Applying maximum entropy principle on quantized neural networks correlates with high accuracy

Lucas Maisonnave
Univ. Paris-Saclay
CEA LIST
F-91120, Palaiseau, France
lucas.maisonnave@cea.fr

Cyril Moineau
Univ. Paris-Saclay
CEA LIST
F-91120, Palaiseau, France
cyril.moineau@cea.fr

Olivier Bichler
Univ. Paris-Saclay
CEA LIST
F-91120, Palaiseau, France
olivier.bichler@cea.fr

Fabrice Rastello
Univ. Grenoble Alpes
Inria, CNRS, Grenoble INP, LIG
38000 Grenoble, France
fabrice.rastello@inria.fr

***Abstract*—Neural network quantization, sparsification or Neural Architecture Search (NAS) have shown great success in reducing model size and computational cost on many different ML tasks and architectures. Appropriately reducing model size without degrading task performance requires a suitable measure for quantifying the amount of important information entailed in the model parameters. Information theory provides such a tool, called entropy, which, from a probabilistic point of view, makes it possible to determine whether a model has effectively extracted important information from the data. Based on information theory and the maximum entropy principle, this paper investigates the influence of the clamping function on the distribution of weights in a quantized neural network. We show that the entropy weights can be increased by using a trainable parameter that evolves during training. We also identify a correlation between high entropy and high performance. Finally, by using a regularizer that enables the model to further increase its entropy we highlighted the importance to split training into a generalization phase and an information optimization phase.**

***Index Terms*—compression, quantization, deep learning, entropy**

I. Introduction

The increase in performance of deep neural networks has been driven by the availability of ever-larger databases and the use of hardware adapted to parallelizable calculations like GPUs. This exponential increase in the size of databases and in the number of operations required during training [1] has given rise to a number of problems: high energy consumption and computation times, and an increase in memory footprint.

Today, the trend in deep learning is to constantly increase the size of models and databases. This trend cannot continue indefinitely, and will eventually comes up against the limits of computer memory and computing capacity. In the past years, such challenges have led to the emergence of neural network compression, which aims to reduce the size of AIs without affecting their performance. A great deal of work has been carried out to reduce the size of neural networks, with 4 main categories emerging: sparsification, quantization, automatic architecture search (NAS) and matrix decomposition. The aim of all these methods is to favor a simple model for a given task, in line with the minimal description length (MDL [6]) principle. It is a very important principle for embedded AI considerations, as it ensures frugality and computational acceleration. The aim is to find the most compact and accurate representation of the data, while maintaining good performance.

Deep Neural Networks can also be studied from an information-theoretic point of view, using Shannon entropy to quantify the amount of important information in a neural network. However, the distribution of weights and activations of many DNNs is not uniform and does not maximize its entropy. There is therefore a difference between the number of bits allocated to encode weights and the amount of information actually present in the distribution. Maximizing entropy to close this gap has been studied and tested in numerous works on DNNs [7] [8] [9] [10] [11] and seems particularly well suited to neural networks quantization.

In this paper we study neural network quantization methods using information theory to apply the maximum entropy principles to achieve the most compact data representation possible and make each bit of information as useful as possible. We show that we can increase weights distribution's entropy using a trainable parameter β in the clamping function. We also show strong correlation between high entropy and high performance especially for low-bit quantization.

II. Related Works and Fondations

A. Quantization

Quantifying parameters of a DNN consists in converting network floating-point 32 bits weights values with a smaller number of bits like 8, 4, or 2 bits (Figure 1) [12]. The main objective is to reduce memory and computing footprint of network while maintaining high performances and favoring faster and less costly integer-only operations.

Quantization methods are numerous [16], [19], [20], and can be used after (post training quantization) [13] or during training (quantization aware training) [14] , be uniform [16] or non-uniform [17]. Other quantization schemes introduce randomness to choose the upper or lower bin to round values [18]. Some methods allocate a different number of bits depending on the model layer [15] (Mixed Precision).

