NEURAL SOURCE CODING FOR BANDWIDTH-EFFICIENT BRAIN-COMPUTER INTERFACING WITH WIRELESS NEURO-SENSOR NETWORKS

Thomas Strypsteen and Alexander Bertrand

KU Leuven, Department of Electrical Engineering (ESAT),
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Leuven, Belgium

ABSTRACT

Neural Source Coding (NSC) is a technique that exploits the modelling power of (deep) neural network for the purpose of source coding. Its goal is to transform the data into a space of low entropy, where they can be coded by classic entropy coding schemes. In this paper, our goal is to investigate the use of NSC in so-called neuro-sensor networks, i.e., a type of body-sensor network consisting of a collection of wireless sensor nodes that record brain activity at different scalp locations, e.g., via electroencephalography (EEG) sensors. All nodes wirelessly transmit their data to a fusion center, where inference is then performed on the joint sensor signals by a given deep neural network. The NSC parameters and inference network are learned jointly, optimizing the trade-off between accuracy and bitrate for a given application. We validate this method on a motor execution task in an emulated EEG sensor network and compare the resulting trade-offs with those obtained by directly quantizing the transmitted data to low-bit precision. We demonstrate that NSC yields more favorable trade-offs than straightforward quantization for very low bit depths and allows for large bandwidth gains at little loss in accuracy on the investigated brain-computer interface (BCI) task.

***Index Terms*—** Quantized Deep neural networks, Distributed deep neural networks, Wireless EEG sensor networks

1. INTRODUCTION

In the last few years, technological advances such as miniaturization of microprocessors and energy-efficient batteries have increasingly enabled the usage of wearable, physiological sensors for ambulant health monitoring. Many applications however, will require recording of different data modalities or multiple channels of the same data type at different locations to extract meaningful patterns. This naturally leads to the concept of a body-sensor network (BSN), where the different sensors wirelessly share their data and solve a given task in a distributed setting. One example of such a BSN is a so-called neuro-sensor network to record brain activity, e.g., based on electroencephalography (EEG) sensors, in which case it is referred to as a Wireless EEG Sensor Network (WESN) [1, 2]. EEG is a widely used, noninvasive way to measure the electrical activity of the brain. These signals can be harnessed for various purposes, including, e.g., sleep stage analysis [3], epileptic seizure detection [4] and brain-computer interfaces (BCI), allowing for direct communication between the human brain and external machines. In a WESN, the EEG is not recorded with a bulky cap as would traditionally be the case, but by a number of lightweight mini-EEG devices, each one capable of recording one or a few EEG channels from its local scalp area, performing some pre-processing and transmitting this data to the other nodes or a fusion center. A major constraint in the design of these networks is that they should be energy-efficient, enabling a maximal battery lifetime. In BSNs, the typical energy bottleneck will be the wireless transmission of the data between the sensors and/or a fusion center [1], motivating the search for solutions that reduce the amount of data to be transmitted without affecting the ability of the BSN to achieve sufficient performance on the given inference task.

In this paper, we investigate the use of Neural Source Coding (NSC) [5,6] as a way to perform efficient, *learnable* quantization and coding to decrease the bandwidth requirements for nodes transmitting their data to be analyzed by a deep neural network (DNN) and investigate whether it is able to outperform straightforward mixed-precision training. In contrast to previous work in NSC, we will employ NSC to perform task-specific compression and minimize the bitrate of multiple nodes simultaneously, while controlling for a balanced bit allocation across the different nodes. While our evaluation use case is focused on brain signals, the methodology is presented in a generic setting without any assumption on the content of the data. In Section 2, we formally present our problem statement and discuss mixed-precision training, i.e., directly quantizing the data at low-bit-precision, as a benchmark method. We then proceed by presenting the NSC framework and how it can be applied to our use case in Section 3. In Section 4, we evaluate and compare both approaches on real EEG data for solving a motor execution task. Conclusions and future extensions are discussed in Section 5.

