ON MULTITASK LOSS FUNCTION FOR AUDIO EVENT DETECTION AND LOCALIZATION

Huy Phan[*1], *Lam Pham*[2], *Philipp Koch*[3], *Ngoc Q. K. Duong*[4], *Ian McLoughlin*[5], *Alfred Mertins*[3]

[1] School of Electric Engineering and Computer Science, Queen Mary University of London, UK
[2] School of Computing, University of Kent, UK
[3] Institute for Signal Processing, University of Lübeck, Germany
[4] InterDigital R&D France, France
[5] Singapore Institute of Technology, Singapore
[*]Corresponding email: h.phan@qmul.ac.uk

ABSTRACT

Audio event localization and detection (SELD) have been commonly tackled using multitask models. Such a model usually consists of a multi-label event classification branch with sigmoid cross-entropy loss for event activity detection and a regression branch with mean squared error loss for direction-of-arrival estimation. In this work, we propose a multitask regression model, in which both (multi-label) event detection and localization are formulated as regression problems and use the mean squared error loss homogeneously for model training. We show that the common combination of heterogeneous loss functions causes the network to underfit the data whereas the homogeneous mean squared error loss leads to better convergence and performance. Experiments on the development and validation sets of the DCASE 2020 SELD task demonstrate that the proposed system also outperforms the DCASE 2020 SELD baseline across all the detection and localization metrics, reducing the overall SELD error (the combined metric) by approximately 10% absolute.

Index Terms— audio event detection, localization, multitask loss, regression, classification

1. INTRODUCTION

Extended from active research on sound (audio) event detection, sound event localization and detection (SELD) task [1, 2] entangles the *what* and *where* questions about occurring sound events. That is, it aims to determine the identities of the events and their spatial locations/trajectories simultaneously. Solving the SELD task would enable a wide range of novel applications in surveillance, human-machine interaction, bioacoustics, and healthcare monitoring, to mention a few.

The joint SELD task can be divided and conquered individually by two separate models, one for sound event detection (SED) [3, 4, 5] and the other for sound source localization (SSL) [6, 7]. The two-stage approach presented in [8] can be also considered to belong to this line of work. Dealing with the joint task in a single model has been known to be more challenging. Three main approaches have been proposed, including sound-type masked SSL [6], spatially masked SED [9], and joint SELD modeling [10, 2]. Joint sound event detection and localization modeling with multitask deep learning has been most commonly adopted in the latest DCASE challenge [11, 12, 13, 2], demonstrating encouraging results.

In the joint modeling approach with a multitask model, the sigmoid cross-entropy (CE) loss is typically used for event detection (via classification) to handle possible multi-label due to occurrences of multiple events while the mean squared error (MSE) loss is often employed for direction-of-arrival (DOA) estimation (via regression). These two losses are usually associated with different weights and then combined to make the total loss for network training. However, there exist no established rules to set the weights for the losses; more often than not, they are set with some trivial weights without a clear justification. For example, while the DCASE 2019 baseline weighted the MSE loss 50 times larger than that of the sigmoid CE loss, the current DCASE 2020 baseline even enlarges this multiplication to 1000 times. Furthermore, the two different types of loss functions might progress at different rates and might not converge synchronously, making the fixed weights suboptimal. We will empirically show in a controlled experiment that, for this joint modeling task, the classification based on the CE loss usually experiences *underfitting* when being optimized jointly with regression based on the MSE loss.

In order to avoid such an issue, we alternatively propose to formulate both the SED and SSL subtasks as regression problems and homogeneously use the MSE loss for both of them. The proposed multitask-regression network features a recurrent convolutional neural network (CRNN) architecture coupled with self-attention mechanism [14]. Experiments on the development set of the DCASE 2020 Task 3 show that the proposed multitask-regression network results in better generalization than the networks using the combination of

the CE loss and the MSE loss. Furthermore, evaluation on the development and evaluation data of the challenge shows that the proposed network outperforms the DCASE 2020 SELD baseline across all the evaluation metrics, some with a large margin.

