



Predicting cnn learning accuracy using chaos measurement

Rémi Piau, Thomas Maugey, Aline Roumy

► To cite this version:

Rémi Piau, Thomas Maugey, Aline Roumy. Predicting cnn learning accuracy using chaos measurement. ICASSP 2023 - IEEE International Conference on Acoustics, Speech, and Signal Processing, Jun 2023, Rhodes, Greece. pp.1-4. hal-04133669

HAL Id: hal-04133669

<https://hal.science/hal-04133669>

Submitted on 20 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

PREDICTING CNN LEARNING ACCURACY USING CHAOS MEASUREMENT

Rémi Piau, Thomas Maugey, Aline Roumy

INRIA, Rennes, France
name.surname@inria.fr

ABSTRACT

Learning on entropy coded data has many benefits. First, it avoids decoding, but also it allows one to process compact data. However, this type of learning has been overlooked due to the chaos introduced by entropy coding functions. Indeed, convolution widely used in learning algorithm requires that the encoding function preserves the distance between pixel positions (spatial closeness), and the distance between pixel values (semantic closeness). Even if these two properties are not satisfied by entropy coding, we had shown previously that learning on entropy coded data is possible and that the accuracy depends on spatial and semantic closeness. In this paper, we quantify this dependence, and introduce a new metric, that measures the chaos in the data representation. This measure is easy to compute as it depends on the encoded data only. Moreover, this measure allows one to predict the accuracy of the learning algorithm, that process entropy coded data.

Index Terms— Image and Video Coding for Machine, Deep-learning, CNN, Entropy coding.

1. INTRODUCTION

Due to their exploding amount, the visual data created everyday are mostly meant to be processed by machines rather than only be watched by users [1]. A first consequence is that, in this new paradigm, one has to revisit image and video compression, as for instance in [2, 3]. Another interesting consequence is that we may wonder if decoding is really necessary before machine’s processing. Indeed, when a user watches an image or a video, the decoding operation, *i.e.*, the mapping between the compressed and spatial domains, is obviously inevitable. As a machine does not have our visual priors, we may wonder if one can train it to analyze the visual data in the compressed domain directly. In doing so, we could avoid decoding and save on storage, since one does not need to keep decoded data while the analyzer could process such data in a more compact form. Some works have tackle this question by showing that deep learning algorithms are able to learn in the compressed domain [2–7]. However, the compressed domain they all consider does not include the entropy coding step, whereas this step is crucial and systematically included in image/video coders to efficiently compact the information.

The reason why the question of *learning on the entropy coded domain* has been overlooked in the literature is simple. Contrary to transform or quantization operations, the entropy coding introduces a complete disorganization in the data representation that is hardly compatible with the learning architecture. Indeed, in order to work properly, the convolution operation (that is massively deployed

in deep neural architecture such as convolutional Neural Networks (CNN)) needs that the encoding function preserves the distance between the pixel positions and between the pixel values. We refer to these two properties as *spatial and semantic closeness*, which we define more formally in the rest of the paper. These properties are however not satisfied in the case of an entropy coder. Indeed, the data representation is only based on the probability distribution, so two close pixel values can be assigned to two drastically different coded values. Moreover, entropy coder have variable length output. Therefore, pixels with close position, have after decoding distance, which is variable and depends on the value.

In [8], we have shown that, to our surprise, learning in the entropy coded domain is *still possible* with convolutional Neural Network (CNN). Still, there is a drop in performance, and we have shown a link between the learning performance and the two data properties we introduced: spatial and semantic closeness. Finally, we have shown that arithmetic entropy coding [9] leads to lower performance than Huffman entropy coding [10] without having a quantitative explanation for this.

To tackle this lack of explanation, we propose, in this work, a metric to measure the impact of an entropy coding function on the learning performance. More precisely, we propose to measure the level of *chaos* introduced by an encoding function f . First, the rationale behind this chaos measurement is related to the closeness assumptions that the encoding function must satisfy, for a convolution based learning algorithm to perform well. Moreover, we show that this chaos measurement is directly linked to the learning accuracy when done in the compressed domain. Finally, we show that the proposed metric is able to predict the learning performance for a given entropy coding algorithm. For instance, for the set of Huffman coding algorithm parameterized by the length of the input vector, the learning accuracy is a monotonic function of the proposed chaos measure.

2. LEARNING IN THE ENTROPY CODED DOMAIN

In this paper we use the same theoretical and practical framework as in [8]. For the sake of completeness, we review this framework in this section.

2.1. In theory

The usual way to learn on coded data is to decode them before learning: this is the (a) *Classical* learning pipeline of Figure 2. In this figure, X denotes respectively the original data and Y the label. In this paper instead, we consider the scheme learning (b) *in the coded domain* of Figure 2, where we bypass the decoding step (decoder g) and directly learn on the encoded data. In both schemes, the encoder/decoder pair (f, g) is fixed in advance and can not be optimized for better learning. This corresponds to the use case of signif-

This work was supported by the Cominlabs excellence laboratory with the French National Research Agency’s funding (ANR-10-LABX-07-01).