2. PROBLEM STATEMENT AND BENCHMARK

Consider a sensor network with node set $\mathcal{N}$ containing N sensor nodes, where each node $n \in \mathcal{N}$ measures a multi-channel EEG signal at its respective scalp location and transmits windows of this data $\mathbf{x}_n$ to a fusion center. At the fusion center, the different data $\{\mathbf{x}_n | n \in \mathcal{N}\}$ are merged and analyzed by a multi-channel model, in our case a DNN. However, simply offloading the measured, raw data $\mathbf{x}_n$ would consume a large amount of transmission energy, which cannot be reasonably supplied by the small batteries on these

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 802895). The authors also acknowledge the financial support of the FWO (Research Foundation Flanders) for project G.0A49.18N, and the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

T. Strypsteen and A. Bertrand are with KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics and with Leuven.AI - KU Leuven institute for AI, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (e-mail: thomas.strypsteen@kuleuven.be, alexander.bertrand@kuleuven.be).

miniaturized devices. Thus, the question arises how we can transmit $\mathbf{x}_n$ as efficiently as possible by compressing it locally at node n and performing reconstruction at the fusion center. By allowing for some distortion on the data we transmit (i.e., performing lossy compression), we could trade a small amount of task accuracy to achieve large bandwidth gains. By training the compression at all the nodes $n \in \mathcal{N}$ and the inference DNN at the fusion center jointly in an end-to-end fashion, it is possible to perform task-dependent compression, allowing for the discarding of signal components that might be useful for a precise reconstruction, but offer little information for the specific task at hand. Furthermore, the model can learn to deal with the added distortion to achieve more favorable trade-offs than it would be able to otherwise. A final aspect to take into account here is that the battery lifetime of the system will ultimately depend on the node with the highest required bitrate: the critical node. We will thus not only have to minimize the average bitrate of the nodes, but also ensure a balanced load across the nodes. In this paper, we will focus on compressing the signal by performing downsampling and coding the samples to a lower amount of bits than the standard 32-bit precision. While this work will focus on the specific use case of a WESN, it is important to note that the methods that we will discuss can be employed in any body-sensor network, or more in general, any setting where the input data for a neural network is distributed across sensor nodes that can only communicate on bandwidth-limited channels.

A straightforward way to reduce the consumed bandwidth of the nodes is to forego the traditional full 32-bit precision used for floating point numbers, but instead represent the transmitted data with a smaller number of bits. In neural networks in particular, moving from 32 to 8-bit precision can typically yield great improvements in memory footprint and inference efficiency while barely affecting task accuracy at all on a wide range of computer vision models [7]. More recently, models reaching even lower bit depths of 2-4 bits have been proposed with only marginal losses in performance. [8–10]. Typically, these models quantize both the network weights and the activation values and employ the same number of bits in every layer of the network. In contrast, since the transmission energy will form the bottleneck of the WESN design, we are only concerned with quantizing the data to be transmitted by the nodes of the WESN, so we will only target a lower bit depth at this point in the network, i.e., we quantize $\mathbf{x}_n$ into a quantized vector $\bar{\mathbf{x}}_n$. Once the data $\bar{\mathbf{x}}_n$ is received at the fusion center, we compute an approximation $\hat{\mathbf{x}}_n$ of the original data that is further processed in standard 32-bit precision. As a benchmark, we use a state-of-the-art learnable quantization scheme, namely the Learnable Step Quantization (LSQ+) approach of [10] for its simplicity and capability of reaching high accuracies with only a low number of bits. To quantize a floating point number x to an integer $\bar{x}$, we pass it through a quantizer Q at the nodes and a dequantizer Q^{-1} at the fusion center:

$$\begin{aligned} Q(x) &= \bar{x} = \lfloor clip(\frac{x-\beta}{s}, -2^{b-1}, 2^{b-1}-1) \rceil \\ Q^{-1}(\bar{x}) &= \hat{x} = s\bar{x} + \beta \end{aligned} \tag{1}$$

