Information-Theoretic Analysis of Convolutional Autoencoders: Initial Insights

Frederico Carvalho Fontes do Amaral e Daniel Guerreiro e Silva

***Abstract*— Despite the success of deep neural networks to solve real world problems, the current theoretical comprehension of their learning mechanisms still deserves further analysis. Recently, various works have explored the use of information-theoretic concepts in order to tackle this issue. This work uses a framework derived from this theory to the study of convolutional autoencoders, in order to better understand its training mechanisms and suggest how the information quantities can be used to determine its bottleneck's size. We conclude by presenting a discussion based on the results obtained that may shed a light on network's learning mechanisms.**

***Keywords*— Information Theoretic Learning, Convolutional Neural Networks, Convolutional Autoencoders**

I. Introduction

Deep neural networks (DNNs) have been extensively studied in recent decades due to the various successes obtained through their application to real world problems [2]. Among the reasons for this success, one can highlight their capacity to capture the underlying structure and statistical behavior of large datasets. In spite of these successes, there is still a latent shortage of widely accepted systematic methods for their design.

In order to address this issue, Information Theoretic Learning (ITL), a framework which employs concepts from Information Theory (IT) within Machine Learning, has gained increasing attention in recent years. By using the well-established informational quantities derived from IT, the ITL framework allows the creation of systematic methods to design and analyze DNNs more rigorously [1]. In particular, it has been used to study Autoencoders (AEs), unsupervised DNNs architectures widely used for data compression, with considerable success in [5].

The AEs' goal is to create more compact representations of its inputs. By doing so, they suppress undesirable elements contained in the data (e.g. redundancies and noise), which in turn enables its better understanding and use [2]. AEs consist of two main parts: the encoder and the decoder. The former encodes the input into a lower dimension representation (i.e. a code), which is stored inside the AE's bottleneck layer, while the latter recreates the input using its respective code [4]. It is evident that the bottleneck's size is a crucial parameter for the proper functioning of AEs, because it determines their capacity to capture the underlying statistical structure of their inputs. If it is too big, not only the overall size of the AE will increase (alongside its memory and computational costs), but the code will contain redundant information. If it is too small, the AE will be unable to completely capture the relevant information of the input. In spite of its importance, the systematic evaluation of the bottleneck's size adequacy is still a topic poorly explored in the literature [9].

To address this issue, the authors of [9] proposed an automatic method to estimate the optimal size for the bottleneck of a stacked autoencoder (SAE). By using its code's entropy as a key performance indicator (KPI), this method's goal is to enable the SAE to achieve maximum data compression without compromising the overall performance of the decoder. In light of the positive results yielded by said method, this work aims to investigate the possibility of its application to determine the bottleneck's size of convolutional autoencoders (CAEs), which are convolutional neural networks (CNNs) structurally analogous to SAEs [4]. In order to do so, we estimated additional information quantities associated with the layers of a CAE's encoder during its training in order to study their behavior. We suggest an explanation for these metrics' evolution during the CAE's training, including the new ones defined in [6], based on the experimental results. Also, we argue that, differently from what was proposed in [9], instead of using only the entropy of the codes as a KPI, these new information quantities should also be used as KPIs.

The rest of the work is organized as follows. Section II exposes the information estimators used in the experiment. Section III details the experimental procedures and shows its results. Section IV discusses them, as well as suggests how the new information quantities proposed in [8] could be suitable KPIs for CAEs. Finally, Section V concludes this work and highlights topics that can be addressed in future studies.

II. Information Estimators

Originally proposed and formalized by Claude Shannon, Entropy and Mutual Information (MI) are the fundamental measurements of information [3]. While there are multiple definitions for these quantities, the family of parametric entropies formalized by Alfréd Rényi is widely used in ITL [1]. Let X be a u-dimensional random variable (RV) (e.g. an image) with probability density function (PDF) $p(\mathbf{x})$, Y be a v-dimensional RV with PDF $p(\mathbf{y})$ and $p(\mathbf{x}, \mathbf{y})$, their joint PDF. The α-Rényi entropy of X is defined as

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int p^\alpha(\mathbf{x})\ d\mathbf{x}, \tag{1}$$

the α-Rényi joint entropy of X and Y is defined as

Frederico Carvalho Fontes do Amaral, Departamento de Engenharia Elétrica, UnB, Brasília-DF, e-mail: 200054457@aluno.unb.br; Daniel Guerreiro e Silva, Departamento de Engenharia Elétrica, UnB, Brasília-DF, e-mail: danielgs@ene.unb.br. This work was funded by CAPES and Edital Conjunto PPGEE/PGEA UnB Nž 01/2021.