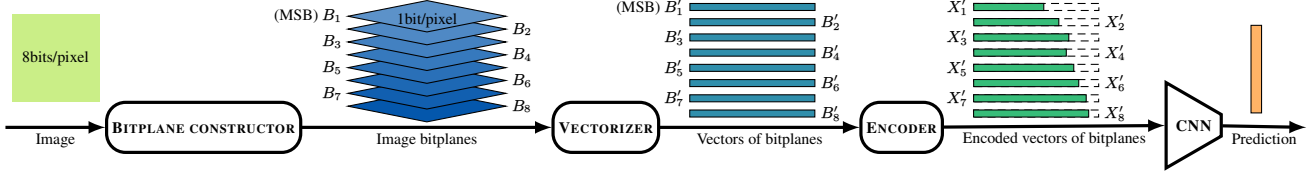


Fig. 1. Experimental pipeline for learning on coded data (figure from [8]).

icant practical interest, where data have been acquired by a classical image/video codecs, and where the learning task is not known upon acquisition.

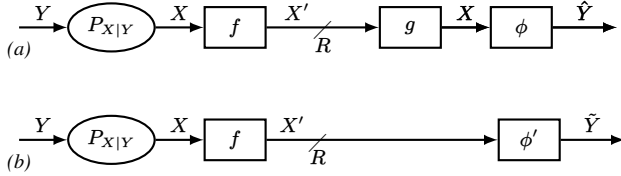


Fig. 2. Classical learning (a) vs learning in the coded domain (b) (figure from [8]).

More formally, given a training set T_{train} composed of pairs image/label (X, Y) and a pairwise distance D , *learning in the entropy coded domain* looks for an estimator ϕ' , and solves (1):

$$\min_{\phi'} \sum_{(X, Y) \in T_{\text{train}}} D(\phi' \circ f(X), Y) \quad (1)$$

where, in our case, the encoder/decoder pair (f, g) is respectively composed of the entropy encoder and its associated decoder.

On one hand, entropy coding is lossless. Therefore, we do not lose any information between X and $f(X)$. As such, our theoretical learning accuracy should not be lower when learning on the coded data $f(X)$ than when learning on non-coded data X . But, on the other hand, this statement does not take into account the characteristics of a practical estimator: in our case CNN. This type of networks seeks pattern in the input data they train on, but entropy coding disturbs these patterns by shifting data distribution toward an independent and uniform one. Learning in case (b) is thus harder than in case (a), but is still possible as shown by experiments from [8].

2.2. In practice

In our experiments, we reuse the same pipeline as in [8]. Following Figure 1, we learn coded data by first taking a grayscale 8 bits/pixel image and splitting it into bitplanes. Then, we run an entropy coder on each bitplane and pad each coded bitplane to a fixed size to finally train the CNN on this transformed data. Each coded and padded bitplane vector is passed to the CNN as a channel.

Table 1 shows the learning accuracy of **UVGG11** (Unidimensional VGG-11), a CNN specially designed for one-dimensional binary input data. Results are reported for two entropy coders (arithmetic and Huffman), where the input symbol size is 8 bits for each. Two data sets are also considered.

We observe that learning on entropy coded data leads to a loss in accuracy, and conjecture that this loss is related to the amount of perturbation introduced in the original (uncoded) data. To verify this conjecture, we first review in the next section the qualitative characteristics a sequence data must satisfy for a CNN to perform

Dataset	Coding Type		
	None	Huffman	Arithmetic
MNIST	0.98911	0.83234	0.63130
YCIFAR-10	0.56573	0.36062	0.29762

Table 1: UVGG11 Network accuracy on entropy coded data (data from [8]).

well. More importantly, we introduce a new quantitative measure of chaos in order to be able to predict the learning performance, when learning is performed on entropy coded data.

3. CHAOS MEASUREMENT

3.1. Closeness properties

For the sake of completeness, we review the two following closeness properties, introduce in our previous paper [8], that data must satisfy for a CNN to work well:

- (i) *Spatial closeness*: two neighboring pixels on the pixel grid should, in the representation, be placed at adjacent and identifiable positions,
- (ii) *Semantic closeness*: neighbor data cell values (pixel values) that are close in terms of meaning (e.g., close in terms of perceived physical color) should be close in terms of representation (integer color value).