with b the number of bits, s a learnable scale parameter and β a learnable offset. This scale allows us to find an optimal quantization step size for the given task that creates a trade-off between the dynamic range and the quantization error. To enable this quantizer to be learned through backpropagation, we employ the straight-through estimator (STE) [11], ignoring the non-differentiable rounding operation in the backward pass and treating it as a simple identity operation. Previous research in quantizing neural networks has provided ample evidence that Quantization-Aware Training (QAT), i.e., re-training the network with knowledge of the quantizer and the resulting quantization noise, yields better bit-accuracy trade-offs than Post-Training Quantization (PTQ), i.e., simply quantizing the weights and activations of a learned full-precision model post-hoc, especially in low-precision settings [12, 13]. Even more advantageous is not performing QAT from scratch, but initializing the model by first performing full-precision training and then fine-tuning the quantized model [14, 15], an approach we will also follow here.

3. NEURAL SOURCE CODING

In order to further reduce the bandwidth, we propose the use of Neural Source Coding (NSC), a framework becoming more and more popular in image compression [6]. Here, we will adopt a similar approach for compressing sensor network signals (in particular for EEG data), and compare it with a standard mixed-precision approach. The main differences between our approach and previous work in NSC are that (1) we perform task-specific compression instead of reconstruction of the original data, (2) we investigate how well the nodes can compress their local data $\mathbf{x}_n$ without having access to the data $\mathbf{x}_j$ of the other nodes $j \neq n$ and (3) we will add some additional regularization to ensure that the transmission load is balanced across the network nodes. The main idea behind NSC is to transform the data into discrete symbols, which are then encoded by entropy coding schemes such as Huffman coding or Asymmetric Numeral Systems [16], which yield an expected code length equal to the entropy of the discrete source. By modeling the distributions of these discrete symbols, estimating their entropy and adding a regularization term to the loss function that penalizes high entropies, the network is encouraged to transform the data in a discrete space of low entropy that can be efficiently coded, while reducing the impact on the task performance as much as possible.

To implement this scheme in our network, we follow the approach of Ballé et al. [6]. The data window $\mathbf{x}_n \sim p_n(\mathbf{x}_n)$ that each node needs to transmit is first fed to an encoding transformation $g_{\boldsymbol{\alpha}_n}(\mathbf{x}_n)$, allowing us to alter the distribution of $\mathbf{x}_n$. In our case, this transformation consists of a convolutional layer and then multiplication with a scaling vector, that can widen or narrow the distribution of the generated elements to increase or lower their entropy. The transformed data $\mathbf{z}_n$ is then discretized to $\bar{\mathbf{z}}_n$, a process that is simulated during training by adding uniform noise in the interval $[-0.5, +0.5]$, which mirrors the effect of quantization noise. During validation, the data is instead rounded to the nearest integer. The encoding transformation and the discretization of $\mathbf{x}_n$ result in a new distribution $p^\star_n(\bar{\mathbf{z}}_n)$. For each element $\bar{z}_{ni}$ in the vector $\bar{\mathbf{z}}_n$, the univariate entropy model $q_{\boldsymbol{\phi}_{ni}}(\bar{z}_{ni})$, a neural network architecture described in [6], is learned to estimate its likelihood. The likelihood of the full vector $q_{\boldsymbol{\phi}_n}(\bar{\mathbf{z}}_n)$ is then estimated as the product of these independent, univariate likelihoods. In order to both fit the entropy model $q_{\boldsymbol{\phi}_n}(\bar{\mathbf{z}}_n)$ to the distribution of the transformed data $p^\star_n(\bar{\mathbf{z}}_n)$ and penalize the entropy of this distribution, we define the loss term $\mathcal{L}_{R,n}$ for each node n:

$$\begin{aligned} \mathcal{L}_{R,n}(\mathbf{x}_n, \boldsymbol{\phi}_n, \boldsymbol{\alpha}_n) &= \mathbb{E}_{\bar{\mathbf{z}}_n \sim p^\star_n(\bar{\mathbf{z}}_n)}[-\log_2(q_{\boldsymbol{\phi}_n}(\bar{\mathbf{z}}_n))] \\ &= \mathbb{E}_{\mathbf{x}_n \sim p_n(\mathbf{x}_n)}[-\log_2(q_{\boldsymbol{\phi}_n}(Q(g_{\boldsymbol{\alpha}_n}(\mathbf{x}_n))))] \end{aligned} \tag{2}$$

with $p_n(\mathbf{x_n})$ the distribution of $\mathbf{x}_n$, the input data of node n, $g_{\boldsymbol{\alpha}_n}$ the encoding transform of node n (a neural network parameterized by $\boldsymbol{\alpha}_n$), $q_{\boldsymbol{\phi}_n}$ the entropy model estimating the likelihood of the data