$$H_\alpha(X,Y) = \frac{1}{1-\alpha}\log\int p^\alpha(\mathbf{x},\mathbf{y})\; d\mathbf{x}\, d\mathbf{y}, \tag{2}$$

and the α-Rényi MI between X and Y can be expressed as

$$I_\alpha(X;Y) = H_\alpha(X) + H_\alpha(Y) - H_\alpha(X,Y). \tag{3}$$

The exact evaluation of expressions (1), (2) and (3) is often impossible in practice, because it requires the knowledge of $p(\mathbf{x})$, $p(\mathbf{y})$ and $p(\mathbf{x},\mathbf{y})$, which are usually unknown. The estimation of these PDFs in practice is often difficult as well, since it usually involves high-dimensional data, for which PDF estimation can be both unreliable and computationally unfeasible. To circumvent this issue, recent works have used the matrix-based Renyis α-order entropy functional, proposed in [7], to estimate these information quantities directly from data, without explicitly evaluating its underlying PDF.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathcal{X}$, be a set of independent and identically distributed (iid) samples of X. The Gram matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ is obtained through the evaluation of a real valued positive definite kernel $\kappa : \mathcal{X}\times\mathcal{X} \mapsto \mathbb{R}$ on all the pairs of samples from $\mathbf{X}$, i.e. $(\mathbf{K})_{ij} = \kappa(\mathbf{x}_i,\mathbf{x}_j)$. Then, $\mathbf{K}$ is normalized in order to create a matrix $\mathbf{A}$ such as $(\mathbf{A})_{ij} = \frac{1}{N}\frac{(\mathbf{K})_{ij}}{\sqrt{(\mathbf{K})_{ii}(\mathbf{K})_{jj}}}$ and $tr(\mathbf{A}) = 1$, which results in the matrix-based Renyis α-order entropy functional

$$S_\alpha(\mathbf{A}) = \frac{1}{1-\alpha}\log_2\left[\sum_{i=1}^N \lambda_i(\mathbf{A})^\alpha\right], \tag{4}$$

where $\lambda_i(\mathbf{A})$ is the i-th eigenvalue of $\mathbf{A}$. Also in [7], similarly to how (4) is used to estimate $H_\alpha(X)$, another matrix estimator was proposed to estimate $H_\alpha(X,Y)$. Although these estimators were applied to determine the bottleneck's size of the SAE with densely connected layers with considerable success [9], they cannot be directly applied to CNNs, as explained in [8]. Thus, the estimator proposed in [7] was expanded in [8] to estimate the multivariate mutual information (MMI) between a single RV and a group of RVs — e.g. the input of a CNN and the feature maps (FMs) of one of its layers. Let $\{\mathbf{s}_i = (\mathbf{x}_1^i,\ldots,\mathbf{x}_C^i)\}_{i=1}^N$, $C \geq 2$, be a collection of N samples obtained from the same realization containing C measurements $\{\mathbf{x}_p \in \mathcal{X}_p\}_{p=1}^C$ each and $\{\kappa_p : \mathcal{X}_p\times\mathcal{X}_p \mapsto \mathbb{R}\}_{p=1}^C$, positive-definite kernels. A matrix-based analog to Rényis α-order joint entropy among C RVs is defined as

$$S_\alpha(\mathbf{A_1},\ldots,\mathbf{A_C}) = S_\alpha\left[\frac{\mathbf{A_1}\circ\cdots\circ\mathbf{A_C}}{tr(\mathbf{A_1}\circ\cdots\circ\mathbf{A_C})}\right], \tag{5}$$

where $\{(\mathbf{A_p})_{ij} = \kappa_p(\mathbf{x}_p^i,\mathbf{x}_p^j)\}_{p=1}^C$ and $\circ$ denotes the Hadamard product. Let $\{T^p\}_{p=1}^C$ be the RVs associated with the C FMs in a CNN's convolutional layer. The MMI between these FMs and the input X is given by

$$\begin{aligned} I(X;\{T^1,\ldots,T^C\}) =& H(X) + H(T^1,\ldots,T^C) \\ &- H(X,\{T^1,\ldots,T^C\}). \end{aligned} \tag{6}$$

By inspecting (4), (5) and (6), it is immediate that the value of $I(X;\{T^1,\ldots,T^C\})$ in a mini-batch of size N can be estimated with

$$\begin{aligned} I_\alpha(\mathbf{B},\{\mathbf{A_1},\ldots,\mathbf{A_C}\}) =& S_\alpha(\mathbf{B}) + S_\alpha\left[\frac{\mathbf{A_1}\circ\cdots\circ\mathbf{A_C}}{tr(\mathbf{A_1}\circ\cdots\circ\mathbf{A_C})}\right] \\ &- S_\alpha\left[\frac{\mathbf{A_1}\circ\cdots\circ\mathbf{A_C}\circ\mathbf{B}}{tr(\mathbf{A_1}\circ\cdots\circ\mathbf{A_C}\circ\mathbf{B})}\right], \end{aligned} \tag{7}$$

where $\mathbf{B}$ and $\mathbf{A_1},\ldots,\mathbf{A_C}$ denote the Gram matrices evaluated on the input tensor and C FM tensors, respectively.

