Robust Online Multi-Channel Speech Recognition

Markus Kitza[1]*, Albert Zeyer, Ralf Schlüter*

Human Language Technology and Pattern Recognition, RWTH Aachen, 52074 Aachen
Email: {kitza,zeyer,schluter}@i6.informatik.rwth-aachen.de
Web: www-i6.informatik.rwth-aachen.de

Jahn Heymann[1]*, Reinhold Haeb-Umbach*

Department of Communications Engineering Paderborn, Paderborn University, 33098 Paderborn
Email: {heymann,haeb}@nt.uni-paderborn.de
Web: nt.uni-paderborn.de

Abstract

In this paper we present a system for robust online far-field multi-channel speech recognition with minimal assumptions on microphone configuration and target location. We employ an online-enabled Generalized Eigenvalue (GEV) beamformer and a Long Short-Term Memory (LSTM) network to robustly calculate the signal statistics necessary for the beamforming operation in the front-end. After multiple channels have been condensed to one, a Bidirectional Long Short-Term Memory (BLSTM) acoustic model is applied on a running window of input speech. This enables online decoding in combination with the beamforming front-end. To assess the performance of the system we test it on the real evaluation set of the CHiME 3 data where we achieve a Word Error Rate (WER) of 10.4 %.

1 Introduction

The recently held CHiME 3 challenge showed how modern Automatic Speech Recognition (ASR) systems can successfully exploit multi-channel input data [1]. Despite severe background noise, the best performing system achieved a Word Error Rate (WER) well below 6% on the evaluation data [2]. Such performance helps to improve the usability of hands-free speech controlled devices which are often equipped with multiple microphones and whose usage is becoming more and more popular among consumers. To achieve this performance, [2] utilizes a Minimum Variance Distortionless Response (MVDR) beamformer in combination with a strong Convolutional Neural Network (CNN) back-end. Unlike the baseline system, this system avoided the calculation of the Time Differences of Arrival (TDOA) and relied on signal statistics instead, which are given by the estimated Cross-Power Spectral Density (PSD) matrices of the target signal and noise. These were obtained by calculating the Speech Presence Probability (SPP) for every time frequency (tf)-bin. Our submission to the challenge used the same idea for the front-end, but instead of a MVDR beamformer, we employed a Generalized Eigenvalue (GEV) beamformer. We also used a different method to calculate the SPP: Instead of relying on a model-based approach, we trained a neural network to carry out the task. However, the baseline back-end was left unmodified.

Yet, the approach in [2] required multiple decoding passes with multiple networks to achieve its impressive performance. In fact, most of the top-scoring systems in the challenge used techniques not available in an online setting. This renders those approaches impracticable for applications which require immediate feedback. The goal of the study described here, is to achieve comparable results with an online-enabled system as required for such applications, while making as few assumptions about the scenario as possible.

In [3] the best performing system was altered to enable an online usage with only a slightly increased WER. This was achieved by using a speaker independent CNN acoustic model and an online update of the SPP estimation: The PSD matrices were initialized from the available training data and then subsequently updated using a Complex Gaussian Mixture Model (CGMM) model. This initialization, however, might be a possible drawback as it implicitly assumes a certain target position at the beginning (i.e., the ones found in the training). While this works well for the CHiME 3 data where the speakers position is more or less the same for every utterance, it might be problematic for use-cases where the device position is fixed but the speaker position changes every time the device is used. This is for example a common scenario for smart home devices.

In [4] it was shown that Bidirectional Long Short-Term Memory (BLSTM) networks give strong improvements in performance as opposed to feedforward Deep Neural Network (DNN) which are limited to a local future context. Further BLSTM networks can be used within a moving window without much loss of performance, thereby removing the advantage of straight forward online decoding of feedforwards DNNs.

In this work we present our approach to an online ASR system which, albeit having a similar setup as [3], does not make any prior assumptions regarding the target position. We organize the paper as follows: In Section 2 we give an overview of our system with a focus on the front-end. We then extensively evaluate this system in Section 3 before drawing our conclusions in Section 4.

2 System Overview

2.1 Front-End

We model an observed signal $\mathbf{Y}$ in the Short Time Fourier Transform (STFT)-domain as the superposition of the target image $\mathbf{X}$ and distortions $\mathbf{N}$. These distortions might be introduced by noise sources or by reverberation effects:

$$\mathbf{Y}(t,f) = \mathbf{X}(t,f) + \mathbf{N}(t,f), \tag{1}$$

where $t \in \{1,\dots T\}$ is the time frame index and $f \in \{1,\dots F\}$ is the frequency bin index. In the following we will omit the frequency index f for brevity.

