SHAKING ACOUSTIC SPECTRAL SUB-BANDS CAN BETTER REGULARIZE LEARNING IN AFFECTIVE COMPUTING

Che-Wei Huang, Shrikanth Narayanan

University of Southern California, Los Angeles, CA 90089
cheweihu@usc.edu, shri@sipi.usc.edu

ABSTRACT

In this work, we investigate a recently proposed regularization technique based on multi-branch architectures, called Shake-Shake regularization, for the task of speech emotion recognition. In addition, we also propose variants to incorporate domain knowledge into model configurations. The experimental results demonstrate: 1) independently shaking sub-bands delivers favorable models compared to shaking the entire spectral-temporal feature maps. 2) with proper patience in early stopping, the proposed models can simultaneously outperform the baseline and maintain a smaller performance gap between training and validation.

Index Terms— Shake-Shake Regularization, Sub-band Shaking, Adversarial Training, Affective Computing, Speech Emotion Recognition

1. INTRODUCTION

Deep convolutional neural networks have been successfully applied to several pattern recognition tasks such as image recognition [1], machine translation [2] and speech emotion recognition [3]. Currently, to successfully train a deep neural network, one needs either a sufficient number of training samples to implicitly regularize the learning process, or employ techniques like weight decay and dropout [4] and its variants to explicitly keep the model from over-fitting.

In the recent years, one of the most popular and successful architectures is the residual neural network (ResNet) [1]. The ResNet architecture was designed based on a key assumption that it is more efficient to optimize the residual term than the original task mapping. Since then, a great deal of effort in machine learning and computer vision has been dedicated to study the multi-branch architecture.

Deep convolutional neural networks have also gained much attention in the community of affective computing mainly because of its outstanding ability to formulate discriminative features for the top-layer classifier. Usually the number of parameters in a model is far more than the number of training samples and thus it requires heavy regularization to train deep neural networks for affective computing. However, since the introduction of batch normalization [5], the gains obtained by using dropout for regularization have decreased [5, 6, 7]. Yet, multi-branch architectures have emerged as a promising alternative.

Regularization techniques based on multi-branch architectures such as Shakeout [8] and Shake-Shake [9] have delivered impressive performances on standard image datasets such as the CIFAR-10 [10]. In a clever way, both of them utilize multiple branches to learn different aspects of the relevant information and then a summation in the end follows for information alignment among branches. Instead of using multiple branches, a recent work [11] based on a mixture of experts showed that randomly projecting samples is able to break the structure of adversarial noise that could easily confound the model and as a result mislead the learning process. Despite not being an end-to-end approach, it shares the same idea of integrating multiple streams of model-based diversity.

In this work, we study the Shake-Shake regularized ResNet for speech emotion recognition. In addition to shaking the entire spectral-temporal feature maps with the same strength, we propose to address different spectral sub-bands independently based on our hypothesis of the non-uniform distribution of affective information over the spectral axis. There has been work on multi-stream framework in speech processing. For example, Mallidi et al. [12] designed a robust speech recognition system using multiple streams, each of them attending to a different part of the feature space, to fight against noise. However, lacking both multiple branches and the final information alignment, the design philosophy is fundamentally different from that of multi-branch architectures. In fact, we intend to serve this work as a bridge between the multi-stream framework and the multi-branch architecture.

1.1. Shake-Shake Regularization

Shake-Shake regularization [9] is a recently proposed technique to regularize training of deep convolutional neural networks for image recognition tasks. This regularization technique based on multi-branch architectures promotes stochastic mixtures of forward and backward propagations from network branches in order to create a flow of model-based adversarial learning samples/gradients during the training phase. Owing to it excellent ability to combat over-fitting even in the presence of batch normalization, the Shake-Shake regularized 3-branch residual neural network [9] has achieved the current state-of-the-art performance on the CIFAR-10 image dataset.

