INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

MatchboxNet: 1D Time-Channel Separable Convolutional Neural Network Architecture for Speech Commands Recognition

Somshubra Majumdar, Boris Ginsburg

NVIDIA, Santa Clara, USA
{smajumdar,bginsburg}@nvidia.com

Abstract

We present *MatchboxNet* - an end-to-end neural network for speech command recognition. MatchboxNet is a deep residual network composed from blocks of 1D time-channel separable convolution, batch-normalization, ReLU and dropout layers. MatchboxNet reaches state-of-the art accuracy on the Google Speech Commands dataset while having significantly fewer parameters than similar models. The small footprint of MatchboxNet makes it an attractive candidate for devices with limited computational resources. The model is highly scalable, so model accuracy can be improved with modest additional memory and compute. Finally, we show how intensive data augmentation using an auxiliary noise dataset improves robustness in the presence of background noise.

Index Terms: key word spotting, speech commands recognition, deep neural networks, depth-wise separable convolution

1. Introduction

We present MatchboxNet, a new compact, end-to-end neural network for keyword spotting (KWS) specifically designed for devices with low computational and memory resources. MatchboxNet builds on the QuartzNet architecture [1]. It consists of a stack of blocks with residual connections [2]. Each block is composed from 1D time-channel separable convolutions (these are similar to 2D depth-wise separable convolutions [3, 4]), batch normalization, ReLU and dropout layers.

This paper makes the following contributions:

1. An end-to-end neural model for speech command recognition based on 1D time-channel separable convolutions
2. The model achieves state-of-the-art accuracy on Google Speech command datasets [5] but requires significantly fewer parameters than models which achieve similar accuracy.
3. The model scales well with the number of parameters.
4. A methodology to improve the model's robustness to background speech and noise.

2. Related Work

Neural network (NN)-based systems for Automatic Speech Recognition (ASR) have a long history, spearheaded by Time Delay Neural Networks (TDNN) for isolated word recognition [6, 7]. TDNN and Recurrent NNs (RNNs) were first used together with Hidden Markov Models (HMMs) in hybrid systems, where NN was used only for phonetic classification [8, 9, 10].

Rapid progress in deep learning for ASR [11, 12, 13] triggered research in end-to-end NN-based models for KWS. In 2015 Sainath and Parada proposed a convolutional NN for a small-footprint KWS [14]. Their model was composed of two convolutional layers, max-pooling in the temporal dimension, linear, and soft-max layers. Following the success of ResNets [2] in computer vision, Qian et al. [15] applied ResNets for ASR. Arik et al. [16] suggested Convolutional-RNN, which combined the strengths of convolutional layers and recurrent layers to exploit long-range context.

The introduction of the Google Speech Command dataset [5] in 2018 accelerated research in KWS and resulted in variety of new NN-based models, including deep residual networks ([17], [18]), special RNN with weight sharing [19], an RNN-Transducer with attention [20], and CNN with dilated convolutions and gating mechanisms [21].

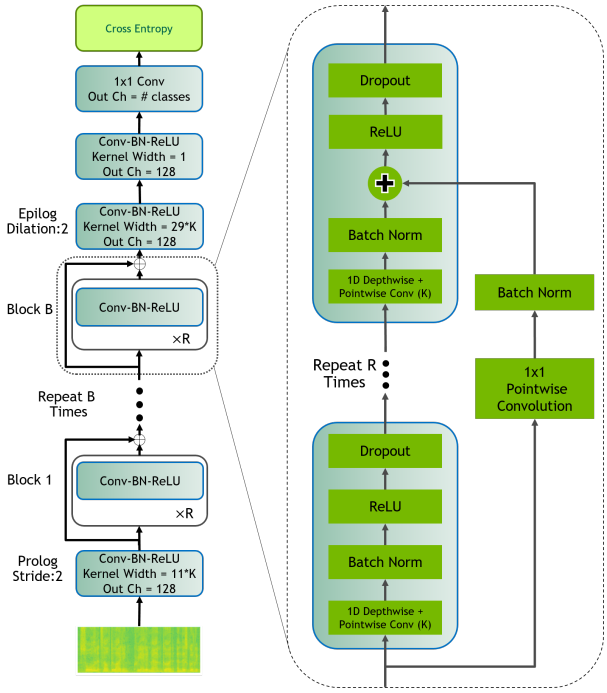

Figure 1: *MatchboxNet BxRxC model: B - number of blocks, R - number of sub-blocks, C - the number of channels.*

3. MatchboxNet Architecture

The MatchboxNet architecture is based on the QuartzNet end-to-end convolutional NN for ASR [1]. Similar to QuartzNet, MatchboxNet uses 1D time-channel separable convolutions to reduce model size versus regular 1D convolutions.