By estimating the MMI between these RVs using (7), the authors of [6] measured the amount of information about X that was captured by all the FMs inside the bottleneck layer. They also used the partial information decomposition (PID) framework to understand how the redundancy and synergy between different FMs evolved during training. According to this framework, the MMI $I(X;\{T^i,T^j\})$ between the input X and the pair of FMs T^i and T^j can be written as the sum of four nonnegative IT components: the unique information associated with T^i and T^j, redundancy and synergy. The first two, denoted by $Unq(X;T^i)$ and $Unq(X;T^j)$, measure the information about X that can be exclusively provided by T^i and T^j respectively. The third, denoted by $Rdn(X,\{T^i,T^j\})$ measures the shared information about X that can be provided by either T^i or T^j. Lastly, the fourth, denoted by $Syn(X,\{T^i,T^j\})$, measures the information about X provided by the combination of T^i and T^j (i.e., the information that cannot be captured by either T^i and T^j alone). All these quantities satisfy

$$\begin{aligned} I(X;\{T^i,T^j\}) =& Syn(X,\{T^i,T^j\}) + Rdn(X,\{T^i,T^j\}) \\ &+ Unq(X;T^i) + Unq(X;T^j), \end{aligned} \tag{8}$$

$$I(X;T^i) = Rdn(X,\{T^i,T^j\}) + Unq(X;T^i) \tag{9}$$

and

$$I(X;T^j) = Rdn(X,\{T^i,T^j\}) + Unq(X;T^j). \tag{10}$$

From their definition, it is preferable to maximize $Unq(X;T^i)$, $Unq(X;T^j)$ and $Syn(X,\{T^i,T^j\})$ while minimizing $Rdn(X,\{T^i,T^j\})$, since this would maximize the amount of non-redundant information between T^i and T^j. However, direct estimation of these quantities currently is not possible. To circumvent this problem, the authors of [6] proposed two new information quantities to characterize intrinsic properties of the CNN's layer representations: redundancy-synergy trade-off (RST) and weighted non-redundant information (WNRI). The RST_{ij} between X, T^i and T^j measures the trade-off of redundancy and synergy between these RVs, being defined as

$$\begin{aligned} \text{RST}_{ij} &= I(X;T^i) + I(X;T^j) - I(X;\{T^i,T^j\}) \\ &= Rdn(X;\{T^i,T^j\}) - Syn(X;\{T^i,T^j\}). \end{aligned} \tag{11}$$

During training, the RST between the input layer and a given convolutional layer can be estimated by averaging the sum of the RST_{ij} of each pair of FMs in the latter, which gives

$$\text{RST} = \frac{2}{C(C-1)} \sum_{i=1}^{C} \sum_{j=i+1}^{C} \text{RST}_{ij}. \quad (12)$$

Although a negative RST during training would be ideal (i.e., synergy dominates over redundancy), practical experimentation done so far indicates that it is always positive [6]. Thereby, it should be as small as possible. The WNRI_{ij} between X, T^i and T^j, in turn, measures the amount of non-redundant information about X that is captured by the pair of FMs T^i and T^j, being defined as

$$\text{WNRI}_{ij} = 2.I(X; \{T^i, T^j\}) - I(X; T^i) - I(X; T^j). \quad (13)$$

As the RST, the WNRI between the input layer and a given convolutional layer can be estimated by averaging the sum of the WNRI_{ij} of each pair of FMs in the latter, which results in

$$\text{WNRI} = \frac{2}{C(C-1)} \sum_{i=1}^{C} \sum_{j=i+1}^{C} \text{WNRI}_{ij}. \quad (14)$$

