# Bitwise Neural Networks for Efficient Single-Channel Source Separation

**Minje Kim***
Department of Intelligent Systems Engineering
Indiana University
Bloomington, IN 47401
minje@indiana.edu

**Paris Smaragdis**†
University of Illinois at Urbana-Champaign
Adobe Research
Urbana, IL 61801
paris@illinois.edu

## Abstract

We present Bitwise Neural Networks (BNN) as an efficient hardware-friendly solution to single-channel source separation tasks in resource-constrained environments. In the proposed BNN system, we replace all the real-valued operations during the feedforward process of a Deep Neural Network (DNN) with bitwise arithmetic (e.g. the XNOR operation between bipolar binaries in place of multiplications). Thanks to the fully bitwise run-time operations, the BNN system can serve as an alternative solution where efficient real-time processing is critical, for example real-time speech enhancement in embedded systems. Furthermore, we also propose a binarization scheme to convert the input signals into bit strings so that the BNN parameters learn the Boolean mapping between input binarized mixture signals and their target Ideal Binary Masks (IBM). Experiments on the single-channel speech denoising tasks show that the efficient BNN-based source separation system works well with an acceptable performance loss compared to a comprehensive real-valued network, while consuming a minimal amount of resources.

## 1 Introduction

Deep learning has become one of the major forces in machine learning-based source separation tasks, thanks to its powerful multi-layered structure that can learn complex mapping functions between a large amount of training samples (e.g. noisy speech) and their corresponding target values (e.g. clean speech) Xu et al. (2014); Huang et al. (2015); Wang & Wang (2013); Erdogan et al. (2015); Weninger et al. (2015); Le Roux et al. (2015). In many research areas the Deep Neural Network (DNN) topology is believed to capture a hierarchy of features Bengio (2009), which eventually provides a better performance in supervised learning tasks. Dictionary-based models by using Nonnegative Matrix Factorization (NMF) Lee & Seung (1999, 2001), on the other hand, learn a set of basis vectors that correspond to the weights of a shallow network. In general, we can say that the multiple hidden layers in the deep network structure can learn some more abstraction about the data at the cost of training and maintaining a larger amount of parameters, which can easily amount to a few millions.

This paper develops an efficient feedforward procedure that reduces the computational and spatial complexity of running and maintaining a DNN-based source separation system. The deep learning advances in source separation systems cost more resources, such as memory and power, due to the enlarged network structure. For example, now the network has to compute the feedforward operation for more hidden layers with larger weight matrices. Since those resources can be constrained in embedded devices, it could be prohibitive for them to perform multiplications between large matrices with millions of elements, although it is a fairly typical computation size in many DNNs.

*This work was partly supported by Intel Corporation.
†This work was supported by NSF grant 1453104.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

We employ Bitwise Neural Networks (BNN) Kim & Smaragdis (2015) to compress the network for an efficient implementation in a resource-constrained environment. BNN's drastically simplified feedforward operation comes from the fact that in a BNN all the input and output signals, weights and biases of the networks, and operations on them are all defined in an efficient bitwise fashion, which we will review more thoroughly in Section 2. If we represent the input signals with bipolar binary numbers, i.e. $+1$ and $-1$, and we train the bipolar binary parameters as well, the feedforward part can be done using only bitwise logics such as XNOR and bit counting, instead of multiplication, addition, and a nonlinear activation (e.g. tanh) on the usual floating or fixed-point variables. Note that since each weight and node will be encoded with binaries, the space requirement is also reduced compared to the multi-bit encoding schemes.

In all these procedures, we use binarized signals, so that the network can work in a fully bitwise fashion. For example, hidden unit output signals are already binarized thanks to the sign function as the activation, though binarization of input and target variables is an open question. In this paper, we propose a binarization technique, Quantization-and-Dispersion (QaD), which effectively encodes magnitude spectra. As for the target variable, Ideal Binary Masks (IBM) are a natural choice, although the same QaD process can potentially convert any real-valued target variables. Experimental results on some single-channel speech denoising tasks verify that the proposed BNN system gives comparable results to the similarly structured DNNs with near-continuous encoding strategy.