2. THE PROPOSED NETWORK

The proposed network is illustrated in Figure 1. The network receives time-frequency input $\mathbf{S} \in \mathbb{R}^{T\times F\times C}$ of T frames, F frequency bins, and C channels. The convolutional part of the network consists of six convolutional layers each of which is followed by a max pooling layer except the first one. We assume that the early convolutional layers are crucial for feature learning, the network is designed to have the first two convolutional layers back-to-back. In order, the six convolutional layers accommodate $\{64, 64, 128, 128, 256, 256\}$ filters, respectively, with a common kernel size of 3×3 and the stride of 1×1. The gradually increasing numbers of filters in the later convolutional layers are to compensate for their smaller feature maps in the frequency dimension. Zero-padding (i.e. *SAME* padding) is used in order to preserve the temporal size. After convolution, batch normalization [15] is applied on the feature maps, followed by Rectified Linear Units (ReLU) activation [16].

The max pooling layers, except the first one, have a common kernel size of 1×2 to reduce the input size by half in the frequency dimension and, by doing so, gain frequency equivariance in the induced feature maps while keeping the temporal size unchanged. Particularly, the pooling kernel size of the first max pooling layer (*max pool 2*, cf. Figure 1) is set to 5×2 in order to reduce the time dimension to $\frac{T}{5}$ to match the frame resolution (100 ms) for computing the evaluation metrics.

Passing through the convolutional block, the input is transformed into a feature map of size $\frac{T}{5} \times 2 \times 256$ which is reshaped to form a sequence of feature vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\frac{T}{5}})$ where $\mathbf{x}_i \in \mathbb{R}^{512}$, $1 \leq i \leq \frac{T}{5}$. A bidirectional recurrent neural network (biRNN) is then employed to iterate through the sequence and encode it into a sequence of output vectors $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{\frac{T}{5}})$. The biRNN is realized by Gated Recurrent Unit (GRU) cells with the hidden size of 256. To further improve encoding the context around a feature $\mathbf{z}_i$, self-attention mechanism [14] is used. The vectors $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{\frac{T}{5}})$ can be viewed as as a set of *key-value* pairs $(\mathbf{K}, \mathbf{V})$. In the context of this work, both the keys and values coincide to $\mathbf{Z}$ (the concatetation of the $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{\frac{T}{5}}$ vectors). We adopt the scaled dot-product attention as in [14], i.e. the attention output at a time index is a weighted sum of $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{\frac{T}{5}}$ where the weights are determined as

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{d_k}})\mathbf{V}. \quad (1)$$

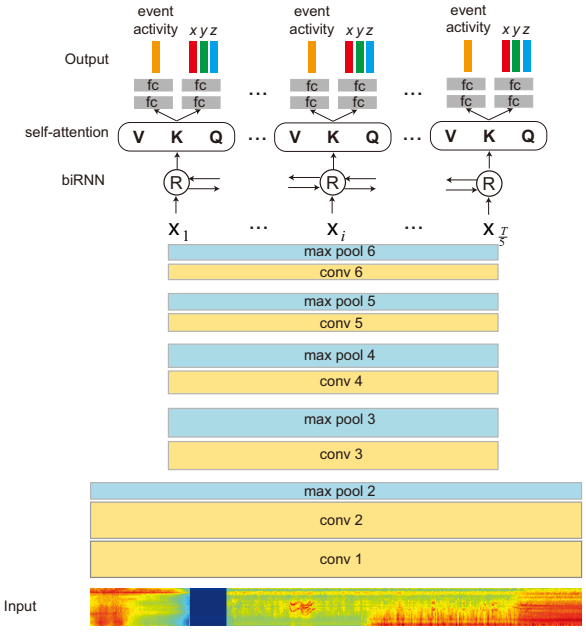

Figure 1: Overview of the proposed multitask regression self-attention CRNN.

