

# Prédiction de la précision d'apprentissage des réseaux de neurones convolutifs par mesure du chaos

Rémi PIAU   Thomas MAUGEY   Aline ROUMY  
INRIA, Rennes, France

**Résumé** – Apprendre sur les données codées avec un codeur entropique présente de nombreux avantages : cela permet d'éviter de décoder tout en ayant des données de taille plus faible à traiter. Pourtant cette manière d'apprendre n'a pas été explorée à cause du chaos introduit par les codeurs entropiques. En effet, la pièce centrale des algorithmes d'apprentissage profond, la convolution a besoin que la fonction d'encodage préserve la distance entre les pixels (proximité spatiale), ainsi que la distance entre les valeurs de ces mêmes pixels (proximité sémantique). Même si ces deux propriétés ne sont pas satisfaites par les codeurs entropiques, nous avons montré précédemment que l'apprentissage sur des données codées par un codeur entropique est possible et que la précision de cet apprentissage dépend du respect de ces proximités. Dans ce papier nous mesurons cette dépendance et introduisons une nouvelle métrique qui mesure le chaos dans une représentation de données. Cette mesure est facile à calculer, car elle ne dépend que des données codées. Cette dernière nous permet de prédire la précision de notre apprentissage sur les données codées.

**Abstract** – Learning on entropy coded data has many benefits. First, it avoids decoding, but also it allows 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, for a family of entropy coders, this measure allows to predict the accuracy of the learning algorithm, that process entropy coded data.

## 1 Introduction

De par l'augmentation ininterrompue de la quantité de données visuelles produite chaque seconde la majorité de ces dernières sont principalement traitées par des machines plutôt que regardées par des humains [3]. Ce changement de paradigme ouvre le champ des possibles pour de nouvelles formes de compression d'image et de vidéo (comme dans [2, 14]).

De fait, nous pouvons nous demander si le décodage des données est vraiment nécessaire avant le traitement par une machine. Lorsque qu'une image ou vidéo est regardée par un humain, le décodage est nécessaire pour pouvoir l'afficher à l'écran de manière intelligible. À l'inverse, pour une machine, nous pouvons envisager d'apprendre directement dans le domaine du codé. Ce faisant, la quantité de donnée à analyser se trouverait réduite. Plusieurs auteurs ont proposé des solutions à ce problème en proposant des algorithmes d'apprentissage profond sur les données codées [1, 2, 4–6, 14]. Pourtant le domaine compressé considéré n'inclut jamais l'étape de codage entropique alors que cette dernière fait partie intégrante de tout les codeurs d'images et de vidéos. Ce codage est en effet primordial pour réduire la taille du flux codé.

Une des principales raisons qui font que l'apprentissage sur les données codées par un codeur entropique n'a pas été exploré est la désorganisation de la représentation que cette transformation introduit. Or, pour fonctionner correctement, l'opération de convolution, pièce centrale des réseaux de neurones convolutifs (CNN en anglais), a besoin que la fonction

de codage préserve la distance entre la position des pixels ainsi qu'entre leurs valeurs. Nous parlons de ces propriétés, respectivement, comme la *proximité spatiale* et la *proximité sémantique*. Lors du codage entropique, ces propriétés sont perdues, car ce type de codage basé sur la fréquence des symboles à une longueur variable (perturbation spatiale) et peut attribuer à des symboles de valeurs proches des représentations très différentes (perturbation sémantique).

Dans notre papier [10], nous avons montré que l'apprentissage sur des données codées entropiquement est toujours possible en utilisant des réseaux de neurones convolutifs même si cela implique une baisse de performance liée à la perte des propriétés de proximité. Nous avons aussi montré que le codage arithmétique [12] entraîne de plus grande perte de performance que le codage de Huffman [12] sans avoir une explication quantitative.

C'est pourquoi nous proposons ici une métrique mesurant l'effet d'un codage entropique sur la performance de l'apprentissage. Plus précisément nous proposons de mesurer le niveau de chaos introduit par une fonction de codage  $f$ . Dans un premier temps nous expliquons comment la formulation de cette mesure est liée aux critères de proximité nécessaire au bon fonctionnement des réseaux de neurones convolutifs. Par la suite nous montrons le lien entre notre métrique et la précision d'apprentissage. Enfin, nous montrons que notre métrique peut grossièrement prédire les performances d'apprentissage d'une famille de codeurs entropiques.