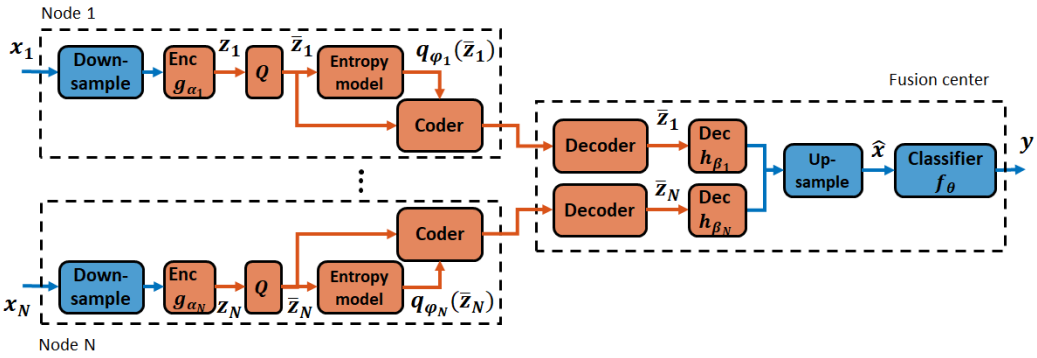


Fig. 1: Schematic overview of the proposed NSC-based network. Blue blocks indicate the distributed baseline with down-and upsampling (both used LSQ+ and NSC), orange blocks indicate modules added in the NSC framework.

(a neural network parameterized by ϕ_n and whose architecture is described by [6]) and Q the quantizer adding noise during training and performing rounding at validation. This loss minimizes the KL-divergence between $p_n^\star$ and q_n, the true and estimated distribution of $\bar{\mathbf{z}}_n$ and at the same time minimizes the entropy of the estimated distribution, encouraging the encoding transformation to transform the original data into a new space of lower entropy and consequently, lower code lengths. The discrete data $\bar{\mathbf{z}}_n$ is then losslessly source coded with the distribution $q_{\boldsymbol{\phi}_n}(\bar{\mathbf{z}}_n)$ and transmitted to the fusion center. Here, a decoding transform $h_{\boldsymbol{\beta}_n}$ transforms the data back into the original space to produce an approximation $\hat{\mathbf{x}}_n$, in this case employing a transposed convolution and scaling to mirror the encoding transform. This reconstruction is then fed to the original classifier network. We then propose the following rate-distortion loss:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \; & \mathcal{L}_{CE}\left(f_{\boldsymbol{\theta}}\left(h_{\boldsymbol{\beta}}\left(Q\left(g_{\boldsymbol{\alpha}}\left(\mathbf{x}\right)\right)\right)\right)\right) \\ & + \lambda\Bigg((1-\mu)\sum_n \mathcal{L}_{R,n}\left(\mathbf{x}_n, \boldsymbol{\phi}_n, \boldsymbol{\alpha}_n\right) \\ & + \mu \max_n \mathcal{L}_{R,n}\left(\mathbf{x}_n, \boldsymbol{\phi}_n, \boldsymbol{\alpha}_n\right)\Bigg)\end{aligned} \tag{3}$$