In our experiments we will base our method on Scale Adjusted Training (SAT [19]) which quantize weights using DoReFa [20] and activations with PArameterized Clipping acTivation (PACT [16]). DoReFa introduces a clamping function that bounds weights between $[0, 1]$ (eq. 1). This function uses

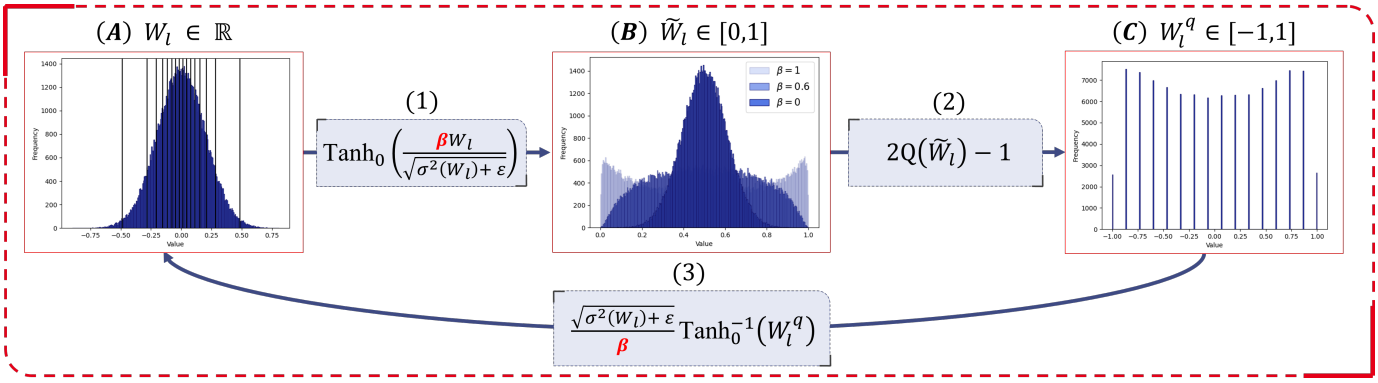


Fig. 1: Quantization pipeline with Tanhβ. (1) First we apply our clamping function to W_l which flatten the distribution depending on β , (2) then we quantize $\tilde{W}_l$ to get our quantized weights used as our knew weights . (3) In our regularization R_H only, we need to "unclamp" our quantized weights to compare them with FP32 W_l.

tanh as a non-linearity to clip outliers before quantization (eq. 2) instead of PACT which uses a trainable parameter as scaling factor to overcome the non symmetric and unbounded activation distributions.

$$\tilde{W}_{l,i} = \mathrm{Tanh}_0(W_{l,i}) = \frac{1}{2}\left(\frac{\tanh(W_{l,i})}{\max_r |\tanh(W_{l,r})|} + 1\right) \quad (1)$$

$$W^q_{l,i} = 2Q(\tilde{W}_{l,i}) - 1 = \frac{2}{(2^b-1)}\mathrm{round}((2^b-1)\tilde{W}_{l,i}) - 1 \quad (2)$$

$W_{l,i}$ being the i^{th} weight of the l^{th} layer and b the bitwidth. SAT also applies constant variance scaling to quantized weights of linear layers without batch normalisation, n_{out} being the number of weights in the layer :

$$W^*_l = \frac{W^q_l}{\sqrt{n_{out}\mathbb{VAR}(W^q_l)}}$$

B. Information theory

Information theory introduces tools to measure information in a set of elements [21] in a probabilistic way by using the $\log_2$ function of the probability distribution of a random variable $X \sim p(x)$ (eq. 3). This way we can add information of two independent random variables.

$$I(X = x) = -\log_2(p(x)) \quad (3)$$

We can now define entropy as the amount of uncertainty in the entire set of discrete values by using the expectation of information across all values:

$$H(X) = \mathbb{E}_{X\sim p(x)}(I(x)) = -\sum_{x\in X} p(x)\log_2(p(x)) \quad (4)$$

Entropy is a fundamental measure of information and has been used in deep learning to quantify information of a dataset filtered by a neural network [24]. Moreover, information theory has been used as an attempt to explain and understand neural network efficiency [22]. In this work entropy and mutual information are used to build a theory linking compression and generalisation of DNNs. Entropy can also be applied to design a regularizer to penalize confident output and maximize uncertainty [23].

