAI*, 2016.
- C-2. **G. Kang**, L. Zheng, Y. Yan, and Y. Yang, “Deep Adversarial Attention Alignment for Unsupervised Domain Adaptation: the Benefit of Target Expectation Maximization”, in *ECCV*, 2018

Contents

Certificate	iii
Abstract	iv
Acknowledgments	v
List of Publications	vi
List of Figures	x
1 Introduction	1
1.1 Background	1
1.2 Thesis Organization	4
2 Literature Survey	6
2.1 Regularization for Supervised Learning	6
2.1.1 Data Augmentation	6
2.1.2 Model Ensemble.	7
2.1.3 Weight Tying or Sharing	7
2.1.4 Adversarial Training	8
2.1.5 Teacher-student Framework	8
2.1.6 Dropout	8
2.2 Regularization for Unsupervised Domain Adaptation	9
2.2.1 Explicitly Inducing Regularization Term	10
2.2.2 Implicitly Imposing Regularization	11

3	Regularization for Supervised Learning	12
3.1	Introduction	12
3.2	Related Work	14
3.3	Method	16
3.3.1	Regularization Effect of Shakeout	18
3.3.2	Shakeout in Multilayer Neural Networks	26
3.4	Experiments	29
3.4.1	Shakeout and Weight Sparsity	30
3.4.2	Classification Experiments	32
3.4.3	Stabilization Effect on the Training Process	43
3.4.4	Practical Recommendations	45
3.5	Conclusion	47
4	Regularization for Unsupervised Domain Adaptation	49
4.1	Introduction	49
4.2	Related Work	52
4.3	Method	54
4.3.1	Adversarial Data Pairing	55
4.3.2	Attention Alignment	56
4.3.3	Training with EM	58
4.3.4	Deep Adversarial Attention Alignment	61
4.4	Experiments	62
4.4.1	Setup	62
4.4.2	Implementation Details	63
4.4.3	Evaluation	64

4.4.4	Ablation Study	69
4.4.5	Comparing Different Attention Discrepancy Measures	69
4.4.6	Impact of Hyper-parameters	69
4.4.7	Comparison with Different Variants of Attention	70
4.5	Conclusion	71
5	Conclusion	72
	Bibliography	73

List of Figures

3.1	Comparison between Shakeout and Dropout operations. This figure shows how Shakeout and Dropout are applied to the weights in a linear module. In the original linear module, the output is the summation of the inputs \mathbf{x} weighted by \mathbf{w} , while for Dropout and Shakeout, the weights \mathbf{w} are first randomly modified. In detail, a random switch \hat{r} controls how each w is modified. The manipulation of w is illustrated within the amplifier icons (the red curves, best seen with colors). The coefficients are $\alpha = 1/(1 - \tau)$ and $\beta(w) = cs(w)$, where $s(w)$ extracts the sign of w and $c > 0$, $\tau \in [0, 1]$. Note the sign of $\beta(w)$ is always the same as that of w . The magnitudes of coefficients α and $\beta(w)$ are determined by the Shakeout hyper-parameters τ and c . Dropout can be viewed as a special case of Shakeout when $c = 0$ because $\beta(w)$ is zero at this circumstance.	18
3.2	Regularization effect as a function of a single weight when other weights are fixed to zeros for logistic regression model. The corresponding feature x is fixed at 1.	24
3.3	The contour plots of the regularization effect induced by Shakeout in 2D weight space with input $\mathbf{x} = [1, 1]^T$. Note that Dropout is a special case of Shakeout with $c = 0$	27

3.4	Distributions of the weights of the autoencoder models learned by different training approaches. Each curve in the figure shows the frequencies of the weights of an autoencoder taking particular values, i.e. the empirical population densities of the weights. The five curves correspond to five autoencoders learned by standard back-propagation, Dropout ($\tau = 0.5$), Gaussian Dropout ($\sigma^2 = 1$) and Shakeout ($\tau = 0.5, c = \{1, 10\}$). The sparsity of the weights obtained via Shakeout can be seen by comparing the curves.	33
3.5	Features captured by the hidden units of the autoencoder models learned by different training methods. The features captured by a hidden unit are represented by a group of weights that connect the image pixels with this corresponding hidden unit. One image patch in a sub-graph corresponds to the features captured by one hidden unit.	34
3.6	Classification of two kinds of neural networks on MNIST using training sets of different sizes. The curves show the performances of the models trained by standard BP, and those by Dropout and Shakeout applied on the hidden units of the fully-connected layer. . .	36
3.7	Classification on CIFAR-10 using training sets of different sizes. The curves show the performances of the models trained by standard BP, and those by Dropout and Shakeout applied on the hidden units of the fully-connected layer.	38
3.8	Comparison of the distributions of the magnitude of weights trained by Dropout and Shakeout. The experiments are conducted using AlexNet on ImageNet-2012 dataset. Shakeout or Dropout is applied on the last two fully-connected layers, i.e. FC7 layer and FC8 layer. .	39