with $\mathcal{L}_{CE}$ the cross-entropy loss between the classification of the input data $\mathbf{x} = [\mathbf{x}_1^T, ...\mathbf{x}_N^T]^T$ and the target label y, $h_{\boldsymbol{\beta}_n}$ the decoding transform for node n parameterized by $\boldsymbol{\beta}_n$, $f_{\boldsymbol{\theta}}$ the original classification network with parameters $\boldsymbol{\theta}$ and λ a parameter to be tuned to perform the trade-off between the rate (the entropy of the transmitted data $\bar{\mathbf{z}}_n$) and the distortion (i.e., the classification accuracy of the BCI task). Important to note is that in our setting, we are not only interested in the *average* bitrate $\mathcal{R}_{avg}$ across the nodes, but the battery life of the system will rather be dependent on the critical node, i.e., the node with the *maximal* bitrate $\mathcal{R}_{max}$. Thus, a third term is included in the loss, which penalizes the rate of the currently most demanding node. This will prevent the rate of a single node to dominate and enforce a more balanced transmission load across the node. This loss is balanced with the total bitrate with a hyperparameter μ, a value between 0 and 1 which we will set at 0.5. A schematic overview of the NSC scheme is presented in Figure 1.

4. EXPERIMENTAL RESULTS

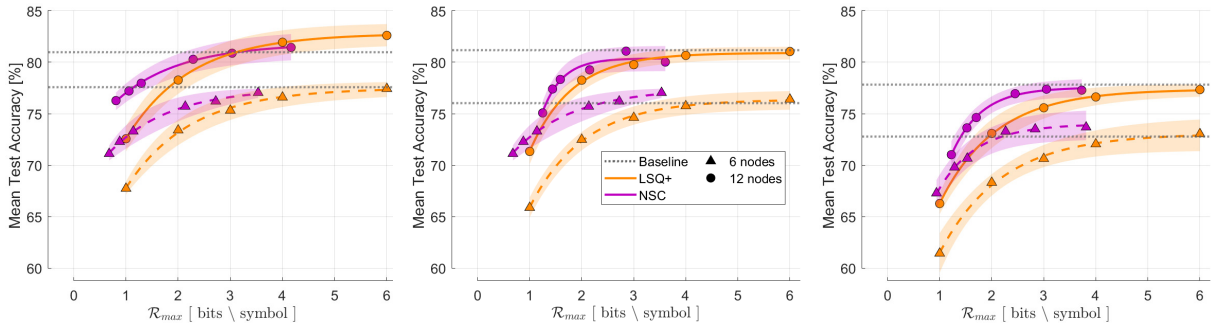
4.1. Experimental setup

We evaluate the capability of the proposed NSC-based pipeline to reduce the bandwidth of the nodes in a simulated WESN solving a motor execution task and compare it to LSQ+. In this setting, each of the nodes is able to record a single EEG channel, perform some local processing and transmit the data to a fusion center where data of all the nodes is aggregated and classified. The goal of the motor execution task is to decode from EEG signals which specific body movement is executed by the subject. For our experiments, we make use of the publicly available High Gamma Dataset[1] [17], containing 128-channel EEG recordings from 14 subjects, with about 880 trials of executed movement per subject following a visual cue. These movements belong to one of four classes: left hand, right hand, feet and rest. We employ the standard preprocessing procedure from [17]. The neural network architecture we employ for classification in the fusion center is the multiscale parallel filter bank convolutional neural network (MSFBCNN) proposed in [18].

In the WESN setting, we are only capable of using a small number of mini-EEG devices. Thus, for each run, we first perform an EEG channel selection step, using the regularized Gumbel-softmax method described in [19]. To do so, we first train the the centralized MSFBCNN after extending it with a selection layer, which jointly learns the optimal set of N channels for the given task and architecture, together with the network weights. Each of these channels is then assigned to a node and we begin training the distributed network for the WESN. In our experiments, we will employ both a value of $N = 6$ nodes, a realistic value for a WESN and a value of $N = 12$ nodes to evaluate the consistency of the methods in larger sensor networks.