This leads to a fundamental principle in information theory : maximum entropy principle (MEP) which states that the distribution that best fits the data is the one with the highest entropy. This principle has been used for quantization in order to better optimize the amount of information allocated for weights and activations [7], [8], [25]. It can be shown that for a Gaussian distribution, maximizing entropy is equivalent to minimizing quantization error [26], which is the $\mathcal{L}_2$ norm between weights and quantized weights :

$$\max_Q H(Q(W)) \iff \min_Q ||W - Q(W)||_2 \quad (5)$$

This equation links a common information measure to a well studied objective in quantized neural networks like in [27] where the quantization error is an objective to minimise to get the best representation possible in 1 bit. In the next sections this equivalence will enable us to develop a regularizer to increase entropy.

III. Method Tanhβ

In the literature DoReFa is a broadly common weights quantization scheme and its clamping function in (eq. 1) is widely used but there is no clear explanation on why tanh clamping would be a more appropriate choice than a naive minmax quantizer (eq. 6).

$$\mathrm{MinMax}(W_l) = \frac{W_l - \min_r(W_{l,r})}{\max_r(W_{l,r}) - \min_r(W_{l,r})} \quad (6)$$

This function is simplier but has the same expected effect : to bound weights values between $[0, 1]$. In fact when we apply MinMax on a MobilenetV1 quantized in 4 bits we can increase Top1 accuracy by 2.14 points and Top5 accuracy by 1.31 points (table II).

One way to explain this gap is to study the amount of information encoded by these different clamping functions. In order to measure the entropy of a neural network we introduce a metric we named H_{norm} which represents the distance

W/A	Model		Baseline FP32	Tanh$_0$	MinMax	Tanhβ
4 bits	MobilenetV1	Top 1 Acc (%)	66.96	67.75	68.25	**68.72**
		H_{norm}	/	0.868	0.681	**0.887**
	MobilenetV2	Top 1 Acc (%)	68.8	66.08	66.38	**67.45**
		H_{norm}	/	0.902	0.752	**0.914**
	Resnet34	Top 1 Acc (%)	67.11	61.71	62.14	**67.85**
		H_{norm}	/	0.789	0.593	**0.865**
3 bits	MobilenetV1	Top 1 Acc (%)	66.96	69.74	69.07	**70.4**
		H_{norm}	/	0.872	0.587	**0.911**
	MobilenetV2	Top 1 Acc (%)	68.8	67.66	**67.74**	67.39
		H_{norm}	/	0.910	0.670	**0.933**
	Resnet34	Top 1 Acc (%)	67.11	64.23	62.09	**66.09**
		H_{norm}	/	0.767	0.481	**0.903**
2 bits	MobilenetV1	Top 1 Acc (%)	66.96	60.52	NaN	**60.64**
		H_{norm}	/	0.829		**0.887**
	MobilenetV2	Top 1 Acc (%)	68.8	**53.56**	NaN	53.49
		H_{norm}	/	0.860		**0.891**
	Resnet34	Top 1 Acc (%)	67.11	**64.68**	5.63	63.71
		H_{norm}	/	0.703	0.127	**0.930**

TABLE I: CIFAR100, Top 1 accuracy and H_{norm} for different models, bitwidth and clamping functions. NaN is for a model that didn't converge. Most of the time our method Tanhβ outperform other functions on accuracy and entropy

Clamping function	Top1 Acc (%)	Top5 Acc (%)	H_{norm}
Tanh	68.93	88.38	0.66
MinMax	71.07	89.69	0.9
Gain	2.14	1.31	0.24

TABLE II: Accuracy and entropy gain for a quantized MobilenetV1 4bits on Imagenet for 150 epoch

between our weight distribution's entropy and the maximum it can reach for each layer (eq. 7).