A MatchboxNet-BxRxC model has B residual blocks. Each block has R sub-blocks. All sub-blocks in a block have the same number of output channels C (see Fig. 1). A basic

http://dx.doi.org/10.21437/Interspeech.2020-1058

sub-block consists of a 1D-time-channel separable convolution, 1x1 pointwise convolutions, batch norm, ReLU, and dropout. The 1D-time-channel separable convolution has C filters with a kernel of the size k. All models have four additional sub-blocks: one prologue layer – 'Conv1' before the first block, and three epilogue sub-blocks ('Conv2', 'Conv3', and 'Conv4') before the final soft-max layer - see Figure 1) for details.

For example, the complete architecture for MatchboxNet-3x2x64 (B=3 blocks, R=2 sub-block per block, C=64 channels) is shown in the Table 1:

Table 1: *MatchboxNet-3x2x64 model has B=3 blocks, each black has R=2 time-channel separable convolutional sub-blocks with C=64 channels, plus 4 additional sub-blocks: prologue - Conv1, and epilogue - Conv2, Conv3, Conv4).*

Block	# Blocks	# Sub Blocks	# Output Channels	Kernel
Conv1	1	1	128	11
B1	1	2	64	13
B2	1	2	64	15
B3	1	2	64	17
Conv2	1	1	128	29, *dilation=2*
Conv3	1	1	128	1
Conv4	1	1	# classes	1
Soft-max Cross-entropy				

4. Experiments

We train MatchboxNet on the Google Speech Commands Dataset [5]. The dataset has two versions which we denote by v1 and v2. Version 1 has 65,000 utterances from various speakers, each utterance is 1 second long. Each of these utterances belongs to one of 30 classes corresponding to common words like "Yes", "No", "Go", "Stop", "Left", "Down", numerical digits, etc. Version 2 has 105,000 utterances, each 1 second long, belonging to one of 35 classes. We re-balanced both training datasets so all classes will have the same number of samples by duplication of random samples.[1]

4.1. Training Methodology

First, the input audio wave is converted into sequence of 64 mel-frequency cepstral coefficients (MFCC) calculated from 25ms windows with a 10ms overlap. We perform symmetric padding of the temporal dimension with zeros to fixed length of 128 feature vectors per sample.

Next, the input is augmented with time shift perturbations in the range of $T = [-5, 5]$ milliseconds and white noise with magnitude $[-90, -46]$ dB. In addition, we applied SpecAugment [22] with 2 continuous time mask of size $[0, 25]$ time steps, and 2 continuous frequency mask of size $[0, 15]$ frequency bands. We also used SpecCutout [23], with 5 rectangular masks with time and frequency dimensions similar to used in SpecAugment.

All models are trained with the NovoGrad optimizer [24], with $\beta_1 = 0.95$ and $\beta_2 = 0.5$. We utilize the Warmup-Hold-Decay learning rate schedule as in [25] with a warm-up ratio of 5%, a hold ratio of 45%, and a polynomial (2nd order) decay for the remaining 50% of the schedule. We use a maximum learning rate of 0.05 and a minimum learning rate of 0.001. We also incorporate weight decay of 0.001. We train all models for 200 epochs using mixed precision [26] on 2 V-100 GPUs with a batch size of 128 per GPU. All experiments were carried out using the NeMo toolkit [27] and plan to make all code necessary to reproduce these results available.

[1] One can use cross-entropy loss with class based weighing instead of re-balancing.

4.2. Results

Comparing with other published results, MatchboxNet-3x1x64 and MatchboxNet-3x2x64 obtain state-of-the-art (SOTA) accuracy on the Google Speech Commands dataset v1 and close to the SOTA on dataset v2, while requiring significantly fewer parameters than other models (see Table 2 and Table 3). For comparison we used the following models:

- DenseNet-BC: a variant of ResNets with dense connectivity in between layers of each block [28]. An intermediate point-wise convolution layer applied prior to the convolution block acts as a "bottleneck (B)" layer to reduce number of parameters. The number of channels in the convolutional layer can be reduced via a "compression (C)" factor.
- EdgeSpeechNet: ResNet-like deep residual ConvNet optimized for edge devices [29].
- Harmonic Tensor 2D-CNN: triangular band-pass filters of the *n*-th harmonic of center frequencies, are extracted and concatenated into a Harmonic Tensor of dimensionality $H \times F \times T$ (harmonic $\times$ frequency $\times$ time) which is then passed into a simple 2D-Convolutional NN [30].
- 'Embedding + Head': the acoustic embedding model with multiple heads is pre-trained to distinguish between various keyword groups on 200 million 2-second audio clips from YouTube. These heads are discarded after pre-training, and a single head is used to fine-tune the embedding model on the downstream task [31].