2.1.1 GEV Beamformer

The GEV beamformer is used to suppress the distortions. It maximizes the signal-to-noise ratio (SNR) of the beamformer output in each frequency bin separately, leading to

[1] Both main authors contributed equally.

ISBN 978-3-8007-4275-2

Table 1: LSTM network configuration for mask estimation

Layer	Units	Type	Non-Linearity	p_{dropout}
L1	1024	LSTM	Tanh	0.5
L2	1024	FF	ELU	0.5
L3	1024	FF	ELU	0.5
L4	1024/1026	FF	Sigmoid	0.0

the beamformer coefficients [5]:

$$\mathbf{F}_{\text{GEV}} = \underset{\mathbf{F}}{\text{argmax}} \frac{\mathbf{F}^{\text{H}}\Phi_{\mathbf{XX}}\mathbf{F}}{\mathbf{F}^{\text{H}}\Phi_{\mathbf{NN}}\mathbf{F}}. \tag{2}$$

The optimal filter coefficient vector $\mathbf{F}_{\text{GEV}}$ is given by the eigenvector corresponding to the largest eigenvalue of the following generalized eigenvalue problem:

$$\Phi_{\mathbf{XX}}\mathbf{F}_{\text{GEV}} = \lambda\Phi_{\mathbf{NN}}\mathbf{F}_{\text{GEV}}. \tag{3}$$

Hence, the solution only relies on the signal statistics: the target PSD matrix $\Phi_{\mathbf{XX}}$ and the noise PSD matrix $\Phi_{\mathbf{NN}}$. For the online scenario these statistics are estimated in a block-wise fashion. We initialize the PSD matrices with a scaled identity matrix (i.e., we do not make any assumptions on the targets position) and subsequently update them as follows:

$$\tilde{\Phi}_{\nu\nu}(kM) = \tilde{\Phi}_{\nu\nu}((k-1)M) + \sum_{i=(k-1)M}^{kM-1} M_\nu(i)\,\mathbf{Y}(i)\,\mathbf{Y}(i)^{\text{H}}$$

$$\Phi_{\nu\nu}(k) = \frac{\tilde{\Phi}_{\nu\nu}(k)}{\sum_{i=0}^{kM} M_\nu(i)} \tag{4}$$

Here, k describes the block index, M the block size, $\nu \in \{\mathbf{X}, \mathbf{N}\}$ indicates which signal statistics we want to estimate and M_ν indicate the tf bins where the signal ν is dominant. These statistics are then used to calculate new beamforming coefficients $\mathbf{F}_{\text{GEV}}$ for the k-th block. Note that we solve eq. 3 for each block individually and do not apply any smoothing technique.

2.1.2 Mask Estimation Network

The online scenario required us to modify the network configuration for spectral mask used in our previous work. We replace the BLSTM with a uni-directional Long Short-Term Memory (LSTM) neural network and increase the number of units. We found the latter to be necessary for a reliable estimation. The complete configuration of the network is shown in Table 1.

The scenario also prohibits us to use batch-normalization that was used in our previous works where we estimated the statistics over a whole utterance, even at test time. One way to cope with this problem would be to estimate the statistics using the training corpus and keep those values fixed during testing as proposed in [6]. However, this deteriorated the masks severely leading to very poor beamforming results even when we just applied it to the first layer. Another possible solution we tried was to abdicate normalization completely. But this led to even worse performance at test time.

We finally settled with a fixed input normalization at training time and a running average normalization at test time for the first layer only. We estimate the mean $\bar{\mathbf{a}}$ and variance $\sigma_{\mathbf{a}}$ of the lateral LSTM connection for the k-th block according to

$$\begin{aligned}
\hat{\mathbf{a}} &= \sum_{i=t-M}^{t} \frac{\mathbf{a}(i)}{M} \\
\bar{\mathbf{a}}(k) &= \bar{\mathbf{a}}(k-1)\frac{k-1}{k} + \frac{1}{k}\hat{\mathbf{a}} \\
\sigma_{\mathbf{a}}(k) &= \sigma_{\mathbf{a}}(k-1)\frac{k-1}{k} + \frac{1}{k}\sum_{i=t-M}^{t}(\mathbf{a}(i)-\hat{\mathbf{a}})^2
\end{aligned} \tag{5}$$

and use these values to normalize the activations. Additionally we replace the Rectified Linear Unit (ReLU) activation function with the Exponential Linear Unit (ELU) activation function which has an effect similar to batch-normalization during training as discussed in [7].

