An asymmetric heuristic for trained ternary quantization based on the weights' statistics

Firstname Lastname NAME@EMAIL.EDU
Department
University
City, State, Country

Firstname Lastname NAME@EMAIL.EDU
Department
University
City, State, Country

Editor: Editor's name

Abstract

When dealing with deep learning models and embedded systems, one of the main difficulties is the memory, computation, and energy resources needed by the former and offered by the latter. In this work, we propose a novel ternarization heuristic (model weights can take only three different values), by quantizing the model using its weights' statistics as well as asymmetric pruning. Indeed, we propose to use the mean and standard deviation of the weights to compute two asymmetric thresholds, allowing to separate the positive values from the negative ones before ternarization. We introduce two hyperparameters in these thresholds, allowing to control the trade-off between compression and classification performances. Then, after thresholding, ternarization is done as in trained ternary quantization (TTQ). We evaluate our method on three datasets, among which two are medical: a cerebral emboli (HITS) dataset, an epileptic seizure recognition (ESR) dataset, and the MNIST dataset. We tested two types of deep learning models: 2D CNNs and 1D CNN-transformers. The results show that our proposed approach, aTTQ, achieves a better trade-off between classification performance and compression rate than TTQ, for all the models and datasets. In fact, our method is able to decrease the memory requirements of a 2D CNN model on the HITS dataset by more than 20% compared to TTQ, with a degradation of the classification performance of only 0.68% in terms of Matthews correlation coefficient (MCC). For the 1D CNN-transformer on the ESR dataset, we achieve even better results, with a decrease of the memory requirements of more than 92% with respect to TTQ, with a degradation of the MCC of only 0.91%. The code is available at: https://github.com/attq-submission/aTTQ[1][2]

1. Introduction

In the past years, deep neural networks such as convolutional neural networks (CNN) or transformers, have reached state-of-the-art performances in several tasks such as computer visions (Li et al., 2022; Dosovitskiy et al., 2021), natural language processing (Wolf et al.,

1. Username attq-submission and password $cgcEsp\&GDYs@VQ42$
2. Mail: attqsubmission@gmail.comn@gmail.com (same password)

2020), and signal processing (Che et al., 2021; Vindas et al., 2022b). However, these models tend to be energy-intensive, having thousands/millions of parameters, and often requiring important computational resources (memory, GPU), which prevent its common use on embedded systems.

This last point is of particular interest as in recent years, deep learning is starting to be used more and more used in the medical field (Piccialli et al., 2021). However, the computation and memory capabilities of medical devices (especially the portable ones) are often limited, making the use of large models difficult. In this work, we focus on medical signal classification, more precisely transcranial Doppler (TCD) ultrasound for cerebral emboli (CE) classification, and electroencephalogram (EEG) for epileptic seizure recognition (ESR). These two tasks are of particular interest for public health as the former can help stroke prevention (as CE can cause ischemic stroke (Rosenkranz et al., 2006)), and both stroke and epilepsy are among the most common neurological disorders leading to disability or death (Feigin et al., 2019; Organization, 2006).

Moreover, recent works have used deep learning models to do medical signal classification based on CNNs and transformers models. For TCD signals classification, 2D CNNs on time-frequency representations (TFRs) have been used to classify artifacts from solid emboli (SE) and gaseous emboli (GE) (Vindas et al., 2022a). Other works exploit directly the signal using hybrid CNN-Transformer models or multifeature models (Vindas et al., 2022b). For ESR classification, the signal is often directly used thanks to 1D CNNs or recurrent neural networks (RNN) (Xu et al., 2020; Hilal et al., 2022). Nevertheless, even though these models have reached great performances, they often have hundreds of thousands or even millions of parameters.

To tackle this problem, recent works have proposed to reduce the memory requirements, using different model compression techniques (Cheng et al., 2018). Several works focus on quantization (Gholami et al., 2022) where the precision of the models' parameters is reduced from 32 or 64 bits to lower precision, allowing to reduce the memory requirements, without a considerable decrease of the model performances. Other works focus on removing redundant parameters of the models using pruning techniques (Hoefler et al., 2022), which also allows reducing latency and memory requirements. Moreover, efficient architectures can be manually designed, such as SqueezeNet (Iandola et al., 2016) and MobileNet (Sandler et al., 2018), or automatically designed through neural architecture search (Elsken et al., 2021). The main drawback of this last family of methods is that they are time-consuming and can require important computation resources during their development. Therefore, we are going to focus on quantization and pruning techniques.

To take advantage of both techniques (quantization and pruning) some works have proposed to apply them sequentially as they are compatible and independent (Han et al., 2016). This allows to further increase the compression rate and latency. Others works, such as binary neural networks (Rastegari et al., 2016) or trained ternary quantization (TTQ) (Zhu et al., 2017) does implicit pruning thanks to their quantization heuristic. However, to our knowledge, few works try to directly take into account pruning in the quantization mechanism.

In this paper, we propose a new ternarization heuristic based on asymmetric pruning and weights' statistics, increasing the sparsity of the weights' parameters, while keeping the rest of the quantized weights in a reduced precision, without an important degradation of the

classification performance. The rationale behind our approach is that asymmetric pruning enlarges the family of neural networks that can be explored during training, allowing to get models with a better trade-off between compression rate and classification performance. What is more, a ternarization heuristic based on the weights' statistic allows a better adaptation of the method to new datasets. On top of that, in our approach, we introduce two hyperparameters, t_{min} and t_{max}, allowing to control the sparsity rate of the quantized weights, and therefore the trade-off between compression rate and model performance. Our main contributions can be summarized as follows:

- A new heuristic for trained ternary quantization, based on the weights' statistics of the model.
- Asymmetric pruning before ternarization, allowing a better trade-off between compression and classification performance.
- Asymmetric parametrization of the sparsity rate (two hyperparameters), allowing to control the trade-off between classification performance and compression.

The rest of the paper is structured as follows. In Section 2 we present some related works. In Section 3 we introduce the proposed compression method in detail. In Section 4 we explain the datasets that we used and how they were pre-processed to obtain the final features. In Sections 5 and 6 we provide the experimental setup and we discuss the results of the different experiments, respectively. Finally, in Section 7 we conclude and give the guidelines to our future work.

Generalizable Insights about Machine Learning in the Context of Healthcare

Several medical devices are limited in terms of memory and energy resources (e.g. portable TCD, EEG or ECG devices, smartwatches, etc.). Deep learning approaches are more and more used in the medical domain because of its impressive performances in several tasks. However, these models are often resource and energy greedy, which can limit their use in practical clinical situations. In this work, we propose to reduce the memory requirements of deep learning models (CNNs and transformers), by proposing a new quantization heuristics for trained ternary quantization. This allows to increase the compression rates of the quantized models, without an important degradation of the classification performances with respect to full precision models. In summary, our method further pushed the compression capabilities of extreme trained ternary quantization while maintaining good classification performances, which can be beneficial for medical applications.

2. Related Work

2.1. Model quantization

Quantization consists in reducing the precision of the weights of a model from 32 or 64 bits, to a lower precision (Gholami et al., 2022), which can be achieved using different approaches. This can be beneficial for memory resources and inference, especially for aggressive quantization where one can profit from efficient logic/arithmetic operations (Jacob et al., 2018) or strategies (Zhu et al., 2017; Trusov et al., 2022). Early approaches were

based on matrix factorization and vector quantization (Gong et al., 2014; Kim et al., 2019), but they were mainly designed for dense layers. More recent works, have quantized convolutional layers using weight sharing, which can be done by applying weight clustering (Han et al., 2016) or Gaussian mixture models (Ullrich et al., 2017). Because of the compression, these methods often reduce the performances of the models. Some works have proposed knowledge-distillation-based techniques (Zhang et al., 2020; Sun et al., 2019) to guide the training of quantized models in order to achieve similar performances than their full-precision counterparts. To do this, the main idea is to train a quantized model with the same architecture as the full-precision one but with quantized weights, and guide it to match the soft output probabilities of the full-precision model.