At each of the sensor nodes, we will first perform downsampling of the recorded data with two strided learnable convolutional layers, already decreasing the bandwidth requirements before quantization, at the cost of some information loss, as in [2]. At the fusion center, the data is upsampled with two transposed, strided convolutional layers, mirroring the strides of the downsampling layers, to restore the data to its original dimension before feeding it to the original, centralized MSFBCNN classifier. In the LSQ+ case, the model is extended with a quantizer and dequantizer at the interface between the nodes and the fusion center, with a learnable quantization step size and offset as described in Equation (1). In the NSC case, we add the encoding and decoding transforms $g_{\boldsymbol{\alpha}_n}$ and $h_{\boldsymbol{\beta}_n}$, discretization Q, the entropy models $q_{\boldsymbol{\phi}_n}$ and the rate-distortion loss (see Figure 1). For both approaches, we will first initialize the weights of the down-and upsampling layer and the MSFBCNN classifier by pre-

[1] https://github.com/robintibor/high-gamma-dataset



(a) Downsampling factor 1 (b) Downsampling factor 4 (c) Downsampling factor 9

Fig. 2: Bandwidth-accuracy trade-off for the LSQ+ and NSC schemes with different downsampling factors and different number of nodes. Mean test accuracies are plotted against average number of bits per element of the transmitted windows for the critical node of the network, which requires the highest transmission rate. 10 runs are performed for different values of the trade-off parameter λ for NSC and different bit lengths for LSQ+. Shaded regions indicate one standard deviation of the individual runs. Dotted lines indicate accuracies of the 6- and 12-node network without any quantization. NSC allows for more favorable Pareto fronts than LSQ+, especially in very low-precision regions.

training the network without quantization on the data of all subjects jointly. In the second step of the training, the modules responsible for the quantization are enabled, while the parts of the network that have already been trained in the first step will be fine-tuned with a learning rate that is 10 times lower.

4.2. Model performance

The bandwidth-accuracy trade-offs obtained by LSQ+ and NSC for a network of 6 and 12 nodes with different downsampling factors are presented in Figure 2. Important to note is that in these trade-offs, we are mainly interested in the bitrate of the critical node $\mathcal{R}_{max}$, not the average bitrate across the nodes R_{avg}. A first thing to note is that both schemes already allow for a large reduction of bandwidth with only small losses in accuracy compared to the full-precision baseline, easily handling 4-bit precision. When enough bits are employed, the quantized networks can even slightly exceed the baseline in some cases, due to the quantization's regularizing effect. NSC however, generally outperforms LSQ+ in the very low-bandwidth regions of 1-3 bit. A second advantage of NSC is that it also allows us to access a continuous range of bit depths, since each window is encoded with a variable amount of bits and the *average* code length is dependent on the entropy of the generated symbols. LSQ+ in contrast, by default allocates a fixed, discrete number of bits to each element of the transmitted vector. Finally, we investigate how balanced the transmission load across the network is and verify whether the achieved bitrates are not caused by a single dominating node. To do this, we look at the relative difference between $\mathcal{R}_{max}$ and $\mathcal{R}_{avg}$ for the 6-node network with a downsampling factor of 9 as an example, illustrated in Figure 3. Here, we see that when μ is 0, i.e., the bitrate of the critical node is not penalized in Equation (3), the bitrate of the critical node is on average about 20% higher than the node average. Enabling the regularization by setting μ to 0.5 reduces this gap about 5%, thus enforcing a more equally balanced load across the nodes.

5. CONCLUSION AND FUTURE OUTLOOK

We have studied how to efficiently code the data to be transmitted in a WESN solving a motor execution task and investigated whether the performance of a state-of-the-art mixed-precision approach based on

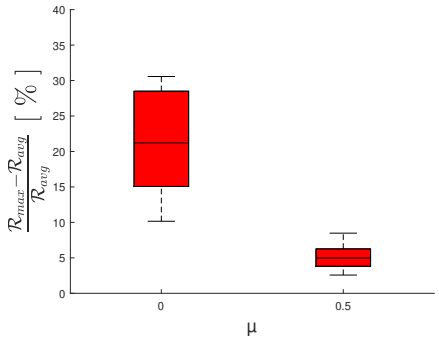


Fig. 3: Relative difference between the bitrate of the critical node and average bitrate, i.e. $\frac{\mathcal{R}_{max}-\mathcal{R}_{avg}}{\mathcal{R}_{avg}}$ for the 6-node NSC network with a downsampling factor of 9. Setting μ to 0.5 to enable penalization of the bitrate of the critical node during training decreases this difference and enforces a more balanced load across the nodes.