## 2 Bitwise Neural Networks (BNN)

### 2.1 Background: Neural Networks with Bitwise Feedforward

Although it has been known that any Boolean function can be represented as a bitwise network with one hidden layer, e.g. by memorizing all the relationships McCulloch & Pitts (1943), its training is an NP-complete problem Pitt & Valiant (1988). $\mu$-perceptron networks were proposed as a bitwise network, but its topology does not allow a full connection between units Golea et al. (1992). Soudry et. al. recently proposed the Expectation Back Propagation (EBP) algorithm to estimate the posterior probabilities for the bitwise parameters Soudry et al. (2014). It is a parameter-free learning algorithm and the discretization is convenient, yet it allows real-valued bias terms and relies on the averaged outputs of multiple networks whose parameters are sampled from the estimated distribution.

BNN can be seen as one of different ways to achieve the fully binary computation during the test time. More recently, there have been more neural networks that learn fully binary network parameters such as BNN Kim & Smaragdis (2015), BinaryConnect Courbariaux et al. (2015), and binarized neural networks Hubara et al. (2016). In the early stage of the neural network research, it has been known that if we decrease the quantization level of an already trained network parameters, the performance of the network drops significantly. One way to avoid this effect is to inject the quantization error during the training phase, by using the quantized version of the original continuous parameters during the feedforward so that the network is aware of the additional error introduced by the quantization and can fix it during backpropagation Fiesler et al. (1990); Hwang & Sung (2014). BNN adopts the quantization noise injection technique to estimate its bitwise parameters.

### 2.2 Feedforward in BNNs

#### 2.2.1 Notations and setup: bipolar binaries

Throughout the paper, we use bipolar binaries where the Boolean values are represented as $+1$ and $-1$. They are more expressive than 0-1 binaries in the sense that we can make use of zeros to explain the sparsity concept. With no loss of generality, in this paper we use the $\pm 1$ bipolar representation.

#### 2.2.2 The feedforward process

In a BNN the feedforward pass is defined as follows:

$$\bar{a}_i^{(l)} = \bar{b}_i^{(l)} + \sum_j^{K^{(l-1)}} \bar{w}_{i,j}^{(l)} \otimes \bar{z}_j^{(l-1)}, \quad \bar{z}_i^{(l)} = \text{sign}\big(\bar{a}_i^{(l)}\big), \tag{1}$$

$$\bar{\mathbf{z}}^{(l)} \in \mathbb{B}^{K^{(l)}}, \bar{\mathbf{W}}^{(l)} \in \mathbb{B}^{K^{(l)} \times K^{(l-1)}}, \bar{\mathbf{b}}^{(l)} \in \mathbb{B}^{K^{(l)}}, \bar{\mathbf{a}}^{(l)} \in \mathbb{Z}^{K^{(l)}},$$

where $\mathbb{B} = \{-1, +1\}$ and all lowercase letters are for scalar elements. $i$ and $j$ indicate $i$-th input and $j$-th output units of a layer, respectively. Bold characters are for vectors, and matrices are capitalized. The upper bar notation is to indicate that the parameters are integers or binaries. For convenience we drop the sample index. Sign function is used as an activation, which will produce bipolar binaries as the output. The sign function takes an integer $a_i^{(l)}$ as its input, whose value can be from $-K^{(l)}-1$ to $K^{(l)}+1$. $\otimes$ and the sign activation can be seen as a special operations designed for the desired speed-up, since they can replace the original real-valued multiplication and the smooth step functions, respectively.

### 2.3 Training BNNs

We can train a BNN by using two sequential runs of the Stochastic Gradient Descent (SGD) procedure. We first train an ordinary real-valued network, whose network structure is the same with the desired BNN. We use its weights to initialize the BNN parameters in the subsequent step. In the second phase of the training we finally learn the bitwise weights using a noisy feedforward pass. The detailed training procedure can be find in Kim & Smaragdis (2015).