$$H_{norm} = \frac{1}{N}\sum_{i=1}^{N}\frac{H(W_i)}{b_i} \tag{7}$$

N being the number of layers and b_i the bitwidth of the i^{th} layer. If $H_{norm} = 1$ our model reached its maximum entropy which is the uniform distribution for every layer. The normalisation by b_i allows to compare layers with different bitwidth especially the first and last layers quantized in 8 bits. H is computed with the empirical distribution of every bin of quantization, then we compute the average normalised entropy of the network. In the table II we can see that MinMax also reaches a higher entropy than Tanh (+0.24). The clamping function seems to have more effect on the distribution and the model performances than just bounding weights values.

$$\tilde{W}_l = \text{Tanh}_0\left(\frac{\beta_l W_l}{\sqrt{\sigma^2(W_l)+\epsilon}}\right) \tag{8}$$

To study the effect of the clamping function on the model we introduce Tanhβ which allows the model to modify the entropy of its weights using a trainable parameter β in the clamping function for each layer (eq. 8), σ^2 being the empirical variance of weights. Indeed, weights values are often very small due to weight decay (figure 1.A) so $\tanh$ is almost equivalent to the identity function and doesn't change weight's distribution and the entropy. So the variance rescaling is necessary to escape this linearity of $\tanh$ for small values.

Figure 1.B shows the effect of β on a reduced gaussian distribution ($\mu = 0$). We can see that β affects the distribution's shape and tends to be uniform for $\beta = 0.86$. We can show that β can make the clamping function transition from Tanh$_0$ to MinMax when it varies from 0 to 1 (eq. 9, 10).

$$\lim_{\beta\to 0}\text{Tanh}_0(\beta W) \propto \text{MinMax}(W) \tag{9}$$

$$\lim_{\beta\to 1}\text{Tanh}_0(\beta W) = \text{Tanh}_0(W) \tag{10}$$

By introducing the parameter β we expect the model to use it during training to flatten its weight's distribution through gradient descent. To help the model to optimize β in this way we can use a regularizer based on (eq. 5) to minimise the quantization error and thus maximise entropy :

$$R_H(W) = \frac{1}{N}\sum_{w\in W}(w - Q(w))^2 \tag{11}$$

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda R_H \tag{12}$$

In (eq. 11) we need to compare FP32 weights with quantized weights which usually have values in $[-1, 1]$ (figure 1.C) and can't be compared to non-quantized weights which values are smaller. So we need to "unclamp" quantized bins to rescale values in the set of FP32 weights (black lines of W_l in figure 1.A) and apply the regularizer R_H to increase entropy of W^q. Then we add this regularizer in the loss function with $\lambda = 10^5$ (eq. 12) to make its value significant on Mobilenet models ($\simeq 10^{-1}$, tested on one batch of Imagenet). In the following sections, we will first test the method without regularization, then add it to verify its effect on entropy.

Finally we make the hypothesis that :

- Tanhβ clamping can effectively increase entropy of a quantized model.
- Our regularizer R_H helps increasing entropy.
- Higher entropy correlates in general to higher task performance.

W/A	Model		Baseline FP32	Tanh$_0$	MinMax	Tanhβ
4 bits	MobilenetV1	Top 1 Acc (%)	71.83	68.93	**71.07**	69.65
		H_{norm}	/	0.66	**0.9**	0.867
	MobilenetV2	Top 1 Acc (%)	71.74	66.62	67.11	**67.72**
		H_{norm}	/	0.815	0.756	**0.908**
3 bits	MobilenetV1	Top 1 Acc (%)	71.83	**67.95**	57.96	65.71
		H_{norm}	/	**0.908**	0.447	0.897
	MobilenetV2	Top 1 Acc (%)	71.74	**65.47**	56.63	63.21
		H_{norm}	/	**0.925**	0.47	0.919
2 bits	MobilenetV1	Top 1 Acc (%)	71.83	**59.27**	NaN	58.22
		H_{norm}	/	**0.91**		0.9
	MobilenetV2	Top 1 Acc (%)	71.74	**55.15**	31.3	53.69
		H_{norm}	/	**0.941**	0.331	0.939

TABLE III: ImageNet, Top 1 accuracy and H_{norm} for different models, bitwidth and clamping functions. NaN is for a model that didn't converge. We notice a correlation between high entropy and high accuracy

IV. Experiments

In this section we will validate hypothesis stated previously using SAT quantization scheme based on DoReFa and PACT on MobilenetV1, MobilenetV2 and ResNet34. We will compare all 3 clamping functions introduced previously : Tanh$_0$, MinMax, Tanhβ on Imagenet and CIFAR100.