Here, $\mathbf{Q}$ is the *query* [14] and also coincides to $\mathbf{Z}$ in the context of this work, i.e. $\mathbf{Q} \equiv \mathbf{K} \equiv \mathbf{V} \equiv \mathbf{Z}$. d_k is the extra dimension into which $\mathbf{Q}$, $\mathbf{K}$ are transformed before the dot product to prevent the inner product from becoming too large. d_k is set to 64 in this work.

At each time index, the SED and SSL subtasks are accomplished via two network branches, each consisting of two fully connected (fc) layers with 512 units each. The first branch's output layer has $\mathcal{Y}$ units with *sigmoid* activation to perform event activity classification/regression of $\mathcal{Y}$ classes. The second branch has $3\mathcal{Y}$ units with *tanh* activation to regress for the target events' DOA trajectories. Normally, when the sigmoid CE loss is used for event activity classification and the MSE loss is used for the DOA estimation, the network is trained to minimize the following weighting loss:

$$\mathcal{L}_{\text{CE+MSE}}(\Theta) = -\, w_{\text{CE}} \sum_{n=1}^{N}\sum_{t=1}^{\frac{T}{5}} (\mathbf{y}_{nt}\log(\hat{\mathbf{y}}_{nt}) + (1-\mathbf{y}_{nt})\log(1-\hat{\mathbf{y}}_{nt})) + w_{\text{MSE}} \sum_{n=1}^{N}\sum_{t=1}^{\frac{T}{5}} ||\hat{\mathbf{d}}_{nt}(\Theta) - \mathbf{d}_{nt}||^2. \quad (2)$$

Here, Θ denotes the network parameters and N denotes the number of training examples. We use $\hat{\mathbf{y}}$ and $\mathbf{y}$ to denote the event activity output and grouthtruth, respectively. In addition, we used $\hat{\mathbf{d}} = (\hat{x}, \hat{y}, \hat{z})$ and $\mathbf{d} = (x, y, z)$ to denote the DOA estimation output and groudtruth in terms of Cartesian coordinates on the unit sphere, respectively. w_{CE} and w_{MSE} indicate the weights given to the corresponding losses.

On the other hand, when the MSE loss is used for both SED and SSL subtasks, the network is trained to minimize the total MSE loss of the two network branches without

weighting:

$$\mathcal{L}_{\text{MSE}}(\Theta) = \sum_{n=1}^{N}\sum_{t=1}^{\frac{T}{5}}(||\hat{\mathbf{y}}_{nt}(\Theta) - \mathbf{y}_{nt}||^2 + ||\hat{\mathbf{d}}_{nt}(\Theta) - \mathbf{d}_{nt}||^2). \quad (3)$$

3. EXPERIMENTS

3.1. DCASE 2020 SELD dataset

The database used for the DCASE 2020 SELD task was synthesized in two spatial sound formats: (1) MIC - 4-channel microphone array extracted from a subset of 32-channel Eigenmike format and (2) FOA - 4-channel first-order Ambisonics extracted from a matrix of 4×32 conversion filters. 714 sound examples from the published NIGENS General Sound Events Database [1] of 14 event classes, including *alarm, crying baby, crash, barking dog, running engine, burning fire, footsteps, knocking on door, female & male speech, female & male scream, ringing phone,* and *piano*, were used for data creation. More information about the data synthesis can be found in [1]. The database was split into eight sets, six of which were used as the development set and the remaining two were used as the evaluation set.

Experiments on the development set: We followed the challenge setup to conduct experiments on the development set. That is, the first set of the development data was used as the unseen data for testing purpose, the second set was used as the validation set for model selection, and the remaining four sets were used as the training data.

Experiments on the evaluation set: To assess performance on the evaluation set, two different systems were trained and submitted to the challenge. The first was trained using the first set of the development data as validation set for model selection and the remaining five sets as the training data (**Submission 1**). The second was trained using the entire development data as the training data (i.e. without validation data for model selection) (**Submission 2**).