By the same token, the performance drop can be reduced by combining different quantization methods in order to have different precision at each part of the model (Gholami et al., 2022; Dong et al., 2019, 2020; Yao et al., 2021). The main difficulty with these mixed quantization methods is the choice of the layers of the model that are going to be quantized and its quantization precision. To handle this, some methods have focused on metrics to evaluate the impact of quantization on the performances of the model. Indeed, (Dong et al., 2019, 2020; Yao et al., 2021) used hessian-based metrics, allowing to evaluate the flatness of the loss landscape, and therefore avoiding the extreme quantization of layers with irregular landscapes.

Finally, quantization methods are rarely straightforward to implement during the training of the models because of its non-differentiable nature. To tackle this problem, the different methods often use the straight through estimator (STE) (Yin et al., 2019; Zhu et al., 2017; Bhalgat et al., 2020), or reformulate the quantization as a differentiable problem in order to be able to use gradient descent (Yang et al., 2019). However, for extreme quantization the optimization of quantized models can be difficult as it often introduces significant noise during training (Fan et al., 2020).

2.2. Model pruning

Pruning consists in removing redundant parameters of a model by setting them to zero. This is particularly interesting in neural network models as they are over-parametrized, so pruning can act as a regularizer improving the generalization capabilities of the models (Hoefler et al., 2022). This family of methods can also improve memory and latency, as the obtained parameters tensors are sparse (Gondimalla et al., 2019). Moreover, different approaches can be used to prune the parameters of a model. Some works remove the weights with minimal norm (L1 or L2 norm), using a pre-defined threshold or a number of weights to prune (Han et al., 2016). More complex methods choose the weights to remove by computing their importance based on the statistics of the following layer's parameters (Luo et al., 2017), or by creating subsets of neurons to fuse together, based on determinantal point processes. By the same token, other approaches have tried to overcome the non-differentiability of threshold operators during pruning by using reinforcement learning (He et al., 2018), genetic algorithms (Xu et al., 2021), or differentiable threshold functions (Manessi et al., 2017).

Finally, even if the aforementioned methods allow removing an important percentage of the parameters of the network by setting them to zero, the rest of the parameters remain

in full-precision (32 or 64 bits), preventing the use of efficient operations and increasing the memory requirements with respect to quantization methods. To address this, some works have tried to combine pruning and quantization with different approaches. (Han et al., 2016; Park et al., 2018) tried a sequential combination of pruning and quantization (combined with other techniques), whereas (Tung and Mori, 2020) used Bayesian optimization techniques and (Ullrich et al., 2017) used soft-weight sharing to achieve both pruning and quantization.

3. Methods

In this section, we present a new quantization heuristic for trained ternary quantization (TTQ), called aTTQ. Our method relies on two assumptions: (1) asymmetric pruning improves the trade-off between compression and classification performance, and (2) pruning and quantization based on the weights' statistics allows a better adaptation to new datasets and models. The first assumption can be justified by the fact that asymmetric pruning increase the model search space during training (a relaxation of the optimization scheme). The second assumption is plausible as the thresholds used for pruning depend on the weights' statistics which change during the training/fine-tuning step.

The global pipeline of our approach can be found in figure 1

3.1. Preliminaries: Ternary quantization

Ternary quantization consists on quantizing the weights of a given layer during training, using only three possible values (ternary values): -1, 0 and 1. To avoid drastic performance drops, given a layer to quantize B, Li et Liu (Li and Liu, 2016) introduce a scaling factor W_B depending on the weights' statistics, thus quantizing in the set of values $\{-W_B, 0, W_B\}$. To improve this, TTQ used two learnable scaling parameters, W_l for the negative values, and W_r for the positive values, thus quantizing in the set of values $\{-W_l, 0, W_r\}$. Quantization of a full-precision weight tensor w into its ternary counterpart w_t is done as follows:

$$w_t = \begin{cases} W_l \text{ if } w < -\Delta_l \\ 0 \text{ if } w \in [-\Delta_l, \Delta_l] \\ W_r \text{ if } w > \Delta_l \end{cases} \tag{1}$$

where $W_l, W_r \in \mathbb{R}$ are learnable scaling parameters (per layer), $\Delta_l = t \times max(|w|)$ is a threshold that depends on a hyperparameter t, controlling the sparsity of TTQ. Following the results in (Zhu et al., 2017), we fix $t = 0.05$ for the rest of the paper.

3.2. Weights statistics based pruning

We propose novel asymmetric ternarization heuristic, where the quantization threshold does not depend on the maximum absolute value of the values of the weights' tensor, but it depends on the statistics of it:

$$w_t = \begin{cases} W_l \text{ if } w < \Delta_{min} \\ 0 \text{ if } w \in [\Delta_{min}, \Delta_{max}] \\ W_r \text{ if } w > \Delta_{max} \end{cases} \tag{2}$$

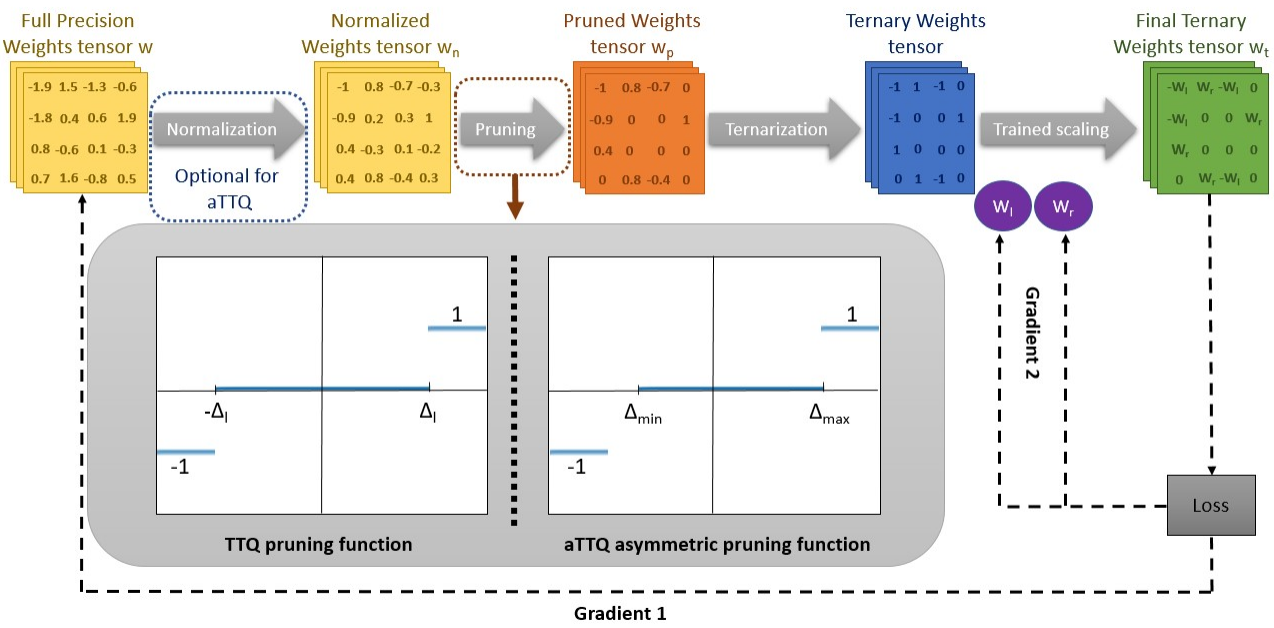

Figure 1: Proposed alternative TTQ (aTTQ) method. The main difference with respect to TTQ lies in the pruning mechanism done before ternarization: we use two asymmetric thresholds, Δ_{min} and Δ_{max} instead of one symmetric threshold $\Delta_{min} = -\Delta_{max} = -\Delta_L$. The normalization step, always used in TTQ, is optional in our approach.

where w and w_t are the full-precision and ternarized weights, $W_l, W_r \in \mathbb{R}$ are learnable scaling parameters, $\Delta_{min} = \mu_w + t_{min} \times \sigma_w$ and $\Delta_{max} = \mu_w + t_{max} \times \sigma_w$ are thresholds for pruning, and t_{min} and t_{max} are two hyperparameters to tune with the constraint $t_{min} \leq t_{max}$, controlling the sparsity rate. Indeed, if $t_{min} > t_{max}$ the $\Delta_{min} > \Delta_{max}$ and therefore $\forall w \in \mathbb{R}, w_t \neq 0$, so no pruning is done.