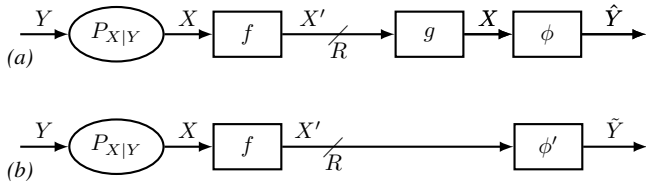


FIGURE 1 : Apprentissage usuel (a) vs apprentissage dans le domaine du codé (b) (Figure traduite de [10]).

## 2 Apprendre dans le domaine des données codées entropiquement

Dans ce papier nous utilisons le même cadre théorique et pratique que dans [10]. Par souci d'exhaustivité, nous rappelons ses grandes lignes dans cette partie.

### 2.1 En théorie

L'apprentissage est habituellement réalisé après le décodage des données : c'est l'*apprentissage classique* (a) de la Figure 1. Dans cet article, nous utilisons la deuxième méthode : *apprentissage dans le domaine du codé* (b) de la même figure. Dans cette méthode nous enlevons l'étape de décodage (décodeur  $g$ ) pour apprendre directement sur les données codées. Dans ces deux méthodes, la paire encodeur/décodeur ( $f, g$ ) est fixée à l'avance et ne peut pas être optimisée pour la tâche d'apprentissage. Cela correspond au cas pratique d'utilisation d'un codec image/vidéo classique lors de l'acquisition où la tâche d'apprentissage n'est alors pas encore spécifiée.

Plus formellement, étant donné un jeu de données d'entraînement  $T_{\text{train}}$  composé de paires image/label  $(X, Y)$  et une distance  $D$ , l'*apprentissage sur le domaine des données codées entropiquement* revient à optimiser un estimateur  $\phi'$  suivant l'équation (1) :

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

où dans notre cas la paire encodeur/décodeur ( $f, g$ ) est respectivement composée du codeur entropique et de son décodeur associé.

D'un côté, le codage entropique est sans perte, nous ne perdons donc aucune information lors du passage de  $X$  à  $f(X)$ . De fait, notre précision d'apprentissage théoriquement atteignable sur les données codées  $f(X)$  devrait être identique à notre précision d'apprentissage sur les données non codées  $X$ . D'un autre côté, cela ne prend pas en compte les caractéristiques pratiques de notre estimateur, ici, les réseaux de neurones convolutifs. Ce type de réseau cherche des motifs dans les données d'entrée. Or, le codage entropique perturbe ces motifs en modifiant la distribution des données vers une distribution indépendante et uniforme rendant l'apprentissage plus difficile.

### 2.2 En pratique

Dans nos expériences, nous réutilisons le même protocole que dans [10]. D'après la Figure 2, nous apprenons sur les données codées en plusieurs étapes. Tout d'abord notre image en

niveaux de gris (8bit/pixel) est divisée en plan de bits. Ensuite nous appliquons un codeur entropique à chacun de ces plans de bit puis complétons les vecteurs obtenus pour atteindre une taille fixe. Ces données sont ensuite passées à l'entrée de notre réseau de neurone (chaque plan de bit est passé comme un canal à l'entrée du réseau).

Le tableau 1 montre la précision d'apprentissage de **UVGG11** (Unidimensional VGG-11), un réseau conçu pour recevoir comme entrée des données binaires et unidimensionnelles. Les deux codeurs entropiques utilisés sont réglés sur une taille de symbole d'entrée de 8 bits.

TABLE 1 : Précision du réseau UVGG11 sur des données codées entropiquement (données de [10]).

Jeu de données	Type de Codeur		
	Aucun	Huffman	Arithmétique
MNIST	0.98911	0.83234	0.63130
YCIFAR-10	0.56573	0.36062	0.29762

Nous observons qu'apprendre sur des données codées entropiquement entraîne une perte de précision. Nous faisons la conjecture que cette perte est en relation directe avec la quantité de perturbation introduite dans les données initiales (non codées). Dans un premier temps, pour vérifier cette conjecture, nous rappelons les caractéristiques quantitatives qu'une séquence de données doit satisfaire pour qu'un réseau de neurones convolutifs fonctionne bien. Partant de ce constant nous introduisons ensuite une nouvelle mesure quantitative : la mesure du chaos introduit. Cette dernière nous permet de prédire la performance d'apprentissage quand celui-ci est réalisé sur les données codées entropiquement.