## 3 Binarization of Signals

### 3.1 Quantization and Dispersion (QaD)

Since a BNN takes a bit pattern as the input, we need a binarization technique to encode any real-valued input signals. We found that Lloyd-Max's quantization Lloyd (1982) is useful, which could convert an input magnitude into a fixed-point value and feed it into an input unit. Instead of this integer input string, we treat each bit of the fixed-point quantized value as a binary feature. For example, after encoding the $f$-th magnitude coefficient of a noisy speech spectrum $\mathbf{x}_t$ into 4 bits, we disperse the 4 bits into 4 corresponding input units. Therefore, the BNN takes a $4F$-dimensional binary vector, where $F$ is the original number of coefficients in the spectrum.

### 3.2 Ideal Binary Masks (IBM)

As for the output units it is natural to form a softmax layer to solve classification problems as in Kim & Smaragdis (2015). On the other hand, the source separation problems often form continuous target variables such as the Ideal Ratio Masks (IRM) Narayanan & Wang (2013), for which we believe that the same QaD technique can be employed to convert them into bit patterns, too. In this paper, we conveniently make use of the IBM as our target, which is a natural choice for us to binarize the target Wang & Wang (2013). We leave the investigation of QaD for the continuous target variables to future work. Finally, the prediction error $\mathcal{E}$ can be measured as follows: $\mathcal{E}(n) = \frac{1}{2}\sum_i^{K^{(L+1)}} \left(t_i(n) - \bar{z}_i^{(L+1)}(n)\right)^2$, where $t_i(n)$ is an element of the bipolarized IBM mask.

## 4 Experiments

### 4.1 Experimental Setups

We prepare 121,280 training spectra (18,020 of them are used for validation). Twelve gender-balanced TIMIT speakers are chosen for training, each of which contributes five chosen utterances, totalling 60 clean speech signals. They are then mixed with ten different non-stationary noise signals proposed in Duan et al. (2012) with 0 dB Signal-to-Noise Ratio (SNR) to form 600 noisy utterances. Among them, we set aside 100 utterances as a validation set. By applying Short-Time Fourier Transform (STFT) with a Hann window of 1024 points and a 75% of overlap. For testing, another four speakers are chosen and mixed with the same set of noise signals, but from different parts to make sure the test mixtures are not seen during training. We apply a 4-bit QaD procedure to these spectra as an input to the BNN systems. We found $\rho = 0.95$ optimal via validation. All signals are with sampling rate 16 kHz. We train three different kinds of neural networks that predict the IBM of the given magnitude spectrum:

• *Baseline*: The baseline networks take the ordinary 513 dimensional real-valued magnitude spectrum as its input. For training this network the first round training algorithm is used, where the additional

| Systems | Topology | SDR | SIR | SAR | STOI |
|---|---|---|---|---|---|
| Baseline with original input | 1024×2 | 10.17 | 26.69 | 10.45 | 0.7880 |
| | 2048×2 | 10.57 | 26.25 | 10.88 | 0.8060 |
| Baseline with binary input | 1024×2 | 9.80 | 27.00 | 10.08 | 0.7790 |
| | 2048×2 | 10.11 | 26.61 | 10.43 | 0.7946 |
| BNN | 1024×2 | 9.35 | 23.38 | 9.82 | 0.7819 |
| | 2048×2 | 9.82 | 23.62 | 10.40 | 0.7961 |

Table 1: Speech denoising performance of the proposed BNN-based separation system compared with the other real-valued networks.

weight compression works like max-norm feature in Srivastava et al. (2014). It employs dropout with 0.95 for the first layer (dropping 5%) and 0.8 for the other layers as the parameter.

• *Baseline with binary input*: This setup is equivalent to the first round of the BNN training. The difference between this and the baseline is that the first round networks take the $4 \times 513$ 4-bit QaD vectors as their input. The prediction results from these first round networks will serve as an upper bound of the separation performance of a real-valued network with the binarized input. Also, the learned parameters will be reused to intialize the second round parameters.

• *The proposed BNN*: Here we combine the first round results (baseline with binary input) and the second round.

Validation determines the learning rate which usually starts from $10^{-7}$ or $10^{-6}$ and gradually decreases. Minibatch and the momentum parameter are set to be 100 frames and 0.95, respectively.