Contrary to TTQ, in this approach we have two asymmetric thresholds, one for the positive weights, and one for the negative ones, giving more degrees of freedom for quantization and for pruning (figure 1). The gradients of L, the loss to optimize, can be computed using the straight forward estimator as in TTQ, the only difference is that the used threshold Δ_l is replaced by the thresholds Δ_{min} and Δ_{max}:

$$\frac{\partial L}{\partial w} = \begin{cases} W_l \times \frac{\partial L}{\partial w_t} \text{ if } w < \Delta_{min} \\ 0 \text{ if } w \in [\Delta_{min}, \Delta_{max}] \\ W_r \times \frac{\partial L}{\partial w_t} \text{ if } w > \Delta_{max} \end{cases} \tag{3}$$

3.3. Layer selection

To select the layers that are going to be quantized, we used the Hessian-based metric introduced in (Dong et al., 2020). This metric is based on the trace of the Hessian matrix, and allows quantifying the curvature of the loss landscape. The rationale behind this metric is that, layers with flat loss landscapes (small values of the trace of the Hessian matrix) are more robust to quantization as it is more difficult to escape the reached local minima than layers with a curved loss landscapes.

3.4. Evaluation metrics

To compare the sparsity and compression of the obtained models, we introduce different metrics defined in the following paragraphs. We denote as $\mathcal{M}_{FP}$ the full precision model and $\mathcal{M}_Q$ a quantized model obtained from $\mathcal{M}_{FP}$. By the same token, we denote as $nbits$ a function allowing to count the number of bits necessary to store the (nonzero) weights of a model, and $nqw/nzqw$ two functions allowing to count the number of weights/zero weights of a model, among the weights selected for quantization using the Hessian-based metric of section 3.3.

Sparsity. To quantify the sparsity achieved during the quantization of the different models, we introduce the sparsity rate over the quantized weights, noted SRQW and defined as:

$$SRQW(\mathcal{M}_{FP}, \mathcal{M}_Q) = \frac{nzqw(\mathcal{M}_Q)}{nqw(\mathcal{M}_{FP})}$$

where higher values of $SRQW$ indicates sparser models.

Compression. We want to quantify the compression reached thanks to ternarization and pruning (simultaneously). To do this, we introduce the compression rate, CR, defined as follows:

$$CR(\mathcal{M}_{FP}, \mathcal{M}_Q) = \frac{nbits(\mathcal{M}_Q)}{nbits(\mathcal{M}_{FP})}$$

To facilitate comparison between methods, we work with the compression rate gain CR_G, defined as:

$$CR_G(\mathcal{M}_{FP}, \mathcal{M}_Q) = 1 - CR(\mathcal{M}_{FP}, \mathcal{M}_Q)$$

We denote as CR_G^T and CR_G^Q the compression rate gains of the whole model and the layers selected for quantization, respectively.

4. Data

To train and evaluate our proposed method, we used one nonmedical dataset, MNIST (LeCun and Cortes, 2010), and two medical datasets: a private Transcranial Doppler (TCD) dataset, called the HITS dataset (Vindas et al., 2022a), and one electroencephalogram (EEG) public dataset from the UCI repository (Andrzejak et al., 2002). For the MNIST dataset, in order to reduce computation resources, we only used 10% of the training samples to train the different models. The two other datasets will be detailed hereafter.

4.1. HITS dataset

We used a private transcranial Doppler (TCD) high intensity transient signals (HITS) dataset described in (Vindas et al., 2022b). Hereafter, we specify how the data was acquired, as well as the preprocessing steps.

4.1.1. Data acquisition

TCD recordings between 30 and 180 minutes were acquired from 39 subjects coming from neurovascular and cardiovascular care units from 11 different healthcare centers. These patients were aged between 21 and 85 years old (median age of 63), and 15 were male, 19 female, and 5 unknown. The recordings were performed using two different TCD devices from Atys Medical, the TCD-X Holter and the WAKIe R3, with a 1.5 MHz robotized probe. Moreover, the recordings conditions were heterogeneous as some patients can have one diagnosed pathologies (carotid stenosis, patent foramen ovale, atrial fibrillation), some received a contrast agent injection (iodine-containing agent or sonovue), and some patients were monitored during surgical procedures (atrial fibrillation ablation and transcatheter aortic valve replacement).

Furthermore, the recordings were carried out in the middle cerebral artery (MCA) and the objective was to detect cerebral emboli (solid or gaseous). Therefore, we obtained the following acquisition information from Atys Medical for emboli detection on the MCA:

- Insonation probe frequency: 1.5 MHz.
- Insonation depth: 45-55 mm.
- Pulse repetition frequency: 4.4-6.2 kHz.
- Sample volume: 8-10 mm^3

4.1.2. Data pre-processing

From the recordings, 1541 high intensity transient signals (HITS) are extracted, and distributed in three classes: artifacts (403 HITS), gaseous emboli (569 HITS), and solid emboli (569 HITS)[3]. Into the bargain, these signals are extracted using the criteria of the Ninth International Cerebral Hemodynamic Symposium (1995): (1) duration smaller than 300 ms, (2) unidirectional in the time-frequency domain, (3) intensity increase with respect to the background signal (blood flow) of at least 3 dB, and (4) musical characteristic in the audible sound, similar to a "chirp". For more details about the distribution of HITS per class and per subject, we invite the reader to see appendix A.

Moreover, from each HITS, we get two types of representations: the raw signal (with a sample frequency between 1100 Hz and 1400 Hz) and a logarithmic scale spectrogram, both normalized with the mean and standard deviation of the training set. The logarithmic scale spectrograms were computed using a length of the windowed signal after padding of $n_{FFT} = 128$, a size of overlap of $n_{overlap} = 8$, and a Blackman window following (Vindas et al., 2022b).

3. Four HITS were not taken into account as the expert annotators were not able to distinguish them between solid or gaseous emboli

Finally, the 1541 used HITS were split into two subsets in a subject-wise manner, one for training (63% of the samples), and one for testing (37% of the samples). For more details about this dataset and the pre-processing steps, we refer the reader to (Vindas et al., 2022b,a).

4.2. ESR dataset

To validate our proposed method we also used a public medical signal dataset: the pre-processed Epileptic Seizure Recognition dataset (ESR) (Andrzejak et al., 2002) from the UCI repository[4]. The dataset is composed of 11 500 pre-processed[5] electroencephalogram (EEG) signals, distributed in five classes equally distributed: (1) seizure activity and (2)-(5) no seizure activity. As most works using this dataset, we do binary classification between seizure activity (2300 samples) and no seizure activity (9200 samples), but the reader can refer to (Andrzejak et al., 2002) or appendix B to get more details about the five classes.