First note that these properties are in practice linked. Indeed, changes in semantic closeness can induce changes in spatial closeness and vice-versa. Moreover and more importantly, both properties have similar effects on the entropy coded data: whatever the property, when it is not satisfied, patterns are difficult to identify. So, when observing the entropy coded data, these properties are not distinguishable. Therefore, in the following subsection, we introduce a new metric that measure the perturbations of both properties at once, allowing for accuracy loss prediction.

3.2. Proposed metric

Measuring the accuracy loss caused by entropy coding on a given CNN model, requires heavy simulation. Indeed, for each entropy coding function f , training (1) has to be performed. This can be cumbersome, especially when the parameter space to be tested for a given entropy coding function is large. An example of such parameter is the vector length at the input of the entropy coder function. By introducing a chaos measurement metric, we can avoid costly re-training, and rely on this metric to gauge the accuracy degradation induced by the encoding function.

To introduce our metric, we consider 2 binary vectors:

- X the original data,
- $Y = f(X)$ the entropy coded data.

Our CNN takes bit-planes as input (but it works the same when taking non-binary symbols). There is b bit-planes where b is the size in bits of the number represented in binary vector X . We number the bit-planes such as bit-plane 0 corresponds to the most-significant-bit (MSB) plane and bit-plane $b-1$ corresponds to the least-significant-bit (LSB) plane. We note $B_i: \{0, 1\}^* \rightarrow \{0, 1\}^*$ the function extracting the bit plane i ($i \in \{0, \dots, b-1\}$) from a sequence of bits. We also note $\text{ApEn}_{m,r}: \{0, 1\}^* \rightarrow \mathbb{N}$ the function computing the Approximate Entropy [11] on a sequence of bits. This Approximate Entropy function is compute for a given window size m and a given threshold r .

Given a function f (our coder) and an input X , we propose to measure the chaos introduced in the sequence by $\mathcal{D}_f(X)$:

$$\mathcal{D}_f(X) = \frac{\sum_{i=0}^B \text{ApEn}_{m,r}(f(B_i(X)))}{\sum_{i=0}^B \text{ApEn}_{m,r}(B_i(X))} \quad (2)$$

In this metric we compute the sum of the Approximate Entropy of each bitplane for the coded data. Intuitively, when the chaos of the sequence increase, this same sequence becomes more difficult to predict, thus the approximate entropy increases. It means the sequence nears a random one and as such it becomes more difficult to find patterns to use for classification.

Secondly, the number of patterns can be linked to a specific input sequence X that contains less information before coding. Thus, we have to compare the sum of the bitplanes approximate entropy for the coded data to the one for the non-coded data. Finally, to ensure stability the final measure is an average over the dataset.

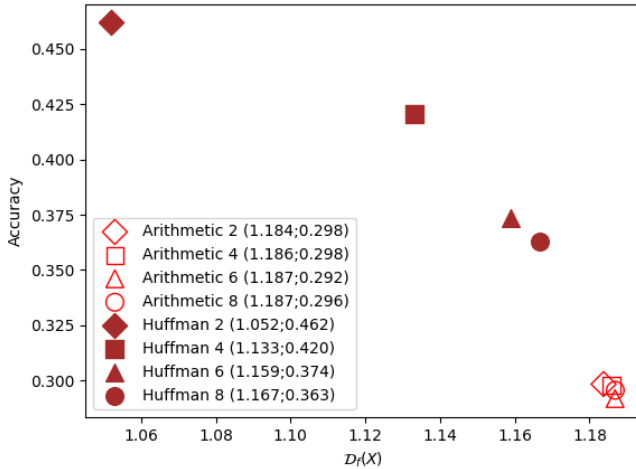


Fig. 3. Accuracy of VGG11 on YCIFAR-10 in function of $\mathcal{D}_f(X)$ for several transform f (Huffman and arithmetic coding with input symbol size 2, 4, 6, 8)

4. EXPERIMENTS

4.1. Setup

We base our experimental setup on [8] as described in Subsection 2.2. For a given entropy coder, we train a CNN on the encoded data that we pad if needed. The evaluation of the trained CNN gives us its accuracy. On the other-hand, we compute our metric \mathcal{D}_f .

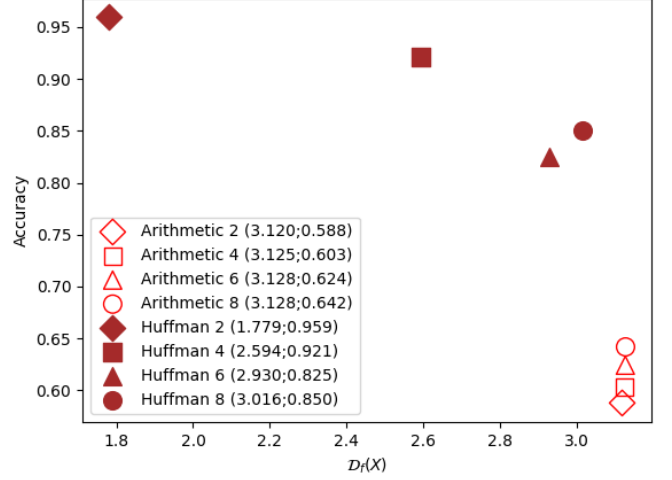


Fig. 4. Accuracy of VGG11 on MNIST in function of $\mathcal{D}_f(X)$ for several transform f (Huffman and arithmetic coding with input symbol size 2, 4, 6, 8)

We then search for a correlation between the accuracy loss and our metric.