## 3 Mesure du Chaos

### 3.1 Propriétés de proximité

Par souci d'exhaustivité, nous rappelons les deux propriétés de proximité, introduite dans notre précédent papier [10], que les données doivent satisfaire pour qu'un réseau de neurones convolutifs fonctionne correctement :

- (i) *Proximité spatiale* : deux pixels adjacents sur la grille doivent, dans leur représentation, être placés à des positions adjacentes et identifiables.
- (ii) *Proximité sémantique* : les données du voisinage d'un pixel (valeurs des pixels) qui sont proche en termes de signification (e.g., proche en termes de couleur perçue physiquement) doivent être proche dans leur représentation (valeur numérique de la couleur).

Tout d'abord, il nous faut noter que ces deux propriétés sont liées en pratique. En effet les changements dans la proximité sémantique peuvent introduire des changements dans la proximité spatiale et vice-versa. De plus, les pertes de chacune de ces deux propriétés ont un effet similaire sur les données codées entropiquement : ces propriétés ne sont pas distinguables. C'est pourquoi dans la prochaine sous-partie, nous proposons

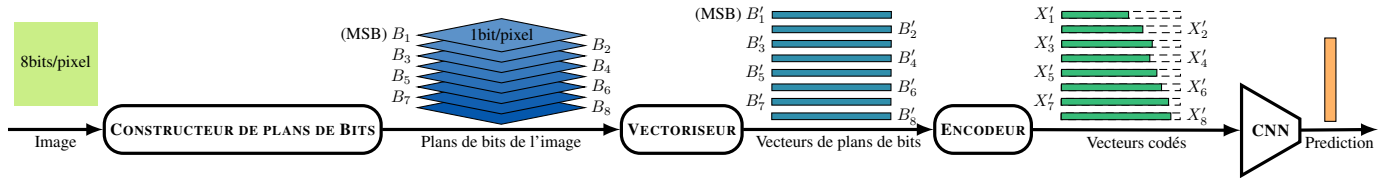


FIGURE 2 : Déroulement de l'expérience d'apprentissage sur les données codées (Figure traduite de [10]).

une métrique qui mesure les perturbations des deux propriétés en même temps ; nous permettant de prédire la perte de précision.

### 3.2 Mesure proposée

Mesurer la perte de précision induite par le codage entropique sur un modèle de réseau de neurones convolutifs donné demande de nombreux calculs. En effet, pour chaque fonction de codage  $f$ , l'entraînement du réseau (1) doit être réalisé. De plus, lorsque l'espace des paramètres à tester pour un codeur entropique donné est grand, le nombre d'entraînements augmente encore. En introduisant cette métrique de mesure, nous pouvons éviter les réentraînements coûteux en utilisant cette dernière pour estimer la perte de précision introduite par la fonction de codage.

Pour calculer cette mesure nous avons besoin des deux vecteurs binaires suivant :

- $X$  les données originales,
- $Y = f(X)$  les données codées par un codeur entropique.

Notre CNN prend des plans de bits en entrée (mais cela fonctionne de la même manière si les symboles ne sont pas binaires). Il y a  $b$  plans de bits, où  $b$  est la taille en bit des nombres représentés dans le vecteur binaire  $X$ . Nous numérotions ces plans de bits avec 0 le plan de bit de poids forts (Most Significant Bits en anglais) jusqu'à  $b - 1$  le plan de bits de poids faibles (Least Significant Bits en anglais). Nous notons  $B_i : \{0, 1\}^* \rightarrow \{0, 1\}^*$  la fonction qui extrait d'une suite de bits le plan de bits  $i$  ( $i \in \{0, \dots, b - 1\}$ ). Nous notons aussi  $\text{ApEn}_{m,r} : \{0, 1\}^* \rightarrow \mathbb{N}$  la fonction calculant l'entropie approximative [11] (Approximate Entropy en anglais) d'une suite de bits. Cette entropie approximative dépend de deux paramètres, une taille de fenêtre  $m$  et un seuil  $r$ .

Étant donné une fonction  $f$  (notre codeur) et une entrée  $X$ , nous proposons de mesurer le chaos  $\mathcal{D}_f(X)$  introduit dans la suite :

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

Dans cette métrique, la somme des entropies approximatives de chaque plan de bits  $i$  codés,  $\text{ApEn}_{m,r}(f(B_i(X)))$  est calculée. Intuitivement, quand le chaos interne à la séquence augmente, cette dernière devient plus difficile à prédire donc l'entropie approximative augmente. En effet, lorsque cette séquence est plus proche de l'aléatoire, il devient plus difficile de détecter des motifs pour la classification.