3.2. Feature extraction

We extracted log-Mel magnitude spectrogram with a window size of 40 ms, 20 ms overlap, and 64 Mel-bands. To encode the phase information, for the FOA data, an acoustic intensity vector was extracted for each Mel-band, whereas, for the MIC data, generalized-cross-correlation with phase-transform (GCC-PHAT) features were computed for each Mel-band. Overall, multi-channel images of size $3000 \times 64 \times 7$ and $3000 \times 64 \times 10$ were resulted for one-minute FOA and MIC recordings, respectively.

3.3. Parameters

Network implementation was based on *Tensorflow* framework. We used spectrogram segments of size $T = 600$ (equivalent to 12 seconds) as inputs. *Dropout* rates of 0.5, 0.1, and 0.25 were employed to regularize the convolutional layers, the biRNN, and the fully-connected layers, respectively.

The network was trained using *Adam* optimizer [17] for 10000 epochs with a minibatch size of 64. Each spectrogram segment in a minibatch was randomly sampled from a 1-minute recording and augmented using spectrogram augmentation [18]. The learning rate was initially set to 2×10^{-4} and was exponentially reduced with a rate of 0.8 after 200, 600, and 1000 epochs. In addition, the first 10 epochs were used as a warmup period in which the network was trained with a small learning rate of 2×10^{-5}.

During training, the network snapshot that achieved the lowest combined SELD error rate on the validation set was retained for evaluation. The retained network was then evaluated on the test recordings with a 2-second segment at a time without overlap. To be able to analyze the effect of using different loss combinations in a controllable manner, no post-processing was carried out. Event activity was determined from the corresponding regression/classification output using a threshold of 0.5.

3.4. Evaluation metrics

The DCASE 2020 challenge evaluated the performance of the SED subtask using localization-aware detection error rate (ER_{20°) and F-score (F_{20°) with a threshold of 20° in one-second non-overlapping segments. For sound event localization, errors only between same-class predictions and references were considered. The class-aware localization error (LE_{CD}) and its corresponding recall (LR_{CD}) were employed for evaluating localization outputs and were also computed in one-second non-overlapping segments. In addition, we also computed the combined SELD error metric:

$$SELD = \frac{1}{4}(ER_{20^\circ} + (1 - F_{20^\circ}) + \frac{LE_{CD}}{180} + (1 - LR_{CD})) \quad (4)$$

to give an overall picture about a system.

3.5. Experimental results

3.5.1. Influence of the loss functions

It is a rule of thumb that the CE loss is preferred over the MSE loss for a classification task since it, in general, leads to quicker learning through gradient descent, at least theoretically [19]. However, when it is used in combination with the MSE task as in (2) as commonly used for joint SELD, it apparently underfits the data as evidenced in Figure 2. When an equal weight is used for the two losses in (2), i.e. $w_{\text{CE}} = w_{\text{MSE}} = 1$, the CE loss (cf. Figure 2 (c)) and the SED error (cf. 2 (d)) are hard to be reduced on both the training and test data (note the scale of the CE loss in Figure 2 (c) is much larger than that of the MSE loss in Figure 2 (a)). The underfitting effect on the SED subtask is even worse under the skewed weighting scheme used in the DCASE 2020 baseline [1], i.e. the MSE loss was given a weight of 1000.0 and the CE loss was given a weight of 1.0, since in this case

[1] https://zenodo.org/record/2535878

Table 1: Results obtained by the proposed system and the DCASE 2020 baseline on the development and evaluation sets.