2.1.3 Masks for Voice Activity Detection

For most of the audio files, the first few 100ms contain no speech from the target. Since we need a target to do beamforming in the first place, we use the target mask $\Phi_{\mathbf{XX}}$ as an indication of speech activity. As long as the cumulative sum of all of its elements is under a certain threshold (1000 in this work), we keep accumulating the frames and do not perform any beamforming or recognition. Once the threshold is reached, we calculate the first beamforming vector and perform the beamforming operation on all frames accumulated so far. We opted to use this method instead of a fixed first blocksize in order to keep the delay at a minimum since for some utterances, there can be speech even in the first few frames.

2.2 Back-End

The online ASR back-end is constructed as follows. During feature extraction, normalization is applied by shifting a window. The BLSTM acoustic model is trained in an offline fashion. But during recognition the BLSTM acoustic model is computed on a running window as outlined in more detail in the following section. After a first pass, a second pass is done applying speaker adaptation using the Constrained Maximum Likelihood Linear Regression (CMLLR) transformation.

2.2.1 Acoustic Model

Recently, LSTMs yield state-of-the-art results for acoustic modeling and perform better than feedforward DNNs [8–10]. It also has been shown that bidirectional LSTMs perform better than unidirectional ones [11]. However, applying BLSTMs in an online setting is not straight-forward. In [4] we demonstrated one approach to use BLSTM acoustic models for online recognition which yields almost the same recognition performance as with offline BLSTM recognition.

In offline BLSTM recognition, we would do the forwarding through the BLSTM given the whole input sequence $x_1^T \in \mathbb{R}^{T\times D}$ to estimate the posterior probabilities $p_t(s|x_1^\infty)$ for all output states s and all $t \in \{1,\ldots,T\}$.

In our approach to online BLSTM recognition, we operate only on a moving window over an input sequence of unknown length $x_1^\infty \in \mathbb{R}^{\mathbb{N}\times D}$. We always move the window of fixed size T_w by T_s time frames, i. e. the i-th window is

$X_i := x_{1+i\cdot T_s}^{T_w+i\cdot T_s}$ with $i \in \mathbb{N}_0$. Note that this results in up to $\lceil T_w/T_s \rceil$ overlapping windows for any specific time frame. For each window X_i, we do one forwarding through the BLSTM to get the posterior probabilities $p_t(s|X_i)$ for the corresponding time frames t. We estimate the final acoustic model posterior probability $q_t(s|x_1^\infty)$ by averaging over the individual window posterior estimations with an optional weighting scheme for each time frame in a window, i. e. we calculate

$$q_t(s|x_1^\infty) = \frac{\sum_{i=0}^{\infty} W_{i\cdot T_s}(t) \cdot p_t(s|X_i)}{\sum_{i=0}^{\infty} W_{i\cdot T_s}(t)} \tag{6}$$

where $W_\tau(t) \in \mathbb{R}_{\geq 0}$ is the weighting for time frame $t \in \{\tau+1, \ldots, \tau+T_w\}$ in the window $x_{\tau+1}^{\tau+T_w}$.

We had the intuition that the BLSTM probability estimations of the center frames of a window might be better than at the edges and we can use the weighting to let them contribute more for the final estimation. In [4] we did several experiments with different weighting schemes. We found the weighting scheme to have a lesser effect than expected. Nevetheless a triangle weighting was observed to be slightly better than a uniform distribution, therefore we use it in this work.

We also found that a window size of about $T_w = 100$, i.e. 1 second is enough, but given that, more important is to have a high amount of overlaps like $\lceil T_w/T_s \rceil = 20$.

Note that each forwarding of each window is independent and thus can be parallelized.

2.2.2 Speaker Adaptation

In an online scenario only previously seen data is available for each algorithm. To estimate CMLLR transformation matrices, a context, which is longer than an average sentence, is required. Therefore different utterances of the same speaker need to be used in conjunction for the estimation. If the speaker is known, only segments previously processed from the target speaker, excluding the current segment, are used to compute the adaptation matrices. This will be denoted as sequential CMLLR (sCMLLR). In case the speaker is not known, an automatic clustering algorithm needs to be applied. A straight forward approach would be online k-means clustering [12]. In this work, supervised speaker information was available and used for adaptation.