Furthermore, as for the HITS dataset, we extract two types of representations: the raw signal sampled at 176 Hz, and a logarithmic scale spectrogram, both normalized using the mean and standard deviation of the training set. The logarithmic scale spectrogram was obtained using $n_{FFT} = 32$, $n_{overlap} = 4$, and a Blackman window.

Finally, as for the HITS dataset, the dataset was split into two subsets: one for training containing 90% of the samples, and one for testing containing 10% of the samples. However, contrary to the HITS dataset, the split was not done subject-wise because the structure of the ESR dataset does not permit it.

5. Experiments

We conduct three experiments to evaluate the main contributions of our method. The first experiment compares our proposed weight-statistics quantization heuristics, aTTQ, with respect to TTQ in terms of classification and compression performances. The second experiment studies the influence of the two hyperparameters of our approach, t_{min} and t_{max}, on two different datasets with two different models. The third experiment focuses on the influence of weight-normalization in the final performance of the different compressed models. The code is available at: https://github.com/attq-submission/aTTQ[6][7]

5.1. Architectures

We used three different models, depending on the dataset: one vanilla 2D CNN for the MNIST dataset, and one 2D CNN and 1D CNN-transformer for the medical signals datasets.

The vanilla 2D CNN MNIST model is composed of two convolutional layers, followed by 2D max polling and ReLU activation applied for both, and dropout after the second convolutional layer. Then a fully connected (FC) classifier was applied, composed of two linear layers, followed by dropout and ReLU activation for the first linear layer, and logarithmic softmax for the second linear layer.

4. We used the public available version found at https://www.kaggle.com/datasets/harunshimanto/epileptic-seizure-recognition
5. We refer the reader to the UCI repository for the pre-processing details.
6. Username attq-submission and password $cgcEsp\&GDYs@VQ42$
7. Mail: attqsubmission@gmail.comn@gmail.com (same password)

Moreover, for the HITS and ESR datasets, we used the same architectures as (Vindas et al., 2022b)[8]. For the HITS dataset, the hyperparameters of the 1D CNN-transformer were the same as (Vindas et al., 2022b). For the ESR dataset, we used the following hyperparameters: the last 1D convolutional layer is applied twice, 4 attention heads (per multi-head attention) are used for the 4 Transformer encoder layers, a Transformer intermediate hidden dimension of 8 is employed, and a dimension of 4 for the projected representation is used for final classification. At last, for the 2D CNN, all the hyperparameters were the same as (Vindas et al., 2022b), except for the number of initial convolutional filters which were set to 64.

5.2. Training parameters

Table 1 presents the training parameters used for the different models on the different datasets. All the models were trained with cross-entropy loss function optimized using Adamax optimizer for the 2D CNN models, and Noam for the 1D CNN-transformer models, with $\beta_1 = 0.0$, $\beta_2 = 0.999$ and 4000 warm up steps for all the models, except for the TTQ and aTTQ quantized HITS models, which were trained with 700 warm up steps. To handle class imbalance, class weights (King and Zeng, 2001; Pedregosa et al., 2011) were applied to all the models. We used a weight decay of 10^{-7} for almost all the models, except for the 1D CNN-transformer models trained on the ESR dataset, and the 2D CNN trained on the MNIST dataset which used a weight decay of 0, and the full-precision ESR 2D CNN which used a weight decay of 10^{-5}. Additionally, we used a batch size of 32 for all the models except for the ESR 1D CNN-transformer models which used a batch size of 64.

Furthermore, using the approach presented in 3.3, we selected the different layers to quantize, and we quantized only their weights, without the biases. For the MNIST 2D CNN, all the convolutional layers were quantized; for the 2D CNN used for the medical datasets, we quantized all the convolutional layers except the first one; for the 1D CNN-Transformer, we quantized the second convolutional layer, plus the second linear layer of all the encoder layers of the transformer encoder. The percentage of selected weights to quantize (for the whole model) can be found in the last column of table 1.

5.3. Evaluation metrics

We used the Matthew correlation coefficient (MCC) to measure the classification performances of the models (well-suited for imbalanced datasets), and $SRQW$, CR_G^T and CR_G^Q to measure the compression performances. Finally, experiments 1 and 3 were repeated 10 times, and experiment 2 was repeated 5 times for statistical purposes, and the reported metrics correspond to the mean computed on the test sets.

5.3.1. Experiment 1: Comparison with trained ternary quantization (TTQ)

The objective of this experiment is to compare the performance of the proposed quantization heuristic, aTTQ, with respect TTQ. This comparison is done with respect to two main

8. For details about the network architectures, we refer the reader to (Vindas et al., 2022b), appendix C or the GitHub repository.

Table 1: Training parameters for the different models based on the dataset and the used ternarization method. In the last column, we specified the percentage of weights of the model that are going to be quantized, selected using the Hessian-based metric in section 3.3.

Dataset	Model	Quant. method	t_{min}	t_{max}	Learning rate	Epochs	No. params.	% weights to quantize
HITS	2D CNN	FP	-	-	10^{-3}	50	1 681 923	-
		TTQ	-	-	3×10^{-3}	50		92.05
		aTTQ	-4	0	10^{-4}	150		
	1D CNN-trans.	FP	-	-	7×10^{-2}	150	766 271	-
		TTQ	-	-	10^{-4}	50		14.97
		aTTQ	-2	1.5	5×10^{-5}	100		
ESR	2D CNN	FP	-	-	10^{-3}	100	1 555 842	-
		TTQ	-	-	10^{-3}	50		99.51
		aTTQ	-3	1	10^{-3}	200		
	1D CNN-trans.	FP	-	-	3×10^{-1}	100	109 942	-
		TTQ	-	-	10^{-3}	100		24.22
		aTTQ	-2	1	5×10^{-4}	100		
MNIST	2D MNIST CNN	FP	-	-	10^{-3}	70	9 840	-
		TTQ	-	-	10^{-4}	200		53.35
		aTTQ	-1	0.5	10^{-3}	200		

aspects: classification performance and the compression performance. The results can be found in table 2 and figure 2.

First, we can note that, in terms of compression and sparsity metrics, aTTQ outperforms TTQ by a large margin. Indeed, in terms of sparsity, aTTQ improves the sparsity rate of the quantized weights (SRQW) of at least 2.7% (and up to 86.8%) with respect to TTQ. Indeed, this sparsity rate is over 13 times larger for aTTQ for the 1D CNN-transformer model trained on the HITS dataset, passing from 6.75% for TTQ to 93.58% for aTTQ. What is more, similar results are observed for the compression rate gain of the quantized layers, CR_G^Q. However, even though a similar behavior is observed for the compression rate gain of the whole model, the increase of aTTQ with respect to TTQ are smaller.

Second, compared to the full precision model, TTQ and aTTQ achieve similar classification performances. Indeed, for the HITS dataset, we observe a maximum MCC drop with respect to the FP model of 3.70% and 3.02% for aTTQ and TTQ respectively. On the contrary, for the 1D-CNN transformer models on the ESR dataset, we note an MCC increase of 1.01% and 1.92% for aTTQ and TTQ, respectively.

Finally, we can observe that, improving the compression rate with aTTQ comes at the cost of a classification performance drop. However, this performance drop is of the same order as with TTQ for most of the datasets and models. Globally, TTQ weakly exceeds aTTQ on almost all the datasets, for almost all the models in terms of classification performance, with an MCC margin going from 0.68% to 2.77%. This is not true for the 2D CNN in the MNIST dataset, where aTTQ outperforms TTQ by a MCC margin of 1.53%.