A. Datasets and Implementation Details

Datasets. The experiments are carried out on the ILSVRC12 ImageNet classification dataset. The ImageNet dataset is made of 1.2 million training images, and 50k validation images with 1000 classes. CIFAR100 is a simpler dataset than Imagenet and made of 50k training images and 10k validation images for 100 classes. In our experiments, we use the classic data augmentation method : resize and crop, horizontal flip.

Experimental settings In our experiments models are trained from scratch with no pretrained weights to study in detail the influence of clamping during training. We train our models for 150 epochs for Imagenet and 100 for CIFAR100 with a learning rate 0.05, a weight decay $4e-5$ and a batch size 256 with SGD optimizer. We initialize weights with a uniform distribution and $\forall l, \beta_l = 0.01$ so $\tanh(\beta x) \sim_0 \text{id}(x)$ because we don't want to change entropy at the beginning and let the model choose where to converge by itself during training.

B. Effect of Tanhβ on entropy

First we study the influence of β on the model's entropy for different bitwidth on CIFAR100. Table I shows Top1 accuracy and entropy for different models and bitwidth and we can see that Tanhβ always has the highest entropy and is very close to the maximum (0.933 for MobilenetV2 3 bits). Whereas MinMax has a lower entropy and a non uniform distribution due to its linearity that can't flatten weights. MinMax is also quite unstable and doesn't always converge in 2bits. With these first results we show that β can increase entropy and the model seems to optimize it in this way without regularization on CIFAR100.

Now we analyze the difference of entropy for each layer between a quantization scheme with Tanh$_0$ and Tanhβ in order to visualize the effect of β in the network. Figure 2 shows

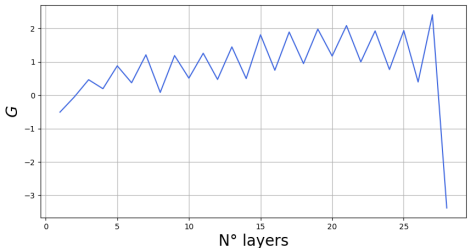


Fig. 2: Entropy gain G of Tanhβ for each layer compared to Tanh$_0$ for MobilenetV1 in 4bits. On average the entropy gain is equal to $G_{mean} = 0.843$ bits

the difference between the entropy of Tanhβ and Tanh$_0$ for every layer and G_{mean} represents the average entropy gain. It appears that for every layer except the first and last one Tanhβ has increased entropy with an averaged gain of 0.848 bits for MobilenetV1 and 0.324 bits for MobilenetV2.

C. Correlation between entropy and accuracy

Now we will study the link between high entropy and high accuracy of a model. First we note that most of the time Tanhβ on CIFAR100 (Table I) has the best performances and can outperform non quantized model at very low precision (70.4% on MobileNetV1 3 bits). These results show potential to this new clamping function and seems to confirm our hypothesis.

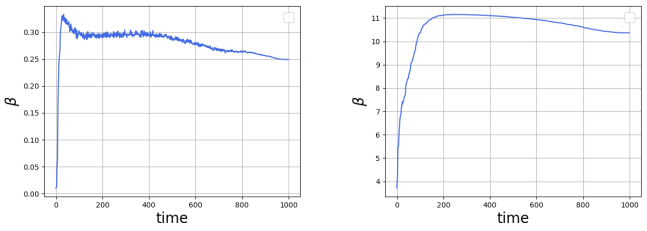
On ImageNet (Table III) this correlation holds for all models and bitwidths. The best model is the one with the highest entropy which is in line with the maximum entropy principle. This trend seems more important for very low bits (3 or 2 bits) of quantization as we can see a huge drop of performance for MinMax (-10% Top1 accuracy in 3 bits) which is also correlated with a drop of entropy (-0.45 on H_{norm}).