Table 2: *MatchboxNet on Google Speech Commands dataset v1, the accuracy is averaged over 5 trials (95% Confidence Interval).*

Model	# Parameters, K	Accuracy, %	Reference
ResNet-15	238	95.8 ± 0.351	[17]
DenseNet-BC-100	800	96.77	[32]
EdgeSpeechNet-A	107	96.80	[29]
MatchboxNet-3x1x64	77	97.21 ± 0.067	
MatchboxNet-3x2x64	93	97.48 ± 0.107	

Table 3: *MatchboxNet on Google Speech Commands dataset v2, the accuracy is averaged over 5 trials (95% Confidence Interval).*

Model	# Parameters, K	Accuracy, %	Reference
Attention RNN	202	94.30	[33]
Harmonic Tensor 2D-CNN	-	96.39	[30]
"Embedding + Head" Model	385	97.7	[31]
MatchboxNet-3x1x64	77	96.91 ± 0.101	
MatchboxNet-3x2x64	93	97.21 ± 0.072	
MatchboxNet-6x2x64	140	97.37 ± 0.110	

We also evaluate MatchboxNet-3 $\times$ 1 $\times$ 64 and MatchboxNet-3 $\times$ 2 $\times$ 64 models on the 12-class challenge from

the Speech Commands Dataset (v2). These models are trained on the following twelve labels - ten words "yes", "no", "up", "down", "left", "right", "on", "off", "stop", and "go" with additional two labels: "silence" and "unknown". The "unknown" category comprises of samples from the remaining 20 classes, where as "silence" is comprised of randomly sampled segments of background audio. We perform the same data split in the ratio 80:10:10 for the train, validation and test set, obtaining precisely 22246, 4445 and 4890 samples respectively. While we closely follow the original algorithm used to generate this dataset as proposed by Warden et al. [5], due to differences between deep learning frameworks, it is not feasible to generate the exact same "silence" and "background" dataset. Therefore, we publish our scores in Table 4 without comparison to other work.

Table 4: *MatchboxNet on Google Speech Commands dataset v2, the accuracy is averaged over 5 trials (95% Confidence Interval).*

Model	# Parameters, K	Accuracy, %	Reference
MatchboxNet-3x1x64	73	98.18 $\pm$ 0.081	
MatchboxNet-3x2x64	89	98.19 $\pm$ 0.097	

4.3. Model Scaling

We study the model scalability on the Google Speech Commands dataset v2 using MatchboxNet-3x2x64 as baseline. We scale model up using two methods: increase the depth $B \times R$ or increase the number of channels C. We found that both methods work in a similar way – the accuracy increases with model size until we hit $\approx 97.6\%$ (Table. 5).[2]

Table 5: *Scaling up MatchboxNet depth and number of channels, Speech Commands Dataset v2*

B	R	C	# Parameters, K	Accuracy,%
3	2	64	93	97.21
3	3	64	109	97.36
3	4	64	125	97.17
3	5	64	149	97.37
4	2	64	109	97.20
5	2	64	124	97.31
6	2	64	140	97.55
3	2	80	118	97.44
3	2	96	145	97.41
3	2	112	177	97.63

5. Model Robustness to Noise

To improve the robustness of MatchboxNet in the presence of noise, we retrained the model with background noise designed to interfere with speech signal. We construct a background noise dataset using audio samples from the *Freesound* database [34]. We partition each of these audio samples into segments of 1 second each, with no overlap between segments. Following this methodology, we obtain close to 55,000 noise samples.

[2]We analyzed the remaining misclassified samples, and found that most of them are very hard to recognize, even for humans.

5.1. Training with Noise Augmentation

We train MatchboxNet-3x1x64 by augmenting all training samples with randomly sampled noise segments. We scale the signal to noise ratio (SNR) randomly between 0 to 50 dB. In cases where the noise segment has a shorter duration than the training sample, we randomly augment a sub-segment of the training sample. The model accuracy on clean data is similar to the baseline model trained with basic augmentation only (Table. 6).