3 Experimental Results

3.1 Experimental Setup and Baseline Systems

To evaluate our approach, a series of experiments were conducted on the 3rd CHiME Challenge [13] dataset. The CHiME 3 scenario is about automatic speech recognition for a multi-microphone tablet device being used in everyday environments. Four different environments have been defined: cafe, street junction, public transport and pedestrian area. For each environment, two types of noisy speech data have been provided, real and simulated. The real data consists of 6-channel recordings of sentences from the WSJ0 corpus [14] spoken within real environments. The simulated data was constructed by mixing clean utterances into background recordings of the above environment. In this work we use nn-gev [15, 16] for beamforming the audio channels and both RASR [17, 18] and RETURNN [10, 19] for building the speech recognition system.

The training data of the system consist of all six microphone channels as well as the beamformed data presented sequentially in a single channel. Each channel contains 15 hours of training data. The baseline system uses 16-dimensional MFCC features and has 1501 tied states. The acoustic model LSTM baseline has 5 layers with 600 hidden units and is trained using ADAM [20] on 16-dimensional MFCC features only. For regularization, three methods are used in conjunction. Dropout with 10% probability and L2-regularization with a weight of 0.001 are applied at each hidden layer. In addition gradient noise [21] with a weight of 0.3 is used during training. The online models are trained by importing the weights of the offline baseline and retraining the model with the online data coming from the beamformer. In the case of online processing, the normalization window has a length of 1 second and the windows of the acoustic model are also 1 second in size but are shifted by 50 ms. This results in an overlap of 20 windows and also increases the computational complexity by the same amount. Increasing the chunk-size and reducing the step-size would increase performance but degrade the real-time factor and increase delay.

3.2 Evaluation of Proposed Online Algorithms

The proposed online decoding scheme is evaluated in this section. All word error rates depicted are from beamformed data of the real condition evaluation set of CHiME 3. It contains 4 speakers each recorded in four environments adding up to 130 minutes of data. Every setting was evaluated on a 3-gram language model provided by the 3rd CHiME Challenge without neural network rescaling.

3.2.1 Influence of the beamformer

The proposed online beamformer has two parameters influencing its performance, namely the blocksize and the threshold (see Section 2.1.3). We also introduce an optional postfilter: The Blind Analytic Normalization (BAN) filter redistributes the signal energies to remove distortions introduced by the beamforming operation [5]. The reasoning behind this is that energies distributed more similar to the noisy data (which represents 85% of the training data) might result in increased performance. We evaluated the influence of those parameters and report the resulting WER on the real development data in Table 3. The acoustic model was a 3 layer LSTM with 500 hidden units.

Contrary to our expectations, the postfilter does not increase the performance. Instead, the results are actually slightly worse. The evaluation also reveals that a threshold of 100 is too low. We therefor recommend to use 1000 as a good trade-off between performance and delay. With this threshold, the blocksize is also not too critical as long as it is bigger than one frame. Here, 10 is a good choice to balance delay and computational complexity. We use this configuration for all following experiments.

The difference between online and offline beamforming is visible in Table 3. Given the same acoustic model and normalization, the WER increases by roughly 3 percent points. Further investigations reveal, that the masks used to estimate the PSD matrices do not differ much. However, we observe that there is a noticeable difference between the cosine distance of the beamforming vectors of the online and offline version at the beginning of an utterance which mostly vanishes half way through the utterance.

ISBN 978-3-8007-4275-2

Table 2: Comparison of successively replacing baseline parts with proposed online system on real evaluation dataset of CHiME3. The evaluation was performance on a 5 layer LSTM for the speaker adapted scenarios and a 7 layer LSTM for the speaker independent cases. In both cases 600 hidden units were used.

Normalization	Beamformer	Acoustic Model	WER [%] with speaker adaptation none	sCMLLR	CMLLR
Offline	Offline	Offline	8.6	7.1	6.9
Online		Online	9.2		
		Offline	9.4		
	Online	Online	12.1	10.4	
		Offline	12.7		

Table 3: WER on development set for different beamformer configurations on a 3 layer LSTM with 500 hidden units.

Postfilter	Threshold	Blocksize 1	5	10	15
yes	100	12.2	12.1	11.7	11.6
	1000	11.4	11.3	11.0	11.0
no	100	12.2	11.6	11.5	11.4
	1000	11.4	11.0	**11.0**	11.0

Table 5: Comparison between optimal layer count at 600 hidden units for adapted and unadapted training data.