Il est à noter que le nombre de motifs peut aussi être du à une séquence d'entrée  $X$  particulière contenant peu d'information.

Il nous faut donc comparer la somme des données codées entropiquement à la somme des données non codées  $X$ . Enfin, cette mesure est moyennée sur le jeu de données pour assurer sa stabilité.

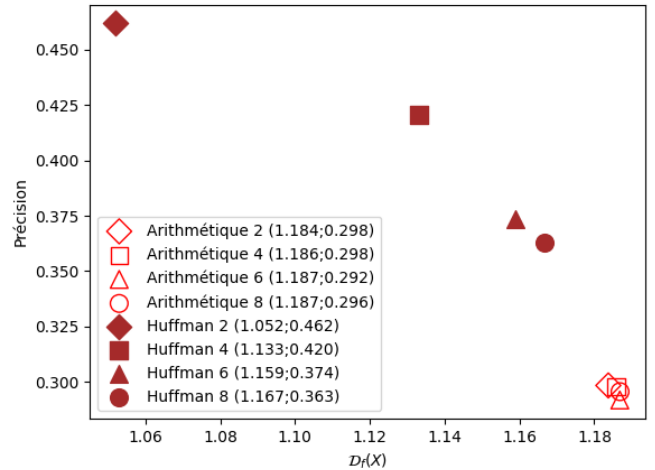


FIGURE 3 : Précision de UVGG11 sur YCIFAR-10 en fonction de  $\mathcal{D}_f(X)$  pour différentes transformations  $f$  (codage de Huffman and arithmétique avec taille des symboles d'entrée de 4, 6, 8, 10)

## 4 Expériences

### 4.1 Mise en place

Notre procédure expérimentale est basée sur [10] comme décrite dans la sous-partie 2.2 de ce papier. Pour un codeur entropique fixé, un réseau de neurones convolutifs est entraîné sur les données codées. L'évaluation de ce réseau entraîné nous donne sa précision. D'autre part, notre métrique  $\mathcal{D}_f$  est calculée avec comme paramètres  $m = 3$  et  $r = 0.5$ . La valeur de  $m$  influence la complexité du calcul de la métrique, elle doit donc rester faible. La valeur de  $r$  n'a pas d'influence dans notre cas binaire mais doit rester inférieure à 1 pour ne pas rendre la mesure constante. Nous recherchons ensuite les corrélations entre la perte de précision et notre métrique.

### 4.2 Résultats

Dans cette expérience, nous utilisons comme réseau de neurones convolutifs **UVGG11** [10] (Unidimensional VGG11 [13]). Ce réseau est entraîné en utilisant l'algorithme du gradient stochastique (ADAM [7]) avec un taux d'apprentissage de  $10^{-4}$ . Cela nous permet d'obtenir la précision du réseau pour plusieurs transformations : le codage de Huffman [12] et le codage arithmétique [12] pour des tailles de 4, 6, 8, 10.

La Figure 3 représente la précision du réseau de neurones convolutifs en fonction de notre métrique de mesure du chaos

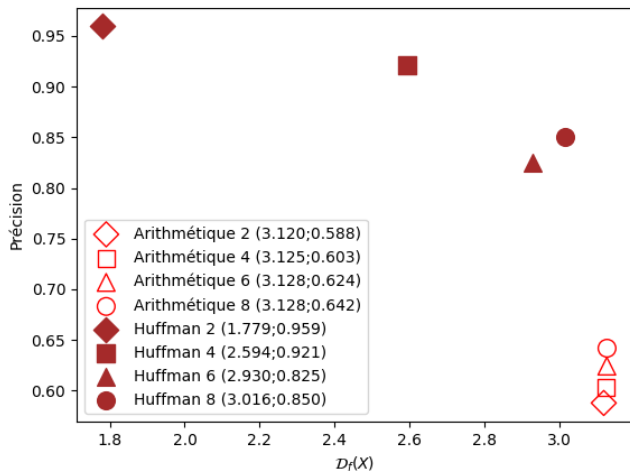


FIGURE 4 : Précision de UVGG11 sur MNIST en fonction de  $D_f(X)$  pour différentes transformations  $f$  (codage de Huffman and arithmétique avec taille des symboles d'entrée de 4, 6, 8, 10)

$D_f$ . Cette figure montre que la précision est une fonction monotone de la mesure que nous proposons en 3.2 pour mesurer le chaos. Il est à noter que la courbe est monotone non seulement pour un codeur entropique donné mais aussi d'un codeur entropique vis-à-vis d'un autre. Ceci prouve donc la pertinence de notre mesure.