	DOA loss (weight)	SED loss (weight)	FOA					MIC				
			LE_{CD}	LR_{CD}	ER_{20°	F_{20°	$SELD$	LE_{CD}	LR_{CD}	ER_{20°	F_{20°	$SELD$
Development results												
Val (DCASE2020)	MSE (1000)	CE (1)	23.5°	62.0	0.72	37.7	0.46	27.0°	62.6	0.74	34.2	0.48
Val (CE+MSE)	MSE (1000)	CE (1)	*16.1°*	*51.7*	*0.83*	*41.4*	*0.50*	*16.5°*	*51.1*	*0.82*	*42.6*	*0.49*
Val (CE+MSE)	MSE (1)	CE (1)	*24.1°*	*67.6*	*0.78*	*42.8*	*0.45*	*27.9°*	*66.6*	*0.86*	*34.7*	*0.50*
Val (MSE)	MSE	MSE	**17.7°**	**68.1**	**0.58**	**52.4**	**0.37**	**17.3°**	**66.0**	**0.56**	**53.9**	**0.37**
Test (DCASE2020)	MSE (1000)	CE (1)	22.8°	60.7	0.72	37.4	0.47	27.3°	59.0	0.78	31.4	0.51
Test (CE+MSE)	MSE (1000)	CE (1)	*18.0°*	*50.6*	*0.88*	*38.9*	*0.53*	*16.7°*	*53.6*	*0.81*	*44.3*	*0.48*
Test (CE+MSE)	MSE (1)	CE (1)	*26.2°*	*62.7*	*0.82*	*39.9*	*0.49*	*28.3°*	*60.0*	*0.93*	*31.2*	*0.54*
Test (MSE)	MSE	MSE	**19.0°**	**65.6**	**0.60**	**49.2**	**0.39**	**18.2°**	**64.1**	**0.59**	**50.8**	**0.38**
Evaluation results												
DCASE2020	MSE (1000)	CE (1)	20.5°	65.0	0.66	43.3	0.42	21.8°	65.9	0.66	44.0	0.42
Submission 1	MSE	MSE	**16.8°**	**69.8**	**0.52**	**57.8**	**0.33**	**14.6°**	**68.2**	**0.55**	**58.8**	**0.34**
Submission 2	MSE	MSE	**15.2°**	**72.4**	**0.49**	**61.7**	**0.31**	**14.6°**	**68.2**	**0.53**	**59.2**	**0.33**

the network further prioritizes optimizing the MSE loss over the CE one. We speculate that a similar phenomenon happened to the DCASE 2020 baseline as it results in limited performance on the SED subtasks (cf. Table 1).

In contrast, when the MSE loss is used for both the SED and SSL subtasks as in (3), the SED performance is improved significantly (cf. Figure 2 (d)) while the DOA estimation performance remains comparable to the case of MSE+CE combination (cf. Figure 2 (b)). These results suggest that the SELD multitask network learns easier when a homogeneous loss is used for all the subtasks than when heterogeneous losses are combined. Although we cannot conclude that the MSE loss is the optimal loss for SELD multitask modeling, these results urge the quest for one in future work.

3.5.2. SELD performance

The performance obtained by the studied systems on the development and evaluation data are shown in Table 1. As expected, using the MSE error homogeneously consistently results in much better performance than the MSE+CE combinations. In addition, the proposed system outperforms the DCASE 2020 SELD baseline across the evaluation metrics, particularly on the SED metrics. This is most likely due to the underfitting effect on the SED subtask of the baseline, making it underperforming on this subtask. Overall, using FOA and MIC data, the proposed system reduces the combined SELD error by 0.08 and 0.11 absolute on the development data from that of the baseline, respectively. The corresponding error reduction by **Submission 2** on the evaluation data reaches 0.11 and 0.09, respectively.

Our submission to the DCASE 2020 Task 3 was ranked 6$^{\text{th}}$ overall. This is an encouraging result given that the submission systems were compact and neither relied on ensemble nor multiple microphone arrays [2].