Table 2: Results of experiment 1, in %. FP corresponds to the full-precision model where no quantization has been done. ΔMCC corresponds to the difference between the MCC of the full precision model and the MCC of the quantized model. CR_G^T, and CR_G^Q evaluate the compression performance of each quantization method and were introduced in 3.4.

Dataset	Model	Quant. method	Norm.	$CR_G^T \uparrow$	$CR_G^Q \uparrow$	MCC ↑	Δ MCC ↑
HITS	2D CNN	FP	No	-	-	89.84 ± 3.09	-
		TTQ	Yes	24.96 ± 2.25	27.12 ± 2.44	$\mathbf{86.82 \pm 2.29}$	$\mathbf{-3.02}$
		aTTQ		$\mathbf{42.98 \pm 0.23}$	$\mathbf{46.69 \pm 0.25}$	86.14 ± 3.37	-3.70
	1D CNN-trans.	FP	No	-	-	82.64 ± 1.77	-
		TTQ	Yes	0.14 ± 0.04	0.91 ± 0.27	$\mathbf{83.22 \pm 2.36}$	$\mathbf{+0.58}$
		aTTQ	No	$\mathbf{13.94 \pm 0.02}$	$\mathbf{93.17 \pm 0.16}$	81.66 ± 4.17	-0.98
ESR	2D CNN	FP	No	-	-	92.81 ± 3.53	-
		TTQ	Yes	85.61 ± 1.37	86.03 ± 1.37	$\mathbf{95.00 \pm 1.11}$	$\mathbf{+2.19}$
		aTTQ	No	$\mathbf{88.48 \pm 0.44}$	$\mathbf{88.91 \pm 0.45}$	92.41 ± 2.22	-0.40
	1D CNN-trans.	FP	No	-	-	94.33 ± 1.51	-
		TTQ	Yes	11.40 ± 2.61	47.07 ± 10.79	$\mathbf{96.25 \pm 0.79}$	$\mathbf{+1.92}$
		aTTQ	No	$\mathbf{21.02 \pm 0.15}$	$\mathbf{86.78 \pm 0.63}$	95.34 ± 0.79	$+1.01$
MNIST	2D MNIST CNN	FP	No	-	-	94.39 ± 0.46	-
		TTQ	Yes	13.86 ± 2.33	25.97 ± 4.37	92.09 ± 0.89	-2.30
		aTTQ	No	$\mathbf{28.98 \pm 1.26}$	$\mathbf{54.32 \pm 2.36}$	$\mathbf{93.62 \pm 0.96}$	$\mathbf{-0.77}$

5.3.2. Experiment 2: Influence of t_{min} and t_{max}

The objective of this experiment is twofold: (1) show the interest of using asymmetric thresholds, and (2) study the influence of the hyperparameters t_{min} and t_{max} of aTTQ on the performances of the models. To do this, we trained the 2D CNN model of experiment 1 on the subset of the MNIST dataset, and the 1D CNN-transformer model of the same experiment on the ESR dataset, varying the values of x and y in $\{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}$. The results can be found in figure 3.

First, we can observe that, the best classification performances are not obtained for symmetrical values of t_{min} and t_{max}. Indeed, for both models on both datasets, the best results are obtained when $t_{min} \leq 0$ and $t_{max} \geq 0$, achieving 93.56% MCC for the 2D CNN on the MNIST dataset, and 95.09% MCC for the 1D CNN-transformer on the ESR dataset. What is more, the best SRQW values are also obtained for the same ranges of (asymmetric) values of t_{min} and t_{max}.

Moreover, we observe a trade-off between compression and classification performance as the best performing models in terms of MCC are not the ones having the higher SRQW (thus the higher compression rate). In terms of sparsity, the higher the gap between t_{min} and t_{max} the higher the sparsity (larger range of weights values that maps to zero). When this gap is large enough, the classification performance is often worse than smaller gaps (translating in smaller sparsity rates).

Finally, it is interesting to note that, small sparsity rates (small gap between t_{min} and t_{max}) do not always give models with the highest classification performances. Indeed, we observe that models with a sparsity rate close 0% tend to give models with worse classification performances than higher sparsity rate models.

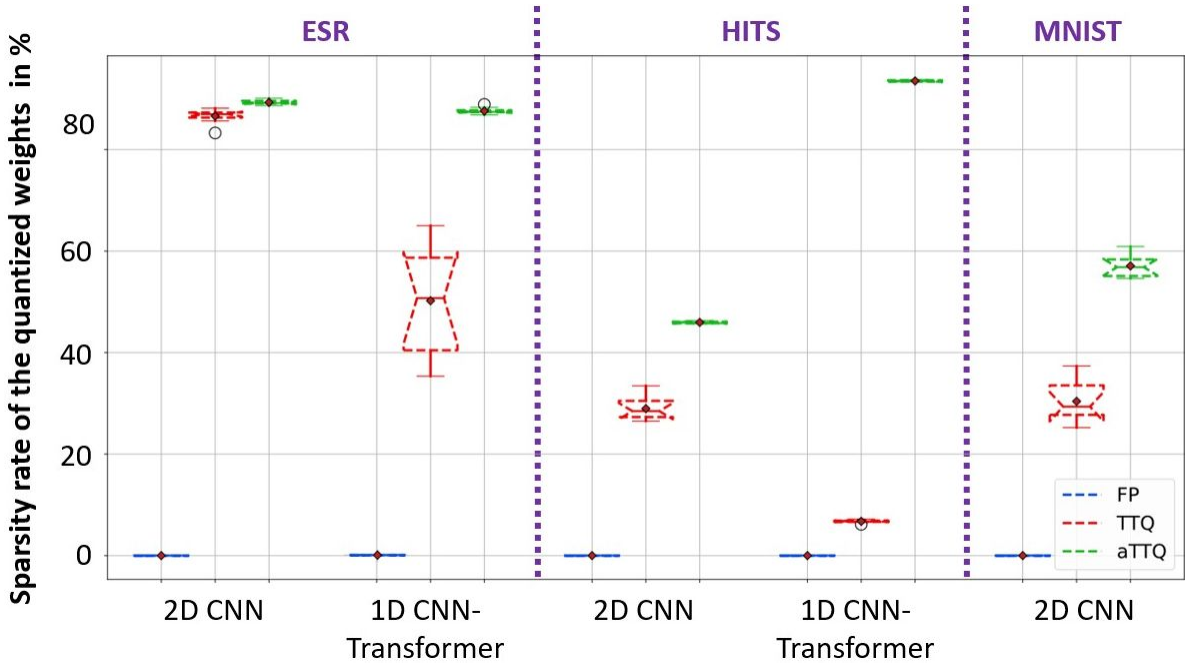

Figure 2: Results of experiment 1. We show the sparsity rates of the quantized weights, SRQW, in %. The blue boxes correspond to the full-precision model (FP), where none of the weights to quantized are set to 0 (so 0% if SRQW). The red and green curves correspond to the TTQ and aTTQ models, respectively.

5.3.3. Experiment 3: Influence of weight normalization

The objective of this experiment is to study the influence of normalization on the performances of the proposed aTTQ ternarization method. To do this, we ternarize all the models presented in experiment 1 with and without normalization of the weights to quantize. Results can be found in table 3.