4.2. Results

In this experiment we use **UVGG11** [8] (Unidimensional VGG11 [12]) as the CNN. We train the network using stochastic gradient descent (ADAM [13]) with a learning rate of 10^{-4} . This give us the respective accuracy for several transforms: Huffman coding [10] and arithmetic coding [9] with sizes of 2,4,6,8.

Figure 3 plots the accuracy evaluated on the trained CNN versus the proposed chaos measurement metric, for the YCIFAR-10 dataset [8] (the grayscale version of CIFAR-10 [14]), a dataset containing realistic images with 10 classes. For Huffman entropy coding, the input length varies from 2 to 8, and we observe a good correlation between the true accuracy and the proposed metric. For arithmetic coding, the tested entropy coders with their various input symbol sizes have little influence on the chaos metric and the accuracy. The introduced chaos is greater with arithmetic coding. Thus, the accuracy is worse than Huffman. At the same time the metric is also greater than Huffman, sign of a good correlation. The accuracy of Figure 4 draws the same features, but for a different dataset: MNIST [15]. Here, the correlation between the proposed metric and the accuracy is also good.

5. CONCLUSION

In this paper, we tackled the difficult problem of learning on entropy coded data. In particular, we introduced a new chaos measurement metric in order to quantitatively measure the effect of an entropy coder on the learning performance. We find interesting that simulations show a good correlation between the proposed metric and the accuracy of the learning algorithm.

6. REFERENCES

- [1] Lingyu Duan, Jiaying Liu, Wenhan Yang, Tiejun Huang, and Wen Gao, "Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics," *IEEE*

Transactions on Image Processing, vol. 29, pp. 8680–8695, 2020.

- [2] Robert Torfason, Fabian Mentzer, Eiríkur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool, “Towards Image Understanding from Deep Compression Without Decoding,” in *International Conference on Learning Representations*, Feb. 2018.
- [3] Lahiru D. Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman, “End-to-end optimized image compression for multiple machine tasks,” *arXiv:2103.04178 [cs]*, Mar. 2021.
- [4] David Edmundson and Gerald Schaefer, “An overview and evaluation of JPEG compressed domain retrieval techniques,” in *Proceedings ELMAR-2012*, Sept. 2012, pp. 75–78.
- [5] Yuming Fang, Weisi Lin, Zhenzhong Chen, Chia-Ming Tsai, and Chia-Wen Lin, “A Video Saliency Detection Model in Compressed Domain,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 27–38, Jan. 2014.
- [6] Aaron Chadha, Alhabib Abbas, and Yiannis Andreopoulos, “Video Classification With CNNs: Using the Codec as a Spatio-Temporal Activity Sensor,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 475–485, Feb. 2019.
- [7] Junyi Feng, Songyuan Li, Xi Li, Fei Wu, Qi Tian, Ming-Hsuan Yang, and Haibin Ling, “TapLab: A Fast Framework for Semantic Video Segmentation Tapping Into Compressed-Domain Knowledge,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1591–1603, Mar. 2022.
- [8] Rémi Piau, Thomas Maugey, and Aline Roumy, “Learning on entropy coded images with cnn,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, 2023.
- [9] Khalid Sayood, “Chapter 4 - Arithmetic Coding,” in *Introduction to Data Compression (Fifth Edition)*, Khalid Sayood, Ed., The Morgan Kaufmann Series in Multimedia Information and Systems, pp. 89–130. Morgan Kaufmann, Jan. 2018.
- [10] Khalid Sayood, “Chapter 3 - Huffman Coding,” in *Introduction to Data Compression (Fifth Edition)*, Khalid Sayood, Ed., The Morgan Kaufmann Series in Multimedia Information and Systems, pp. 41–88. Morgan Kaufmann, Jan. 2018.
- [11] S M Pincus, “Approximate entropy as a measure of system complexity,” *Proc Natl Acad Sci U S A*, vol. 88, no. 6, pp. 2297–2301, Mar. 1991.
- [12] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.
- [13] Diederik P. Kingma and Jimmy Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations*, 2014.
- [14] Alex Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” Tech. Rep., University of Toronto, Apr. 2009.
- [15] Yann LeCun, “THE MNIST DATABASE of handwritten digits,” <https://yann.lecun.com/exdb/mnist/>.