[2] http://dcase.community/challenge2020/task-sound-event-localization-and-detection-results

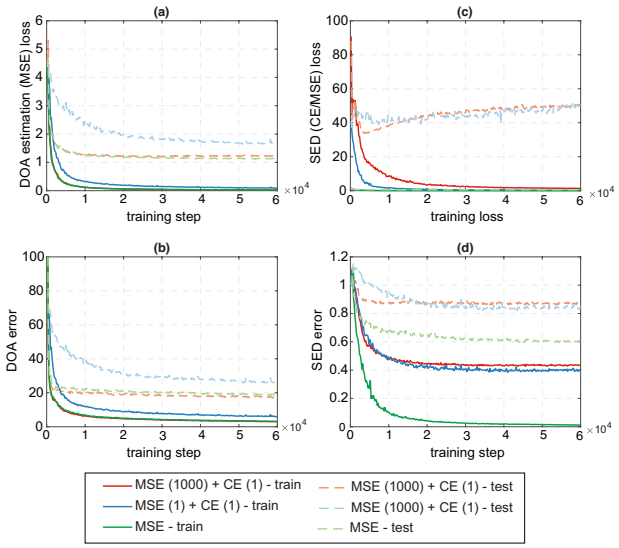

Figure 2: Variation of the MSE loss, the CE loss, the DOA error, and the SED error on the training and test sets of the DCASE 2020 development data with different loss combinations. (a) The MSE loss, (b) the CE loss, (c) the DOA error, and (d) the SED error. The number in bracket indicates the weight assigned to the corresponding loss.

4. CONCLUSIONS

This work investigated the loss functions used for SELD multitask modeling. We showed empirical evidence that the combination of the sigmoid CE loss (for the SED subtask) and the MSE loss (for the DOA estimation subtask), which is commonly used, often results in underfitting effect on the former. As an alternative, when the two subtasks were formulated as regression problems and the MSE loss was used for both, the multitask network was able to converge better, resulting in better and balanced performance. Experimental results on the development and evaluation set of the DCASE 2020 SELD task showed significant improvements over the DCASE 2020 baseline across all the evaluation metrics.

5. REFERENCES

[1] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," *arXiv preprint 2006.01919*, 2020.

[2] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[3] H. Phan, O. Y. Chén, P. Koch, L. Pham, I. McLoughlin, A. Mertins, and M. D. Vos, "Unifying isolated and overlapping audio event detection with multi-label multi-task convolutional recurrent neural networks," in *Proc. ICASSP*, 2019.

[4] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 5, no. 6, pp. 1291–1303, 2017.

[5] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao, and H. Phan, "Continuous robust sound event classification using time-frequency features and deep learning," *PLoS ONE*, vol. 12, no. 9, 2017.

[6] N. Ma, J. A. Gonzalez, and G. J. Brown, "Robust binaural localization of a target sound source by combining spectral source models and deep neural networks," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 11, p. 2122–2131, 2018.

[7] R. Chakraborty and C. Nadeu, "Sound-model-based acoustic source localization using distributed microphone arrays," in *Proc. ICASSP*, 2014, p. 619–623.

[8] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. Plumbley, "Polyphonic sound event detection and localization usinga two-stage strategy," in *Proc. Detection and Classification ofAcoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.

[9] I. Trowitzsch, C. Schymura, D. Kolossa, and K. Obermayer, "Joining sound event detection and localizationthrough spatial segregation," in *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, 2020, pp. 487–502.

[10] W. He, P. Motlicek, and J.-M. Odobez, "Joint localization and classification of multiple sound sources using a multi-task neural network," in *Proc. Interspeech*, 2018.

[11] F. Grondin, I. Sobieraj, M. Plumbley, and J. Glass, "Sound event localization and detection using crnn on pairs of microphones," in *Detection and Classification of Acoustic Scenesand Events 2019 Workshop (DCASE2019)*, 2019.

[12] H. Cordourier, P. L. Meyer, J. Huang, J. D. H. Ontiveros, and H. Lu, "Gcc-phat cross-correlation audio featuresfor simultaneous sound event localization and detection(seld) on multiple rooms," in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.

[13] S. Kapka and M. Lewandowski, "Sound source detection, localizationand classification using consecutive ensemble ofcrnn models," in *Procc. Detection and Classification of AcousticScenes and Events 2019 Workshop (DCASE2019)*, 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[17] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–13.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[19] M. Nielsen, *Neural Networks and Deep Learning*, 2019, ch. Improving the way neural networks learn.