Adaptation	Normalization	# of layers	WER %
CMLLR	Offline	5	6.9
		6	7.6
		7	7.6
none	Online	5	10.6
		6	10.4
		7	9.4

3.2.2 Influence of Adaptation

In this section the viability of sequential CMLLR and the influence of unadapted training data on the complexity of the model are investigated.

Table 4: Impact on WER of number of segments used per speaker for estimation of CMLLR matrices on a 3 layer BLSTM with 500 hidden units. The WER is an average of 4 speakers with 330 segments each.

# of segments	WER %
1	14.7
5	11.1
10	10.9
30	10.5
20	10.2
30	10.0
80	10.0
330	9.9

Table 4 depicts the impact on the WER by limiting the number of available segments for estimation of CMLLR matrices. The acoustic model, which is a 3 layer BLSTM with 500 hidden units, was trained with CMLLR matrices estimated on all segments of the target speaker. During recognition only a subset of segments are used. It can be seen that after 10 segments, which on average equals roughly 60 seconds of audio data, the loss in WER is about 10%. But after 3 minutes there is no more loss in performance. This makes sequential CMLLR a viable technique for adaptation in online scenarios.

Table 5 shows the ability of LSTMs to handle the increased variability of unadapted data. All investigated networks had 600 hidden units. In case of adapted training data, the performance peaks at 5 layers with 6.9% WER and degrades slightly if more layers are added. But in case of unadapted data, further improvements can be gained by increasing the complexity of the model. Models with a higher number of layers are able to better handle the increased diversity in the data.

3.2.3 From offline to online

In this section the influence of successively replacing each component of the offline baseline with a corresponding online component will be investigated. Table 2 shows the WER for different combinations of baseline and the proposed online components. The offline model has a strong baseline of 6.9% WER which is only diminished slightly to 7.1% by applying sequential CMLLR. Switching from a segment-wise normalization to a moving window decreases performance by a relative 9%. Applying the online beamformer preprocessing degrades the performance a additional 30%. But applying the running window decoding method even improves performance by a slight amount, but at the cost of highly increased computational complexity.

Conclusively switching from online to offline processing provides an improvement of 10.1 % to 6.9 % (34% relative improvement)

4 Conclusions

In this paper we propose a multi-channel far field speech recognition system which is capable of online decoding from the preprossessing to the acoustic model with minimal assumptions on the speakers position and microphone array geometry. We investigated the impact on the performance for each part going from offline to online processing and found that the beamformer is the biggest bottleneck while evaluating BLSTMs with averages of moving windows increases the performance of the acoustic model compared to evaluation on the whole sequence. Overall, the performance difference switching from an offline based system to the online one are 3.5 percent points in WER.

ISBN 978-3-8007-4275-2

References

[1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Scottsdale, AZ, USA), pp. 504–511, Dec 2015.

[2] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Scottsdale, AZ), pp. 436–443, Dec. 2015.

[3] T. Higuchi, N. I. T. Yoshioka, and T. Nakatani, "Robust mvdr beamforming using time-frequency masks for online/offline asr in noise," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Shanghai, China), Apr. 2016.

[4] A. Zeyer, R. Schlüter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," *submitted to Interspeech 2016*, 2016.

[5] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1529–1539, July 2007.

[6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)* (D. Blei and F. Bach, eds.), pp. 448–456, JMLR Workshop and Conference Proceedings, 2015.

[7] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[8] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[9] J. T. Geiger, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *INTERSPEECH*, (Singapore), pp. 631–635, 2014.

[10] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," *submitted to Interspeech 2016*, 2016.

[11] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[12] E. Liberty, R. Sriharsha, and M. Sviridenko, "An algorithm for online k-means clustering," *arXiv preprint arXiv:1412.5721*, 2014.

[13] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*, (Scottsdale, AZ, USA), Dec 2015.

[14] J. Garofalo *et al.*, "CSR-I (WSJ0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[15] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "Blstm supported gev beamformer front-end for the 3rd chime challenge," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Scottsdate, AZ), pp. 444–451, Dec 2015.

[16] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Shanghai, China), 2016.

[17] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Florence, Italy), pp. 3313–3317, May 2014.

[18] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - the RWTH Aachen university open source speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop*, (Waikoloa, HI, USA), Dec. 2011.

[19] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," *submitted to Interspeech 2016*, 2016.

[20] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014.

[21] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding Gradient Noise Improves Learning for Very Deep Networks," *arXiv preprint arXiv:1511.06807*, Nov. 2015.

ISBN 978-3-8007-4275-2