### 4.2 Discussion

Table 1 lists the speech denoising performance of the systems in terms of Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), Signal-to-Artifact Ratio (SAR) Vincent et al. (2006), and Short-Time Objective Intelligibility (STOI) Taal et al. (2010). First, we can see that doubling up the number of hidden units generally improves SAR, which eventually improves SDR as well. STOI gets better with more hidden units, too. For example, BNN with 2048 hidden units catches up the performance of $1024 \times 2$ DNN with binary input (9.82 versus 9.80 dB in SDR and 0.7861 versus 0.7790 in STOI). If we compare the effect of the QaD binarization (baseline with original versus binary input), we see a slight performance drop in both SDR (0.37 dB and 0.47 dB for the 1024 and 2048 hidden units, respectively) and STOI (0.009 and 0.0114). Starting from there, BNN catches up with the performance of the baseline system with binarized input by a margin 0.45 dB and 0.29 dB for the 1024 and 2048 units, respectively. The 2048×2 BNN lost 0.0045, but 1024×2 happens to improve STOI by 0.0029.

Overall, we see that the proposed BNN that is fully binarized from its input to the output shows reasonable performance compared to its corresponding real-valued networks. We believe that our experiments on the Fourier spectra and IBM prove the concept well enough and are ready to be extended to the other types of features, target variables, and the choice of context window, since the QaD technique and the BNN training methods work for general purposes.

## 5 Conclusion

We proposed a novel efficient source separation system by employing BNNs, which redefined the feedforward pass in a bitwise fashion. A two-stage training strategy was introduced to prepare a set of compressed weights, and then to initialize the BNN parameters that are eventually binarized during the feedforward pass. By binarizing the input magnitude spectra with the QaD technique and having IBM as the target, we showed that BNN performs well for the speech denoising job with a minimal computational cost.

## References

Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1–127, 2009.

Courbariaux, M., Bengio, Y., and David, J.-P. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3105–3113, 2015.

Duan, Z., Mysore, G. J., and Smaragdis, P. Online PLCA for real-time semi-supervised source separation. In *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 34–41, 2012.

Erdogan, H., Hershey, J. R., Watanabe, S., and Le Roux, J. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

Fiesler, E., Choudry, A., and Caulfield, H. J. Weight discretization paradigm for optical neural networks. In *The Hague'90, 12-16 April*, pp. 164–173. International Society for Optics and Photonics, 1990.

Golea, M., Marchand, M., and Hancock, T. R. On learning $\mu$-perceptron networks with binary weights. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 591–598, 1992.

Huang, P., Kim, M., Hasegawa-Johnson, M., and Smaragdis, P. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2136–2147, Dec 2015.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized neural networks. In *Advances in Neural Information Processing Systems*, pp. 4107–4115, 2016.

Hwang, K. and Sung, W. Fixed-point feedforward deep neural network design using weights +1, 0, and -1. In *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, Oct 2014.

Kim, M. and Smaragdis, P. Bitwise neural networks. In *International Conference on Machine Learning (ICML) Workshop on Resource-Efficient Machine Learning*, Jul 2015.

Le Roux, Jonathan, Hershey, John R., and Weninger, Felix. Deep NMF for speech separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13. MIT Press, 2001.

Lloyd, S.P. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2), Mar 1982.

McCulloch, W. S. and Pitts, W. H. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

Narayanan, A. and Wang, D. L. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7092–7096, May 2013.

Pitt, L. and Valiant, L. G. Computational limitations on learning from examples. *Journal of the Association for Computing Machinery*, 35:965–984, 1988.

Soudry, D., Hubara, I., and Meir, R. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, January 2014.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

Vincent, E., Fevotte, C., and Gribonval, R. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

Wang, Y. and Wang, D. L. Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1381–1390, July 2013.

Weninger, Felix, Erdogan, Hakan, Watanabe, Shinji, Vincent, Emmanuel, Le Roux, Jonathan, Hershey, John R., and Schuller, Björn. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, August 2015.

Xu, Y., Du, J., Dai, L.-R., and Lee, C.-H. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal Processing Letters*, 21(1):65–68, 2014.