First, we can notice that, for most of the datasets and models with or without normalization, the classification performances are similar. More interestingly, for the 2D CNN models trained on the ESR and MNIST datasets, the use of normalization have a large negative impact on the classification performances, with a gap of at least 84.16% in terms of MCC between the non-normalized and normalized models. However, on the HITS dataset, the 2D CNN quantized model with normalization outperforms the one without normalization by a margin of 0.69% in terms of MCC, while the compression performances remain the same. Finally, for all the datasets and models, normalization do not have a significant impact on the compression performances, as the different metrics are almost identical for both normalized and non-normalized models

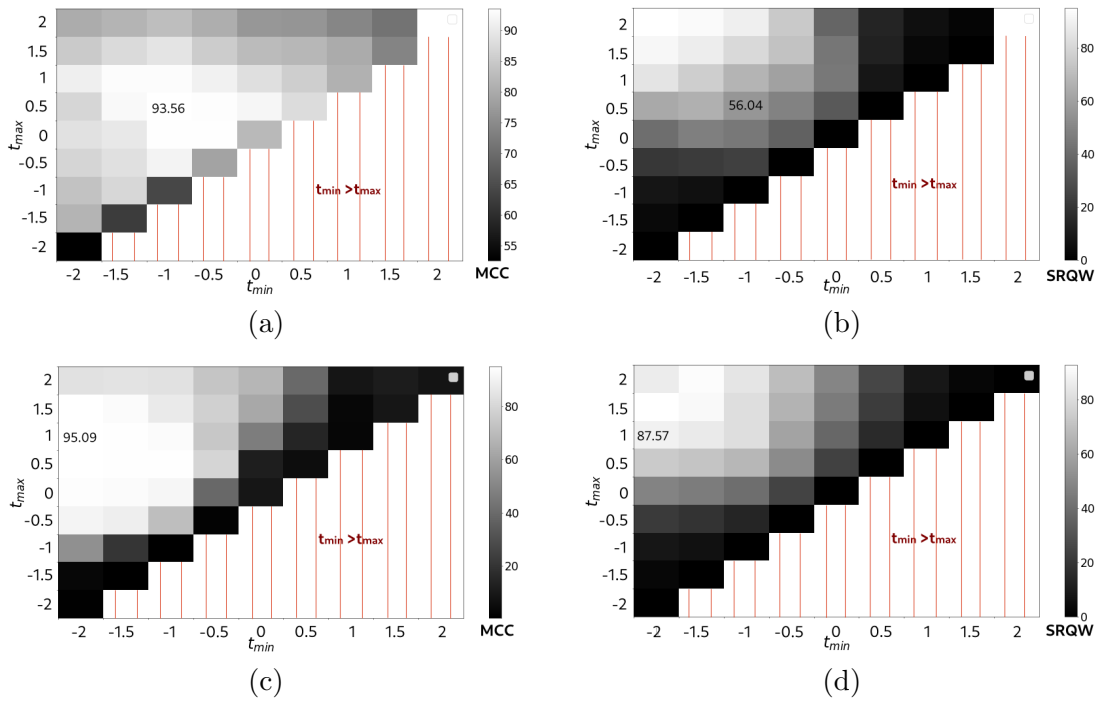

Figure 3: Results of experiment 2. (a) MCC for the 2D CNN model trained on the MNIST dataset. (b) SRQW for the 2D CNN model trained on the MNIST dataset. (c) MCC for 1D CNN-transformer model trained on the ESR dataset. (d) SRQW for 1D CNN-transformer model trained on the ESR dataset. The x-axis corresponds to the different tested values of t_{min} and the y-axis to the different values of t_{max}. All the values are given in %.

6. Discussion

Experiment 1: Comparison with trained ternary quantization (TTQ). The results of experiment 1 confirm the interest of our quantization approach, aTTQ, in terms of compression and classification, with respect to TTQ. Our method always outperforms TTQ by a large margin in terms of compression and sparsity rates, even though it often has slightly smaller classification performances than TTQ. Therefore, aTTQ offers a better trade-off between classification and compression performances. Indeed, in some cases, a slight decrease in the classification performance can be justified by lighter models. A good example are (medical) embedded applications, where, if the best model does not fit in the device, the good (not the best) classification performances, are relevant.

Furthermore, we can note that, for the used compression metrics, the observed increase in the performances is not the same for all the metrics. Indeed, the compression metrics focusing only on the layers that are quantized are often significantly higher than the compression metric using all the layers in the model. This can be explained by the fact that the quantized layers do not always have the majority of the parameters of the model. There-

Table 3: Results of experiment 3. The normalization column indicates if normalization of the weights by the maximum of the absolute value of the weights was performed before quantization. CR_G^Q evaluate the compression performance of each quantization method and were introduced in 3.4.

Dataset	Model	Normalization	MCC ↑	CR_G^Q ↑
HITS	2D CNN	No	85.45 ± 3.33	46.69 ± 0.25
		Yes	$\mathbf{86.14 \pm 3.37}$	46.69 ± 0.25
	1D CNN-trans.	No	$\mathbf{81.66 \pm 4.17}$	93.11 ± 0.16
		Yes	80.45 ± 3.59	93.17 ± 0.16
ESR	2D CNN	No	$\mathbf{92.41 \pm 2.22}$	88.91 ± 0.45
		Yes	8.25 ± 13.47	88.95 ± 0.58
	1D CNN-trans.	No	95.34 ± 0.79	86.78 ± 0.63
		Yes	$\mathbf{95.40 \pm 0.73}$	86.77 ± 0.62
MNIST	2D MNIST CNN	No	$\mathbf{93.62 \pm 0.96}$	54.32 ± 2.36
		Yes	0 ± 0	60.36 ± 3.01

fore, even if all the parameters of those layers are removed, the compression will not be significant.

Finally, we can see that, in some cases, the classification performances can be increased after quantization. This can be justified by three factors. The first factor is that sparsity can act as regularization, as several works have shown it (Hoefler et al., 2022). The second factor is that quantization can also help regularization, as deep neural networks are highly over-parametrized and redundant. At last, the quantized models are obtained from pre-trained full precision models, and the chosen layers to quantized are based on a hessian-based quantization sensitivity metric. Therefore, the fine-tuning quantization step could help the models to get closer to a local minimum, as the loss landscape should be relatively flat for the chosen layers to quantize.

Experiment 2: Influence of t_{min} and t_{max}. The results of experiment 2 showed the importance of the asymmetric pruning as well as the importance of the choice of t_{min} and t_{max} for aTTQ.

Indeed, the results showed that asymmetric thresholds obtained through asymmetric values of t_{min} and t_{max} allow achieving better classification performances than symmetric thresholds $t_{min} = -t_{max}$, while still achieving a good sparsity rate for the quantized weights. This can be explained by the fact that, within a neural network, positive and negative values of the weights do not necessarily have the same impact on the final classification performances. Moreover, when normalization is not done, the minimum and maximum values of the weights are not necessarily opposites, so the two thresholds should be adapted to this situation.

What is more, the results also show the trade-off existing between classification performance and model compression through sparsity. In fact, very high sparsity rates tend to lead to a decrease in the classification performances. However, this decrease is not always

significant, whereas the gain in sparsity rate of the quantized layers is. Therefore, based on the application, if memory requirements are an important factor, higher sparsity rates could be chosen despite the reduction in classification performances. In our case, we decided to choose the models giving the higher classification performance, without taking into account the sparsity rates, but this selection strategy can be adapted based on the targeted application. This is an advantage of our method as the two hyperparameters t_{min} and t_{max} allows controlling this trade-off between compression and classification performances.

Finally, we can notice the regularization effect of sparsity thanks to figure 3. Indeed, when the sparsity rate of the quantized weights increases, the classification performance tends to increase up to a certain point, when the classification performance starts decreasing.

Experiment 3: Influence of weight normalization Experiment 3 shows the influence of weight normalization on our proposed method, aTTQ. The results showed that, for almost all the models and datasets, better or similar classification performances are obtained when the model's weights to quantize are not normalized, and this without degrading the compression performances. What is more, the only case where normalization outperform non-normalization, was the case of the 2D CNN model trained on the HITS dataset (margin of 0.69% in terms of MCC). Therefore, we recommend using our method without prior normalization of the weight to quantize.