To quantify this correlation we compute the Pearson's correlation $\left(\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}\right)$ betweeen Top1 Acc and H_{norm} for CIFAR100 and Imagenet and we get 0.7 and 0.58 respectively.

Now we want to know if the value of entropy is sufficient to explain this correlation so we analyze the evolution of β and H_{norm} during training

D. Evolution of β

The value of β is very important parameter as it defines the entropy of a layer and, the shape of its distribution. We first notice that β is optimized differently on convolutional and linear layers (figure 3). We see that β converge to a very high value for linear layers which represents a distribution almost binary with 2 long bins at -1 and 1. On the figure 2 we see this drop of entropy on the last layer that reaches -4.1 and -5.1 bits of entropy for MobilenetV1 and MobilenetV2 respectively. We interpret that result as over-parametrization of linear layers which need less information to be efficient.



(a) Mean of β_{conv} (b) β_{lin}

Fig. 3: Evolution of β during training for MobilenetV1 4 bits. β_{conv} is the average of β for all convolutional layers and β_{lin} is for the last linear layer.

E. Regularization R_H

As presented in (eq. 11) we can use a regularizer to increase entropy and help the model to optimize β. As we can see in figure 4, minimizing quantization error seems to be a good objective as it increases entropy for a gaussian distribution quantized in 4 bits. But it is not perfect and entropy can decrease for very low quantization error due to approximations made to compute H in 4 bits of quantization.

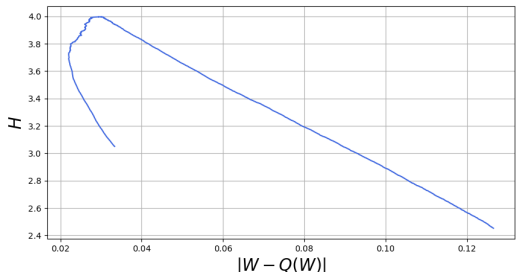


Fig. 4: Entropy over Quantization error for a Gaussian distribution using different values of β. 4 bits being the maximum H can reach

We now test this regularizer on Imagenet with MobilenetV1 and MobilenetV2 quantized in 4 bits (figure 5). We validate that R_H increases entropy compared to Tanhβ without regularizer during training with MBV1 + Tanhβ converging to 0.901 (0.867 without R_H) and MBV2+Tanhβ to 0.932 (0.908 without R_H). Moreover we see that R_H increases entropy at the very beginning of training and seems to remain almost constant.

However Top1 accuracy decreased with this regularizer (around 2%) which means that only looking to the value of entropy is not enough to understand its effect. Maximizing entropy at the beginning could be a part of the issue as seen in [11] they set a threshold to decide when to start maximizing entropy. It's also in line with the results from [22] where they stated there are 2 phases during training :

- **Generalization**. A first short phase where the model tries to fit data and increase accuracy
- **Compression**. A long phase where the model compresses its information

We state that the model first need to fit labels before optimizing its inner information with an entropic regularizer. To go further in this study we could try to set a threshold to the regularizer, use a pre-trained model which already converged and could be easier to compress with entropy constraint, use a scheduler to gradually add the regularizer in the loss.