LSQ+ could be exceeded by the usage of NSC. We have compared the capability of both approaches to reduce the required bandwidth while maintaining as much task accuracy as possible in this problem. We have also made sure the transmission load across the nodes is balanced (i.e., there is no single node with a dominating contribution to the average bitrate) by introducing an extra regularization term in the rate-distortion loss. We have demonstrated that while both approaches can easily reduce precision to at least 4 bits with little decrease in accuracy, the NSC yields more favorable bandwidth-accuracy trade-offs than LSQ+ for regions of low bitrates. In future work, we will extend the NSC framework, e.g., by replacing the encoding and decoding transform with specific autoencoders for EEG, which could allow us to transform the time series into a latent space with higher sparsity. Another possible extension lies in the entropy model, which currently factorizes the distribution of the full window into independent distributions for each element, but could be extended with conditional dependencies which can model the correlations between subsequent elements in the time series. Future work will involve studying the impact of these extensions on WESNs and the generalization to other kinds of body-sensor networks.

6. REFERENCES

[1] Alexander Bertrand, "Distributed signal processing for wireless EEG sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, 2015.

[2] Thomas Strypsteen and Alexander Bertrand, "Bandwidth-efficient distributed neural network architectures with application to body sensor networks," *arXiv preprint arXiv:2210.07750*, 2022.

[3] Ofelie De Wel, Mario Lavanga, Alexander Caicedo Dorado, Katrien Jansen, Anneleen Dereymaeker, Gunnar Naulaers, and Sabine Van Huffel, "Complexity analysis of neonatal EEG using multiscale entropy: applications in brain maturation and sleep stage classification," *Entropy*, vol. 19, no. 10, pp. 516, 2017.

[4] Amir H Ansari, Perumpillichira J Cherian, Alexander Caicedo, Gunnar Naulaers, Maarten De Vos, and Sabine Van Huffel, "Neonatal seizure detection using deep convolutional neural networks," *International journal of neural systems*, vol. 29, no. 04, pp. 1850011, 2019.

[5] Jay Whang, Anish Acharya, Hyeji Kim, and Alexandros G Dimakis, "Neural distributed source coding," *arXiv preprint arXiv:2106.02797*, 2021.

[6] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[7] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry, "Scalable methods for 8-bit training of neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[8] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha, "Learned step size quantization," *arXiv preprint arXiv:1902.08153*, 2019.

[9] Jungwook Choi, Swagath Venkataramani, Vijayalakshmi Viji Srinivasan, Kailash Gopalakrishnan, Zhuo Wang, and Pierce Chuang, "Accurate and efficient 2-bit quantized neural networks," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 348–359, 2019.

[10] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak, "Lsq+: Improving low-bit quantization through learnable offsets and better initialization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 696–697.

[11] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[12] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021.

[13] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, 2021.

[14] Jeffrey L McKinstry, Steven K Esser, Rathinakumar Appuswamy, Deepika Bablani, John V Arthur, Izzet B Yildiz, and Dharmendra S Modha, "Discovering low-precision networks close to full-precision networks for efficient embedded inference," *arXiv preprint arXiv:1809.04191*, 2018.

[15] Wonyong Sung, Sungho Shin, and Kyuyeon Hwang, "Resiliency of deep neural networks under quantization," *arXiv preprint arXiv:1511.06488*, 2015.

[16] Jarek Duda, "Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding," *arXiv preprint arXiv:1311.2540*, 2013.

[17] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[18] Hao Wu, Fu Li, Yuchen Li, Boxun Fu, Guangming Shi, Minghao Dong, and Yi Niu, "A parallel multiscale filter bank convolutional neural networks for motor imagery EEG classification," *Frontiers in Neuroscience*, vol. 13, pp. 1275, 2019.

[19] Thomas Strypsteen and Alexander Bertrand, "End-to-end learnable EEG channel selection for deep neural networks with gumbel-softmax," *Journal of Neural Engineering*, vol. 18, no. 4, pp. 0460a9, 2021.