Table 6: *MatchboxNet-3x1x64 trained with additional background speech and noise augmentation, Google Speech Commands dataset v2. Accuracy (%) is averaged over 5 trials (95% confidence interval).*

Model	Augmentation	Accuracy, %
MatchboxNet 3x1x64	basic	96.91 $\pm$ 0.101
MatchboxNet 3x1x64	+ background speech and noise	97.05 $\pm$ 0.099

In order to evaluate the model robustness to environmental noise and background speech, we test the model with different noise conditions with SNR from -10 dB to +50 dB. We evaluate each test sample with 10 different randomly sampled noise segments, and compute the average accuracy over the entire test set. The model trained with additional noise augmentation is significantly more robust to external noise, even when the noise signal is much higher in amplitude than the noise used during training (Fig. 2).

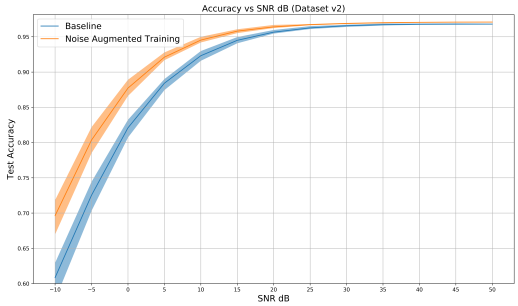

Figure 2: *MatchboxNet-3 $\times$ 1 $\times$ 64 trained with background noise augmentation, Speech Commands dataset v2. Accuracy vs SNR.*

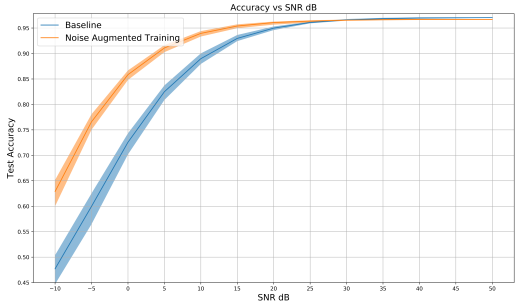

Figure 3: *MatchboxNet-3 $\times$ 1 $\times$ 64 trained with additional background speech and noise augmentation, expanded Google Speech Commands dataset v2. Accuracy vs SNR.*

5.2. Speech Commands Recognition with Background Speech and Noise Detection

To use a keyword spotting model in a continuous audio stream, it should be able to differentiate speech commands from the background speech or noise. For this, we added roughly 3500 samples for environmental noise and similar number of background speech samples from Freesound database to the training set. We re-trained a MatchboxNet-3x1x64 model to classify all original commands plus two additional classes - 'background noise' and 'background voice'. The model accuracy on the expanded speech commands datasets is shown in Table 7. Training with additional background speech and noise augmentation significantly improves the model robustness to noise (Fig. 3).

Table 7: *MatchboxNet-*$3 \times 1 \times 64$ *trained with additional background speech and noise augmentation, expanded Speech Commands dataset. Accuracy (%) is averaged over 5 trials (95% confidence interval).*

Model	Dataset	# Parameters	Accuracy, %
MatchboxNet-3x1x64	v1	77K	96.88 ± 0.073
MatchboxNet-3x1x64	v2	77K	96.97 ± 0.071

5.3. Robustness To Noise With Model Scaling

We further evaluate the relative robustness of larger MatchboxNet models to environmental noise and background speech. We train two models, MatchboxNet-3x1x64 and 6x2x64 with the exact same noise augmentation scheme as described above. We then evaluate the models on the unseen test set, perturbed by 10 random noise samples per test sample and compute the average accuracy. While both models are highly robust to external noise, MatchboxNet-6x2x64 consistently outperforms the smaller MatchboxNet-3x1x64 (see Table 8 and Figure 4)

Table 8: *MatchboxNet-*$3 \times 1 \times 64$ *and MatchboxNet-*$6 \times 2 \times 64$ *trained with additional background speech and noise augmentation. Accuracy (%) is averaged over 10 trials with random noise.*

Model	SNR (in dB)						
	-10	0	10	20	30	40	50
3x1x64	69.62	87.21	94.53	96.40	96.89	97.05	97.09
6x2x64	71.02	88.81	95.04	96.74	97.16	97.29	97.33