An overview of a 3-branch shake-shake regularized ResNet is depicted in Fig. 1. In addition to the short-cut flow (in light gray), there are other two residual branches $\mathbf{B}(x)$,

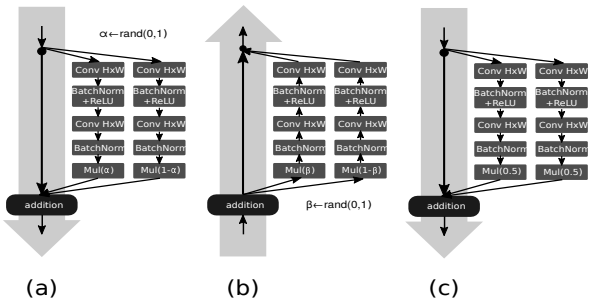

Fig. 1. An overview of a 3-branch shake-shake regularized residual block [9]. (a) Forward propagation during the training phase (b) Backward propagation during the training phase (c) Testing phase. The coefficients α and β are sampled from the uniform distribution over $[0, 1]$ to scale down the forward and backward flows during the training phase.

each of them consisting of a sequence of layers stacked in order: Conv$H \times W$, Batch Normalization, ReLU, Conv$H \times W$, Batch Normalization, where Conv$H \times W$ represents a convolutional layer with filters of size $H \times W$ without pooling and ReLU is the rectified linear unit $\mathrm{ReLU}(\mathrm{x}) = \max(0, \mathrm{x})$.

The Shake-Shake regularization adds to the aggregate of the output of each branch an additional layer, called the ShakeShake layer, to randomly generate adversarial flows in the following way:

$$\mathrm{ShakeResNet}_N(\mathbf{X}) = \mathbf{X} + \sum_{n=1}^{N} \mathrm{ShakeShake}\left(\{\mathbf{B}_n(\mathbf{X})\}_{n=1}^{N}\right)$$

where in the forward propagation for $\mathbf{a} = [\alpha_1, \cdots, \alpha_N]$ sampled from the $(N-1)$-simplex (Fig. 1 (a))

$$\mathrm{ShakeResNet}_N(\mathbf{X}) = \mathbf{X} + \sum_{n=1}^{N} \alpha_n \mathbf{B}_n(\mathbf{X}),$$

while in the backward propagation for $\mathbf{b} = [\beta_1, \cdots, \beta_N]$ sampled from the $(N-1)$-simplex and $\mathbf{g}$ the gradient from the top layer, the gradient entering into $\mathbf{B}_n(x)$ is $\beta_n \mathbf{g}$ (Fig. 1 (b)). At the testing time, the expected model is then evaluated for inference by taking the expectation of the random sources in the architecture (Fig. 1 (c)).

In each mini-batch, to apply scaling coefficients α or β either on the entire mini-batch or on each individual sample independently can also make a difference [9].

2. PROPOSED MODELS

In addition to batch- or sample-wise shaking, when it comes to the area of acoustic processing, there is another orthogonal dimension to consider: the spectral domain. Leveraging domain knowledge, our proposed models are based on a simple but plausible hypothesis that affective information is distributed non-uniformly over the spectral axis [13]. Therefore, there is no reason to enforce the entire spectral axis to be shaken with the same strength concurrently. Furthermore, adversarial noise may exist and extend over the spectral axis. By deliberately shaking spectral sub-bands independently, the structure of adversarial noise may be broken and become less confounding to the model.

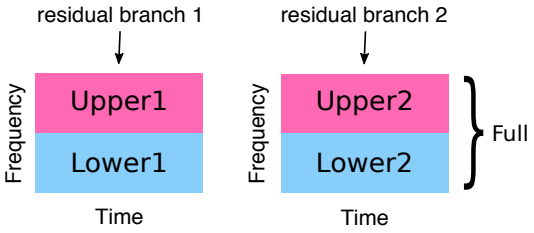

Fig. 2. An illustration for the sub-band definitions.

Before we formally define the proposed models, we introduce the definition of sub-bands first. Fig. 2 depicts the definition for sub-bands in a 3-branch residual block. Here we slightly abuse the notations of frequency and time because after two convolutional layers these axes are not exactly the same as those of input to the branches; however, since convolution is a local operation they still hold the corresponding spectral and temporal nature. At the output of each branch, we define the high-frequency half to be the upper sub-band while the low-frequency half to be the lower sub-band. We take the middle point on the spectral axis to be the border line for simplicity. The entire output is called the full band.

Having defined these concepts, we denote $\mathbf{X}$ the input to a residual block, $\mathbf{X}^i$ the full band from the i-th branch, $\mathbf{X}_u^i$ the upper sub-band from the i-th branch and $\mathbf{X}_l^i$ the lower sub-band from the i-th branch. Naturally, the relationship between them is given by $\mathbf{X}^i = \left[\mathbf{X}_u^i | \mathbf{X}_l^i\right]$. We also denote $\mathbf{Y}$ the output of a Shake-Shake regularized residual block.