Limitations. In this work we showed the advantage of our aTTQ approach, over classical TTQ, especially in terms of compression. However, our method have some limitations. First, it is difficult to choose t_{min} and t_{max} to match a prescribed trade-off between classification performance and compression rate. Second, in order to take advantage of the compressed methods in terms of energy consumption and inference time, specialized hardware has to be designed to perform energy and inference efficient operations with the obtained quantized model. Finally, as we do extreme quantization, our approach is not applicable to all the layers of a given model without important classification performance degradation.

7. Conclusion

In this paper, we proposed to modify the quantization heuristics of trained ternary quantization (TTQ), in order to improve the trade-off between classification performance and compression rate. Indeed, instead of using symmetric thresholds for the positive and negative weights to quantize, we propose to use asymmetric thresholds computed using the weights' statistics (mean and standard deviation) and two hyperparameters, t_{min} and t_{max}, controlling the sparsity rate of the quantized weights. Extensive experiments on three datasets and two types of models, demonstrate the effectiveness of our method, being able to improve the compression performances up to 92.26% in terms of CR_G^Q (compression rate gain of the layers selected for quantization), with similar classification results as TTQ (degradation of 1.56% in terms of MCC).

In future work, we plan to develop specialized hardware to efficiently do the operations needed by our quantized (sparse) models, allowing to accelerate inference and reduce energy consumption in practice. Moreover, to increase model compression and reduce energy consumption, we plan to use mixed quantization to quantize the entire model, without important degradation of the classification performances.

References

Ralph Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 64:061907, 01 2002.

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 (CVPR) Workshops*, June 2020.

Chao Che, Peiliang Zhang, Min Zhu, Yue Qu, and Bo Jin. Constrained transformer network for ecg signal processing and arrhythmia classification. *BMC Medical Informatics and Decision Making*, 21, 2021.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, 35(1):126–136, 2018. doi: 10.1109/MSP.2017.2765695.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18518–18529. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d77c703536718b95308130ff2e5cf9ee-Paper.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(1):1997–2017, mar 2021. ISSN 1532-4435.

Angela Fan, Pierre Stock*, , Benjamin Graham, Edouard Grave, Remi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme model compression. 2020.

Valery Feigin, Emma Nichols, Tahiya Alam, Marlena Bannick, Ettore Beghi, Natacha Blake, William Culpepper, E. Dorsey, Alexis Elbaz, Richard Ellenbogen, James Fisher, Christina Fitzmaurice, Giorgia Giussani, Linda Glennie, Spencer James, Catherine Johnson, Nicholas Kassebaum, Giancarlo Logroscino, Benoît Marin, and Theo Vos. Global, regional, and national burden of neurological disorders, 1990-2016: a systematic analysis

for the global burden of disease study 2016. *The Lancet Neurology*, 18:459–480, 05 2019. doi: 10.1016/S1474-4422(18)30499-X.

Asghar Gholami, Sehoon Kim, Dong Zhen, Zhewei Yao, Michael Mahoney, and Kurt Keutzer. *A Survey of Quantization Methods for Efficient Neural Network Inference*, pages 291–326. 01 2022. ISBN 9781003162810. doi: 10.1201/9781003162810-13.

Ashish Gondimalla, Noah Chesnut, Mithuna Thottethodi, and T. N. Vijaykumar. Sparten: A sparse tensor accelerator for convolutional neural networks. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, page 151–165, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369381. doi: 10.1145/3352460.3358291. URL https://doi.org/10.1145/3352460.3358291.

Yunchao Gong, L. Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization, 2014.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1510.00149.

Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 815–832, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01234-2.

Anwer Mustafa Hilal, Amani Abdulrahman Albraikan, Sami Dhahbi, Mohamed K. Nour, Abdullah Mohamed, Abdelwahed Motwakel, Abu Sarwar Zamani, and Mohammed Rizwanullah. Intelligent epileptic seizure detection and classification model using optimal deep canonical sparse autoencoder. *Biology*, 11(8), 2022. ISSN 2079-7737.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(1), jul 2022. ISSN 1532-4435.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv*, abs/1602.07360, 2016.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Hyeji Kim, Muhammad Umar Karim Khan, and Chong-Min Kyung. Efficient neural network compression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12561–12569, 2019. doi: 10.1109/CVPR.2019.01285.

Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9: 137–163, Spring 2001.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Fengfu Li and Bin Liu. Ternary weight networks. *ArXiv*, abs/1605.04711, 2016.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. doi: 10.1109/TNNLS.2021.3084827.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. pages 5068–5076, 10 2017. doi: 10.1109/ICCV.2017.541.

Franco Manessi, Alessandro Rozza, Simone Bianco, Paolo Napoletano, and Raimondo Schettini. Automated pruning for deep neural network compression. 12 2017.

Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Basic identification criteria of doppler microembolic signals. *Stroke*, 26(6):1123, 1995.

World Health Organization. *Neurological disorders: public health challenges*. World Health Organization, 2006.

Mi Sun Park, Xiaofan Xu, and Cormac Brick. Squantizer: Simultaneous learning for both sparse and low-precision neural networks. *CoRR*, abs/1812.08301, 2018. URL http://arxiv.org/abs/1812.08301.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Francesco Piccialli, Vittorio Di Somma, Fabio Giampaolo, Salvatore Cuomo, and Giancarlo Fortino. A survey on deep learning in medicine: Why, how and when? *Information Fusion*, 66:111–137, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.09.006. URL https://www.sciencedirect.com/science/article/pii/S1566253520303651.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 525–542, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

M. Rosenkranz, J. Fiehler, W. Niesen, C. Waiblinger, B. Eckert, O. Wittkugel, T. Kucinski, J. Röther, H. Zeumer, C. Weiller, and U. Sliwka. The amount of solid cerebral microemboli during carotid stenting does not relate to the frequency of silent ischemic lesions. *American Journal of Neuroradiology*, 27(1):157–161, 2006. ISSN 0195-6108, 1936-959X. URL http://www.ajnr.org/content/27/1/157.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

S. Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

A. Trusov, E. Limonova, D. Nikolaev, and V. V. Arlazarov. Fast matrix multiplication for binary and ternary cnns on arm cpu. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3176–3182, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society. doi: 10.1109/ICPR56361.2022.9956533. URL https://doi.ieeecomputersociety.org/10.1109/ICPR56361.2022.9956533.

Frederick Tung and Greg Mori. Deep neural network compression by in-parallel pruning-quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3): 568–579, 2020. doi: 10.1109/TPAMI.2018.2886192.

Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HJGwcKclx.

Yamil Vindas, Blaise Kévin Guépié, Marilys Almar, Emmanuel Roux, and Philippe Delachartre. Semi-automatic data annotation based on feature-space projection and local quality metrics: an application to cerebral emboli characterization. *Medical Image Analysis*, page 102437, 2022a. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102437. URL https://www.sciencedirect.com/science/article/pii/S1361841522000883.

Yamil Vindas, Blaise Kévin Guépié, Marilys Almar, Emmanuel Roux, and Philippe Delachartre. An hybrid cnn-transformer model based on multi-feature extraction and attention fusion mechanism for cerebral emboli classification. In *Proceedings of the 7th Machine Learning for Healthcare Conference*, Proceedings of Machine Learning Research. PMLR, 05–06 Aug 2022b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Gaowei Xu, Tianhe Ren, Yu Chen, and Wenliang Che. A one-dimensional cnn-lstm model for epileptic seizure recognition using eeg signal analysis. *Frontiers in Neuroscience*, 14, 2020. ISSN 1662-453X.