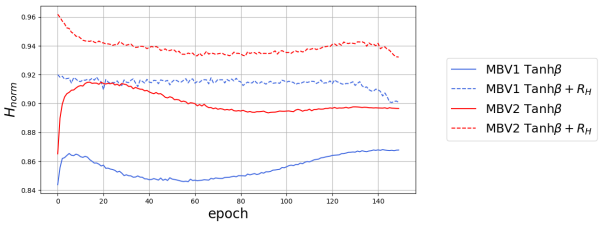


Fig. 5: Entropy over time for a MobilenetV1 and MobilenetV2 in 4 bits with and without R_H

V. Conclusion

In this paper we studied the impact of clamping functions on the entropy of a quantized neural network. We introduced a parameter β in the classic clamping function $\tanh$ in order to study its effect on entropy and accuracy. We found that high entropy is correlated to high accuracy in most of the cases on CIFAR100 and Imagenet. On CIFAR100 the model naturally optimizes β to increase its own entropy but it is the not case on Imagenet and it needs to be regularized. We built a regularizer R_H minimizing quantization error which is equivalent to maximizing entropy for a gaussian distribution and used it during training. We showed that combining Tanhβ with this regularizer allows us to control and increase the entropy of weights. Entropic regularization also showed that only focusing on the value of entropy is not sufficient to explain the correlation with accuracy and that we need to study entropy over time during training.

References

[1] openai.com/blog/ai-and-compute

[2] Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Ashish Vaswani, Noam Shazeer. Attention is all you need. NIPS, 2017.

[3] Andrew M. Dai, Simon Tong, Nan Du, Yanping Huang. Glam : Efficient scaling of language models with mixture-of-experts. ICML, 2021.

[4] Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, Jeff Dean, David Patterson, Joseph Gonzalez. Carbon emissions and large neural network training. arxiv, 2021.

[5] Sara Solla, Yann LeCun, John Denker. Optimal brain damage. NIPS, 1989

[6] A. Barron ; J. Rissanen ; Bin Yu. The minimum description length principle in coding and modeling. IEEE, 1998.

[7] Xin Liu, Zhongdao Wang, Ya-Li Li, Shengjin Wang. Self-supervised learning via maximum entropy coding. NIPS, 2022.

[8] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, Guodong Guo. Q-vit : Accurate and fully quantized low-bit vision transformer. NIPS, 2022.

[9] Messerschmitt D. Quantizing for maximum output entropy. IEEE, 1971

[10] Flemming Topsøe Peter Harremoës. Maximum entropy fundamentals. MDPI, 2001

[11] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, Geoffrey Hinton Regularizing neural networks by penalizing confident output distributions. ICLR, 2017.

[12] Lei Deng; Guoqi Li; Song Han; Luping Shi; Yuan Xie. Model compression and hardware acceleration for neural networks : A comprehensive survey. IEEE, 2020

[13] Ron Banner, Yury Nahshan, Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. NIPS, 2019.

[14] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, Dharmendra S. Modha. Learned step size quantization. ICLR, 2020.

[15] Qing Jin, Linjie Yang. Fracbits : Mixed precision quantization via fractional bitwidths. AAAI, 2021

[16] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085, 2018.

[17] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen, Incremental network quantization: Toward lossless CNNs with low-precision weights, 2017, ICLR

[18] Jianfei Chen, Yu Gai, Zhewei Yao, Michael W Mahoney, and Joseph E Gonzalez. A statistical framework for low-bitwidth training of deep neural networks. arXiv preprint arXiv:2010.14298, 2020.

[19] Qing Jin, Linjie Yang, Zhenyu Liao, Towards Efficient Training for Neural Network Quantization, arXiv:1912.10207, 2019

[20] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, Yuheng Zou, DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradient, arXiv:1606.06160, 2016

[21] Thomas M.Cover, Joy A.Thomas, Elements of information theory

[22] Ravid Schwartz-Ziv, Naftali Tishby, Opening the black box of Deep Neural Networks via Information, 2017

[23] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, Geoffrey Hinton, Regularizing Neural Networks by Penalizing Confident Output Distributions, 2017

[24] Luke Nicholas Darlow, Amos Storkey, What Information Does a ResNet Compress?, ICLR, 2019

[25] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric Xing, Zhiqiang Shen, Nonuniform-to-Uniform Quantization: Towards Accurate Quantization via Generalized Straight-Through Estimation, 2022

[26] Messerschmitt, Quantizing for maximum output entropy (Corresp.), 1971

[27] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi . Xnor-net : Imagenet classification using binary convolutional neural networks. ECCV, 2016.