To demonstrate that shaking sub-bands can better regularize the learning process, we propose the following models for benchmarking:

1. Shake the full band (**Full**)

$$\mathbf{Y} = \mathbf{X} + \sum_{n=1}^{N} \mathrm{ShakeShake}\left(\{\mathbf{X}^n\}_{n=1}^{N}\right). \quad (1)$$

2. Shake the upper sub-band (**Upper**)

$$\begin{aligned} \mathbf{Y} &= \mathbf{X} \\ &+ \left[\sum_{n=1}^{N} \mathrm{ShakeShake}\left(\{\mathbf{X}_u^n\}_{n=1}^{N}\right) \middle| \sum_{n=1}^{N} \mathbf{X}_l^n\right]. \end{aligned} \quad (2)$$

3. Shake the lower sub-band (**Lower**)

$$\begin{aligned} \mathbf{Y} &= \mathbf{X} \\ &+ \left[\sum_{n=1}^{N} \mathbf{X}_u^n \middle| \sum_{n=1}^{N} \mathrm{ShakeShake}\left(\{\mathbf{X}_l^n\}_{n=1}^{N}\right)\right]. \end{aligned} \quad (3)$$

4. Shake both sub-bands but independently (**Both**)

$$\mathbf{Y} = \mathbf{X} + [\mathbf{Y}_u | \mathbf{Y}_l], \quad (4)$$

$$\mathbf{Y}_u = \sum_{n=1}^{N} \text{ShakeShake}\left(\{\mathbf{X}_u^n\}_{n=1}^N\right),$$

$$\mathbf{Y}_l = \sum_{n=1}^{N} \text{ShakeShake}\left(\{\mathbf{X}_l^n\}_{n=1}^N\right).$$

3. EXPERIMENTS

3.1. Datasets

We use four publicly available emotion corpora to demonstrate the effectiveness of the proposed models, including the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [14], the eNTERFACE'05 Audio-Visual Emotion Database [15], the EMOVO Corpus [16] and the Surrey Audio-Visual Expressed Emotion (SAVEE) [17]. All of these corpora are multi-modal in which speech, facial expression and text all convey a certain degree of affective information. However, in this paper we solely focus on the acoustic modality for experiments.

The intersection of emotional classes in these four corpora consists of joy, anger, sadness and fear. Therefore, we formulate the experimental task into a sequence classification of 4 classes. In particular, we employ both speaking and singing sets from the RAVDESS corpus, all of the eNTERFACE'05, EMOVO and SAVEE corpora. However, one of the female actors in RAVDESS corpus does not have the singing part and we thus leave her speech part out of the experiments as well. The actor 23 in eNTERFACE'05 has only 3 utterances portraying joy, which makes the emotional class distribution slightly imbalanced. As a result, we have 2885 utterances in total. Table 1 summarizes the information about these four corpora.

Corpus	No. Actors	No. Utterances			
		joy	anger	sadness	fear
RAVDESS	23	368	368	368	368
eNTERFACE	42	207	210	210	210
EMOVO	6	84	84	84	84
SAVEE	4	60	60	60	60
Total	75	719	722	722	722

Table 1. An overview of these selected corpora, including the number of actors and the distribution of utterances in the emotional classes.

For the evaluation, we adopt a 4-fold cross validation strategy. To begin with, we split the actor set into 4 partitions. Moreover, we impose extra constraints to make sure that each partition is as gender and corpus uniform as possible. For example, each actor set partition is randomly distributed with 2-3 female actors and 8-9 male actors from the eNTERFACE'05 corpus. More details are provided in Table 2. By partitioning the actor set, it becomes easier to maintain speaker independence between training and validation throughout all of the experiments.

Corpus	Actor Set Partition			
	1	2	3	4
RAVDESS	3F, 3M	3F, 3M	3F, 3M	2F, 3M
eNTERFACE	2F, 9M	2F, 8M	2F, 8M	3F, 8M
EMOVO	1F, 0M	1F, 1M	1F, 1M	0F, 1M
SAVEE	0F, 1M	0F, 1M	0F, 1M	0F, 1M
Total	6F, 13M	6F, 13M	6F, 13M	5F, 13M

Table 2. F: female, M: male. The gender and corpus distributions in each actor set partition of the cross validation.