- 3.9 Distributions of the maximum magnitude of the weights connected to the same input unit of a layer. The maximum magnitude of the weights connected to one input unit can be regarded as a metric of the importance of that unit. The experiments are conducted using AlexNet on ImageNet-2012 dataset. For Shakeout, the units can be approximately separated into two groups and the one around zero is less important than the other, whereas for Dropout, the units are more concentrated. 40
- 3.11 The value of $-V(D, G)$ as a function of iteration for the training process of DCGAN. DCGANs are trained using standard BP, Dropout and Shakeout for comparison. Dropout or Shakeout is applied on the discriminator of GAN. 42
- 3.10 Relative accuracy loss as a function of the weight pruning ratio for Dropout and Shakeout based on AlexNet architecture on ImageNet-2012. The relative accuracy loss for Dropout is much severe than that for Shakeout. The largest margin of the relative accuracy losses between Dropout and Shakeout is 22.50%, which occurs at the weight pruning ratio $m = 96\%$ 43
- 3.12 The minimum and maximum values of $-V(D, G)$ within fixed length intervals moving from the start to the end of the training by standard BP, Dropout and Shakeout. The optimal value $\log(4)$ is obtained when the imaginary data distribution $P(\hat{\mathbf{x}})$ matches with the real data distribution $P(\mathbf{x})$ 44

- 3.13 Validation error as a function of training epoch for Dropout and Shakeout on CIFAR-10 with training set size at 40000. The architecture adopted is WRN-16-4. “DPO” and “SKO” represent “Dropout” and “Shakeout” respectively. The following two numbers denote the hyper-parameters τ and c respectively. The learning rate decays at epoch 60, 120, and 160. After the first decay of learning rate, the validation error increases greatly before the steady decrease (see the enlarged snapshot for training epochs from 60 to 80). It can be seen that the extent of error increase is less severe for Shakeout than Dropout. Moreover, Shakeout recovers much faster than Dropout does. At the final stage, both of the validation errors steadily decrease (see the enlarged snapshot for training epochs from 160 to 200). Shakeout obtains comparable or even superior generalization performance to Dropout. 46
- 4.1 Attention visualization of the last convolutional layer of ResNet-50. The original *target* input images are illustrated in **(a)**. The corresponding attentions of the source network, the target network trained on labeled target data, and the target network adapted with adversarial attention alignment are shown in **(b)**, **(c)**, and **(d)** respectively. 50

- 4.2 The framework of deep adversarial attention alignment. We train a source network and fix it. The source network guides the attention alignment of the target network. The target network is trained with real and synthetic images from both domains. For labeled real source and synthetic target data, we update the network by computing the cross-entropy loss between the predictions and the ground-truth labels. For unlabeled real target and synthetic source images, we maximize the likelihood of the data with EM steps. The attention distance for a pair of images (as illustrated in the “Data Pairs” block) passing through the source network and the target network, respectively, is minimized. 54
- 4.3 Paired data across domains using CycleGAN. **(a)** and **(c)**: real images sampled from source and target domain, respectively. **(b)**: a synthetic target image paired with **(a)** through G^{ST} . **(d)**: a synthetic source image paired with a real target image **(c)** through G^{TS} 56
- 4.4 Analysis of the training process (EM is implemented). **Left**: The trend of \mathcal{L}^{AT} during training with and without imposing the \mathcal{L}^{AT} penalty term. **Right**: The curves of test accuracy on the target domain. The results of tasks $\mathbf{W} \rightarrow \mathbf{A}$ and $\mathbf{D} \rightarrow \mathbf{A}$ are presented. The results for other tasks are similar. One iteration here represents one update of the network M^{post} (see Section 4.3.3). 67
- 4.5 The impact of hyper-parameters on the classification accuracy of target model. The results for task $\mathbf{D} \rightarrow \mathbf{A}$ on Office-31 are illustrated, with a comparison to the previous state-of-the-art (SOTA). The trends are similar for other tasks. **Left**: Accuracy vs. p_t . **Right**: Accuracy vs. β 70