Ke Xu, Dezheng Zhang, Jianjing An, Li Liu, Lingzhi Liu, and Dong Wang. Genexp: Multi-objective pruning for deep neural network based on genetic algorithm. *Neurocomputing*, 451:81–94, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.04.022. URL https://www.sciencedirect.com/science/article/pii/S092523122100549X.

Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, and Kurt Keutzer. Hawq-v3: Dyadic neural network quantization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11875–11886. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yao21a.html.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Skh4jRcKQ.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. Ternarybert: Distillation-aware ultra-low bit bert. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. Trained ternary quantization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1_pAu9xl.

Table 4: Distribution of the HITS per class and per subject (subjects 0 to 19). The HITS are classified using three classes: solid embolus, gaseous embolus, and artifact. Some HITS cannot be classified by one expert annotator in one single class, so they are labeled as unknown (these are not used to train or evaluate the classification models). This happens mainly between solid and gaseous emboli, or between small intensity solid emboli and artifacts.

Subject ID	Solid embolus	Gaseous embolus	Artifact	Unknown	Total
0	0	123	15	1	139
1	24	3	1	0	28
2	0	72	0	0	72
3	11	0	46	0	57
4	1	0	0	0	1
5	2	0	0	0	2
6	0	0	48	0	48
7	3	0	0	0	3
8	56	0	0	0	56
9	1	0	54	0	55
10	0	4	0	0	4
11	1	0	0	0	1
12	0	15	0	0	15
13	0	76	0	0	76
14	2	0	0	0	2
15	5	0	46	0	51
16	3	0	0	0	3
17	14	0	4	0	18
18	2	0	0	0	2
19	0	54	0	0	54

Appendix A. Distribution of HITS per class and per subject.

Table 5: Distribution of the HITS per class and per subject (subjects 20 to 38). The HITS are classified using three classes: solid embolus, gaseous embolus, and artifact. Some HITS cannot be classified by one expert annotator in one single class, so they are labeled as unknown (these are not used to train or evaluate the classification models). This happens mainly between solid and gaseous emboli, or between small intensity solid emboli and artifacts.

Subject ID	Solid emboli	Gaseous embolus	Artifact	Unknown	Total
20	0	7	0	0	7
21	20	0	0	0	20
22	0	0	1	0	1
23	17	0	0	0	17
24	1	0	0	0	1
25	1	0	0	0	1
26	1	0	0	0	1
27	45	6	0	0	51
28	268	2	48	0	318
29	42	181	0	3	226
30	0	7	0	0	7
31	24	0	0	0	24
32	7	1	4	0	12
33	0	0	48	0	48
34	0	0	34	0	34
35	17	0	0	0	17
36	1	0	15	0	16
37	0	4	0	0	4
38	0	14	39	0	53

Table 6: HITS population characteristics computed with the available information. F stands for female, M for male, and U for unknown.

Sex	Number	Median Age	Range Age	Mean n° HITS/min
F	19	69	24-85	4,55
M	15	56	21-81	3,98
U	5	74.5	71-78	4,45
All	39	63	21-85	4,32

Table 7: Number of samples per class in the ESR dataset (Andrzejak et al., 2002). The class 1 corresponds to a seizure activity recording. Classes 2-5 corresponds to non seizure activity. In class 2, the sample was obtained from a tumor area in the brain. Class 3 corresponds to a sample coming from a healthy area of the brain. Class 4 corresponds to an EEG sample obtained for a patient that was closing their eyes. Class 5 corresponds to an EEG sample obtained for a patient with its eyes openend.

Class	Number of samples
1	
2	
3	2300
4	
5	

Appendix B. Distribution of ESR samples

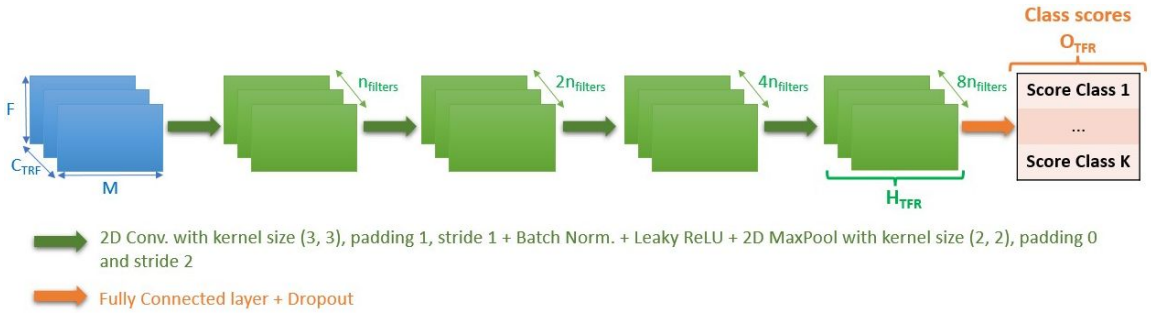

Figure 4: 2D CNN architecture used for the HITS and ESR dataset, taking as input a time-frequency representation of the raw signal. Image from (Vindas et al., 2022b).

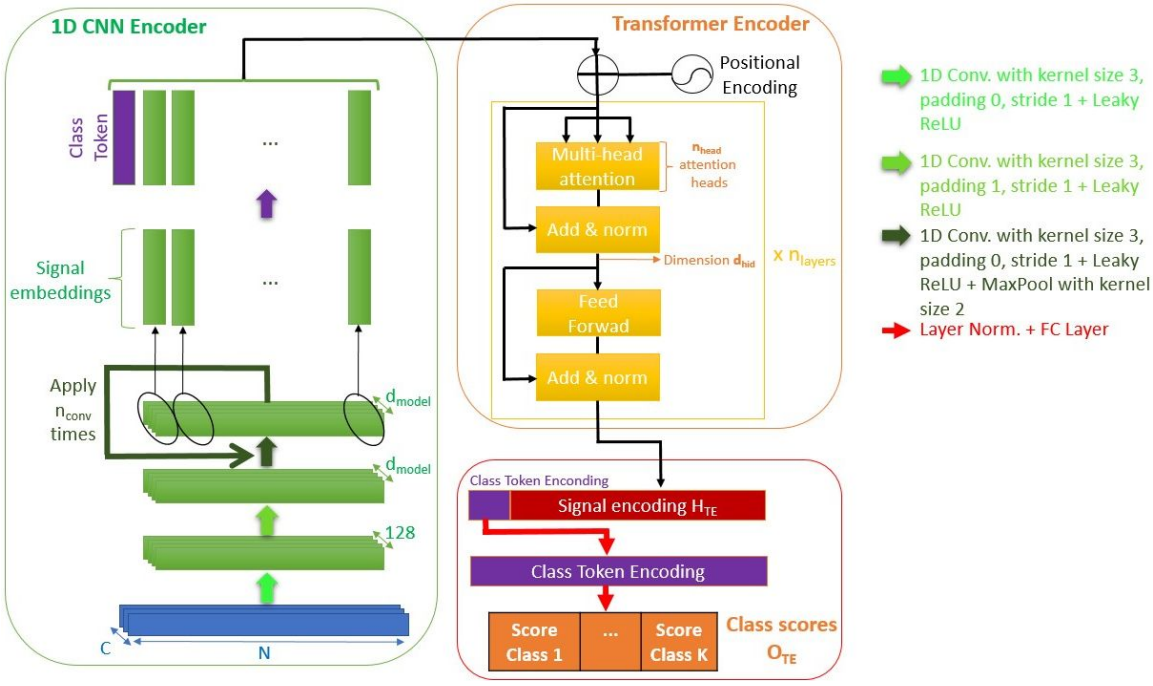

Figure 5: 1D CNN-transformer architecture used for the HITS and ESR dataset, taking as the normalized raw signal. Image from (Vindas et al., 2022b).

Appendix C. Used network architectures details