Models	Layers	No. Params
3-Branch ResNet [w/ shake reg. {on **Both**}]	Conv2d(4,2,16) + BatchNorm2d + ReLU + (Shortcut, Branch$\times$2) + [ShakeShake $\{\times 2\}$+] ReLU + Mean-Pooling + Dropout(0.5) + Linear(256) + ReLU + Dropout(0.25) + Linear(256) + ReLU + Linear(4)	1.17 M
Branch	Conv2d(4,2,64) + BatchNorm2d + ReLU + Conv2d(4,4,128) + BatchNorm2d	10.2 K

Table 3. Network architecture, layers and the number of parameters in the baseline and proposed models. Conv2d(N, H, W) stands for a $2D$ convolutional layer with N filters of size $H \times W$ and Linear(N) for a fully connected layer with N nodes. The Mean-Pooling layer represents the temporal pooling for generating an utterance representation.

3.2. Experimental Setup

To start with, we extract the spectrograms of each utterance with a 25ms window for every 10ms using the Kaldi [18] library. Cepstral mean and variance normalization is then applied on the spectrogram frames per utterance. To equip each frame with a certain context, we splice it with 10 frames in the left and 5 frames in the right. Therefore, a resulting spliced frame has a resolution of 16×257. Since emotion involves a longer-term mental state transition, we further down-sample the frame rate by a factor of 8 to simplify and expedite the training process.

We establish a baseline of 3-branch ResNet and list the details in Table 3. For each utterance, a simple mean pooling is taken at the output of the residual block to form an utterance representation before feeding it to the fully connected layers. We avoid explicit temporal modeling layers such as a long short-term memory recurrent network because our focus is on shaking the ResNet. Note that the ShakeShake layer has no parameter to learn and hence the model size does not change during this work. We implement the ShakeShake layer as well as the entire network architecture using the Py-

Model	Patience in Early Stopping												
	9	11	13	15	17	19	21	26	31	36	41	46	51
Baseline	48.01	48.01	48.57	50.59	51.78	56.03	56.03	56.49	57.66	57.66	57.66	58.85	58.85
Full	47.26	47.26	48.46	52.58	**53.23**	53.23	53.23	56.24	56.24	56.94	56.94	57.34	57.34
Upper	47.30	48.62	52.66	53.31	54.73	55.26	55.47	56.00	57.50	57.50	57.79	57.79	57.79
Lower	45.62	46.55	47.66	48.04	48.79	48.79	49.21	51.34	51.48	54.18	54.18	54.18	54.18
Both	46.97	49.66	50.72	51.61	**54.13**	54.13	54.13	54.58	55.08	57.20	**57.66**	57.79	57.79

Table 4. Averaged unweighted accuracy (%) on the validation partition over 4-fold cross validation.

Model	Patience in Early Stopping												
	9	11	13	15	17	19	21	26	31	36	41	46	51
Baseline	-0.14	-0.14	1.45	1.90	4.50	8.93	8.93	11.42	14.74	14.74	14.74	16.34	16.34
Full	0.08	0.08	0.84	3.16	**3.33**	3.33	3.33	8.05	8.05	10.57	10.57	11.13	11.13
Upper	2.35	2.68	4.98	5.85	7.38	10.47	11.45	14.09	16.08	16.08	18.62	18.62	18.62
Lower	0.13	-0.18	0.84	1.66	3.34	3.34	3.99	8.56	8.90	14.27	14.27	14.27	14.27
Both	1.41	2.90	2.74	2.22	**2.73**	2.73	2.73	4.57	6.13	8.16	**10.14**	11.53	11.53

Table 5. Averaged gap between the unweighted accuracy (%) on the training and validation partitions over 4-fold cross validation.

Torch [19] library. Only the Shake-Shake combination [9] is used and shaking is applied independently per frame. Due to space limit, we leave other combinations for future work. The models are learned using the Adam optimizer [20] with an initial learning rate of 0.001 and the training is carried out on an NVIDIA Tesla K80 GPU. We use a mini-batch of 64 utterances across all model training and let each experiment run for 200 epochs in order to investigate the regularization power when over-training occurs.

3.3. Results

Table 4 and 5 summarize the benchmarking of the unweighted accuracy (UA) of cross validation and the gap of UA between training and validation with respect to different patience in early stopping.

In Table 4, the underlined numbers indicate when a model performs better than the baseline. A clear trend is that if the training process is stopped early, models with regularization tend to outperform the baseline. On the other hand, if the training goes too far, the situation is almost entirely the opposite. However, even when over-trained the margin that the baseline has over the other regularized models is only around 1% except for the model **Lower**. One thing to note, in particular, is that the model **Lower** seems to struggle with difficulties in capturing the affective pattern since the beginning.