Plus précisément, le jeu de données utilisé dans cette figure est YCIFAR-10 [10] (la version niveaux de gris de CIFAR-10 [8]) qui contient 10 classes avec des images réelles. Pour le codage de Huffman, la taille des symboles d'entrées varie entre 2 et 8 ; nous observons une bonne corrélation entre la précision réelle et note métrique. Pour le codage arithmétique, les précisions réelles sont proches pour les mêmes variations de la taille des symboles d'entrées, notre métrique les classes positionne donc dans la même zone. De plus, nous remarquons que la précision réelle de l'apprentissage sur des données codées arithmétiquement est plus faible que celle sur des données codées avec un codage de Huffman. Cet état de fait est fidèlement retranscrit par notre métrique (ce qui n'était pas le cas avec l'ancienne métrique dans le papier du workshop HMM-QOE2023). La Figure 4 représente la même précision en fonction de notre métrique mais pour un jeu de données différents : MNIST [9]. L'analyse de cette figure est concordante avec figure la précédente et renforce donc les conclusions antérieures.

## 5 Conclusion

Dans ce papier, nous nous intéressons au problème d'apprentissage sur les données codées entropiquement. En particulier, nous avons introduit une nouvelle mesure du chaos pour pouvoir quantifier la perte de précision introduite par le codage entropique.

## Références

[1] Aaron CHADHA, Alhabib ABBAS et Yiannis ANDREOPOULOS : Video Classification With CNNs : Using the Codec as a Spatio-Temporal Activity Sensor. *IEEE Tran-*

*sactions on Circuits and Systems for Video Technology*, 29(2):475–485, février 2019.

- [2] Lahiru D. CHAMAIN, Fabien RACAPÉ, Jean BÉGAINT, Akshay PUSHPARAJA et Simon FELTMAN : End-to-end optimized image compression for multiple machine tasks. *arXiv :2103.04178 [cs]*, mars 2021.
- [3] Lingyu DUAN, Jiaying LIU, Wenhan YANG, Tiejun HUANG et Wen GAO : Video Coding for Machines : A Paradigm of Collaborative Compression and Intelligent Analytics. *IEEE Transactions on Image Processing*, 29:8680–8695, 2020.
- [4] David EDMUNDSON et Gerald SCHAEFER : An overview and evaluation of JPEG compressed domain retrieval techniques. *In Proceedings ELMAR-2012*, pages 75–78, septembre 2012.
- [5] Yuming FANG, Weisi LIN, Zhenzhong CHEN, Chia-Ming TSAI et Chia-Wen LIN : A Video Saliency Detection Model in Compressed Domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(1):27–38, janvier 2014.
- [6] Junyi FENG, Songyuan LI, Xi LI, Fei WU, Qi TIAN, Ming-Hsuan YANG et Haibin LING : TapLab : A Fast Framework for Semantic Video Segmentation Tapping Into Compressed-Domain Knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1591–1603, mars 2022.
- [7] Diederik P. KINGMA et Jimmy BA : Adam : A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2014.
- [8] Alex KRIZHEVSKY : Learning Multiple Layers of Features from Tiny Images. Rapport technique, University of Toronto, avril 2009.
- [9] Yann LECUN : THE MNIST DATABASE of handwritten digits. <https://yann.lecun.com/exdb/mnist/>.
- [10] Rémi PIAU, Thomas MAUGEY et Aline ROUMY : Learning on entropy coded images with cnn. *In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, 2023.
- [11] S M PINCUS : Approximate entropy as a measure of system complexity. *Proc Natl Acad Sci U S A*, 88(6):2297–2301, mars 1991.
- [12] Khalid SAYOOD : *Introduction to Data Compression*. Elsevier, 2018.
- [13] Karen SIMONYAN et Andrew ZISSERMAN : Very Deep Convolutional Networks for Large-Scale Image Recognition. *In Yoshua BENGIO et Yann LECUN, éditeurs : 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [14] Robert TORFASON, Fabian MENTZER, Eirikur AGUSTSSON, Michael TSCHANNEN, Radu TIMOFTE et Luc Van GOOL : Towards Image Understanding from Deep Compression Without Decoding. *In International Conference on Learning Representations*, février 2018.