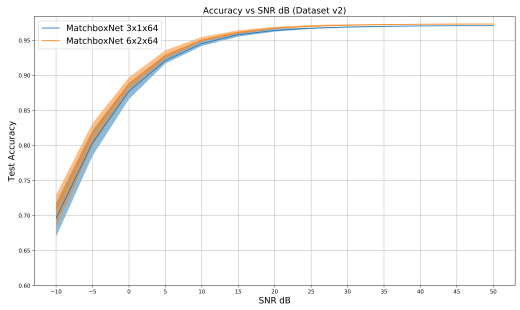

Figure 4: *MatchboxNet-*$3 \times 1 \times 64$ *and MatchboxNet-*$6 \times 2 \times 64$ *trained with additional background speech and noise augmentation. Accuracy vs SNR.*

6. Conclusions

In this paper, we present MatchboxNet, a new end-to-end deep neural network architecture for efficient recognition of speech commands on devices with limited computational and memory resources. MatchboxNet is a deep residual network composed from 1D time-channel separable convolution, batch-norm layers, ReLU and dropout layers. The model has state-of-the-art accuracy on the Google Speech Commands v1 dataset with significantly fewer parameters than models with similar accuracy. MatchboxNet is scalable, allowing it to be deployed on devices with different memory and compute capabilities. By using intensive data augmentation with auxiliary background noise during training, we have shown the model can be made very robust with respect to background noise.

7. Acknowledgments

We would like to thank NVIDIA AI Applications team for the help and valuable feedback.

8. References

[1] S. Kriman *et al.*, "QuartzNet: deep automatic speech recognition with 1D time-channel separable convolutions," *arXiv:1910.10261*, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.

[3] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1251–1258.

[4] L. Kaiser, A. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *arXiv:1706.03059*, 2017.

[5] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv:1804.03209*, 2018.

[6] A. Waibel, T. Hanazawa, G. Hinton, K. Shirano, and K. Lang, "A time-delay neural network architecture for isolated word recognition," *ICASSP*, 1989.

[7] K. Lang, A. Waibel, and G. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, 1990.

[8] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," *IEEE Transactions on Neural Networks, 3(2), 252–259*, 1992.

[9] T. Robinson, M. Hochberg, and S. Renals, "IPA: improved phone modelling with recurrent neural networks," *ICASSP*, 1994.

[10] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," *ICASSP*, 2000.

[11] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber, "Biologically plausible speech recognition with LSTM neural nets," in *Biologically Inspired Approaches to Advanced Information Technology. BioADIT*, 2004.

[12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks, vol. 18*, pp. 602—-610, 2005.

[13] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[14] T. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Interspeech*, 2015.

[15] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 481–488.

[16] S. O. Arik *et al.*, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv:1703.05390*, 2017.

[17] J. Tang, Y. Song, L. Dai, and I. McLoughlin, "Acoustic modeling with densely connected residual network for multichannel speech recognition," in *Interspeech*, 2018.

[18] S. Choi *et al.*, "Temporal convolution for real-time keyword spotting on mobile devices," *arXiv:1904.03814*, 2019.

[19] A. Kusupati *et al.*, "FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network," in *NIPS*, 2018.

[20] Y. He *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[21] A. Coucke *et al.*, "Efficient keyword spotting using dilated convolutions and gating," in *ICASSP*, 2019.

[22] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv:1904.08779*, 2019.

[23] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv:1708.04552*, 2017.

[24] B. Ginsburg *et al.*, "Stochastic gradient methods with layer-wise adaptive moments for training of deep networks," *arXiv:1905.11286*, 2019.

[25] T. He *et al.*, "Bag of tricks for image classification with convolutional neural networks," in *CVPR*, 2019, pp. 558–567.

[26] P. Micikevicius *et al.*, "Mixed precision training," *arXiv:1710.03740*, 2017.

[27] O. Kuchaiev *et al.*, "NeMo: a toolkit for building ai applications using neural modules," *arXiv:1909.09577*, 2019.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger, "Densely connected convolutional networks," *arXiv:1608.06993*, 2016.

[29] Z. Q. Lin, A. G. Chung, and A. Wong, "EdgeSpeechNets: Highly efficient deep neural networks for speech recognition on the edge," *arXiv:1810.08559*, 2018.

[30] M. Won, S. Chun, O. Nieto, and X. Serra, "Data-driven harmonic filters for audio representation learning," in *ICASSP*, 2020.

[31] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," *arXiv:2002.01322*, 2020.

[32] B. Li, F. Wu, S.-N. Lim, S. Belongie, and K. Q. Weinberger, "On feature normalization and data augmentation," *arXiv:2002.11102*, 2020.

[33] D. C. de Andrade, S. Leo, M. L. D. S. Viana, and C. Bernkopf, "A neural attention model for speech command recognition," *arXiv:1808.08929*, 2018.

[34] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 411–412.