In Table 5, the underlined numbers indicate when a model has a smaller gap than that of a baseline under the same patience. The apparent trend here is that if we let the training keep going, almost all regularized models tend to have a smaller gap compared to the baseline; in other words, the baseline tends to overfit more under the same patience in early stopping. We also note the model **Upper**, despite being regularized, appears to have a larger gap than the baseline does since the beginning of learning.

Fortunately, these two trends overlap about when patience equals 17. In both Table 4 and 5, the boldfaced numbers represent when a model performs not worse than the baseline and has a smaller gap. Based on these two criteria, the models **Full** and **Both** both demonstrate a superior performance while staying far from being over-trained. Moreover, the model **Both** is able to match the performance of the baseline even in the over-trained region where patience equals 41, while still achieving a smaller gap compared to the baseline. Another observation is that regularized models generally require more patience to reach the same gap, especially the model **Both**. This suggests early stopping under the same patience may not be an universally optimal strategy.

When benchmarked with the model **Full**, the model **Both** always has a higher accuracy whenever they achieve comparable gaps (e.g. 3.33 versus 2.73, 8.05 versus 8.16, etc), and most of the time when under the same patience. This phenomenon corroborates our hypothesis that independently shaking the sub-bands would help to learn a better model for affective computing. Nevertheless, the concerning fact that the models **Upper** and **Lower** show totally different characteristics requires further investigation in the future.

4. CONCLUSIONS

We have proposed a Shake-Shake regularized multi-branch ResNet model for speech emotion recognition. In particular, we have experimented with different configurations for the Shake-Shake regularization on the full band, the upper and the lower sub-bands alone and simultaneously. The results support our hypothesis that shaking different sub-bands with independent strength would benefit learning in affective computing. With a commonly used patience, say 17, the models **Full** and **Both** are able to deliver competitive performances over the baseline with reduced over-fitting. However, given the opposite behaviors of the models **Upper** and **Lower**, further investigation is necessary in the future.

5. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin, "Convolutional Sequence to Sequence Learning," 2017, arXiv:1705.03122.

[3] Che-Wei Huang and Shrikanth S. Narayanan, "Deep Convolutional Recurrent Neural Network with Attention Mechanism for Robust Speech Emotion Recognition," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2017.

[4] 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.*, vol. 15, no. 1, Jan. 2014.

[5] Sergey Ioffe and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[6] Sergey Zagoruyko and Nikos Komodakis, "Wide Residual Networks," in *BMVC*, 2016.

[7] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger, "Deep Networks with Stochastic Depth," in *ECCV*, 2016.

[8] Guoliang Kang, Jun Li, and Dacheng Tao, "Shakeout: A New Regularized Deep Neural Network Training Scheme," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[9] Xavier Gastaldi, "Shake-Shake Regularization," 2017, arXiv:1705.07485.

[10] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "CIFAR-10 (Canadian Institute for Advanced Research)," .

[11] Nguyen Xuan Vinh, Sarah M. Erfani, Sakrapee Paisitkriangkrai, James Bailey, Christopher Leckie, and Kotagiri Ramamohanarao, "Training Robust Models Using Random Projection," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[12] Harish Mallidi and Hynek Hermansky, "Novel Neural Network Based Fusion for Multistream ASR," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2016.

[13] Chul Min Lee, Serdar Yildirim, Murtaza Bulut, Abe Kazemzadeh, Carlos Busso, Zhigang Deng, Sungbok Lee, and Shrikanth Narayanan, "Emotion Recognition Based on Phoneme Classes," in *Proceedings of InterSpeech*, 2004.

[14] S. R. Livingstone, K. Peck, and F. A. Russo, "RAVDESS: The Ryerson Audio-Visual Database of Emotional Speech and Song," in *Proceedings of the 22nd Annual Meeting of the Canadian Society for Brain, Behaviour and Cognitive Science*, 2012.

[15] Olivier Martin, Irene Kotsia, Benoit M. Macq, and Ioannis Pitas, "The eNTERFACE'05 Audio-Visual Emotion Database," in *Proceedings of the International Conference on Data Engineering Workshops*, 2006.

[16] Giovanni Costantini, Iacopo Iaderola, andrea Paoloni, and Massimiliano Todisco, "EMOVO Corpus: an Italian Emotional Speech Database," in *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.

[17] Sana-Ul Haq and Philip J.B. Jackson, *Machine Audition: Principles, Algorithms and Systems*, chapter Multimodal Emotion Recognition, IGI Global, 2010.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, and Georg Stemmer, "The KALDI speech recognition toolkit," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[19] "Pytorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration," 2017, https://github.com/pytorch/pytorch.

[20] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014.