Expressions (12) and (14) are important because they grant access to the insights provided by both the redundancy and synergy without their explicitly estimation. It is evident that, to maximize the amount of non-redundant information present in two given FMs, the WNRI's value should increase during training. On the other hand, in order to prevent redundancy from limiting the capacity of those FMs to store non-redundant information, the RST's value (always positive in practice) should be kept as small as possible during training. It is also worth noting that, by adding (11) and (13), we obtain

$$I(X; \{T^i, T^j\}) = \text{RST}_{ij} + \text{WNRI}_{ij}, \quad (15)$$

which suggests that, from (12) and (14), we obtain

$$\text{RST}_{\%} = \frac{2}{C(C-1)} \sum_{i=1}^{C} \sum_{j=i+1}^{C} \frac{\text{RST}_{ij}}{I(X; \{T^i, T^j\})} \quad (16)$$

and

$$\text{WNRI}_{\%} = \frac{2}{C(C-1)} \sum_{i=1}^{C} \sum_{j=i+1}^{C} \frac{\text{WNRI}_{ij}}{I(X; \{T^i, T^j\})}. \quad (17)$$

Equations (16) and (17) show the percentages that the RST and WNRI account for the MMI in each pair of FMs, respectively.

III. Experimental Procedures and Results

The real-world dataset "Wiki-Art: Visual Art Encyclopedia" [11] was selected for training and evaluation of the CAE. It has a size of 37 GB and consists of 72.619 RGB images of various works of art, with distinct qualities and dimensions, distributed between 14 different classes according to their classification (e.g, abstract and animal paintings). This dataset was chosen due to the substantial differences between the themes of its images and the peculiarities of their various authors' painting styles, which make it very rich from an information standpoint. Before the experiment, the images contained in the dataset were shuffled and partitioned into 75% of the examples for training and 25% for testing. Besides conversion from RGB to grayscale and resizing of the images to 128×128 pixels, no image pre-processing was conducted. The codes were written in Python using the Keras API and various Python libraries.

Two symmetric CAEs, denoted by CAE_1 and CAE_2, were used in the experiments. Both consisted of seven hidden convolutional layers: three were located in the encoder, one in the bottleneck and three in the decoder. Table I contains the amount of FMs inside the bottleneck (Z), first (Conv_1), second (Conv_2) and third (Conv_3) convolutional layers of both CAEs' encoders. All layes used the rectified linear unit (ReLU) function and had stride 1. After each layer other than the Z one, there was a Max Pooling layer with stride 2.

TABLE I
CAEs' encoders' specifications.

	Conv_1	Conv_2	Conv_3	Z
CAE_1	16	32	64	8
CAE_2	64	32	16	8

These CAEs were chosen to investigate the effects, caused by the FMs' placement inside the convolutional layers, on the information quantities' evolution during training. The experiment was conducted using the Python 3 Google Computer Engine backend (GPU) with 83,49 GB of RAM and 166,83 GB of Disk. The CAE was trained via SGD using the "Adam" optimizer and the MSE loss function. A mini-batch size of 128 was adopted, and the CAE was trained for 50 epochs in order to allow the information quantities' stabilization. For their estimation, we fixed $\alpha = 2$ and used the radial basis function (RBF) kernel often used in the literature [5], [6] to obtain the Gram matrices. The kernel size σ is determined based on Silverman's rule of thumb $\sigma = h.N^{-1/(4+d)}$, where N is the mini-batch's size, d is the sample dimensionality, and h is an empirical value selected experimentally. Similarly to [6], we pick $h = 5$. Differently from them, however, no vector rastering was done in order to preserve the spatial relationships between neighboring pixels.

A. Experimental Results

Before the experiment, the images were shuffled and separated into 425 mini-batches for training, each containing 128 images. The information quantities were estimated after each epoch's completion. The evolution of the information quantities' magnitudes during the training of CAE_1 and CAE_2 are depicted in Figs. 1 and 2, respectively. Moreover, during the last epoch of the training of both architectures, it was observed that the MSE between the input and its reconstruction remained below 0,015.

IV. Discussion of the Experimental Results

In both figures it is evident that, the further a given layer is from the CAE's input, the slower its entropy will converge to

the MMI between the CAE's input and output. This is expected due to the fact that information is lost when propagated through the CNN. Therefore, the deeper a hidden layer is, the larger will be the amount of epochs necessary for its entropy to converge to the MMI. One can also see that, even after the entropy of a given encoder layer converges, the values of both its RST and WNRI can keep changing during training. They also show that, even though CAE_1 and CAE_2 have similar structures and the same bottleneck size, the placement of the FMs inside Conv_1 and Conv_3 has a substantial impact on the value to which the codes's entropies converge.

These results show that, differently from what was observed in [9], the CNN's code entropy cannot be used as the sole KPI for evaluating the bottleneck's size adequacy. From its definition, it is clear that it is desirable for the bottleneck's WNRI value to be as large as possible, because it indicates that a larger amount of the information stored in its code is non-redundant. The fact that both the RST and WNRI continue to evolve in spite of the convergence of the code's entropy, indicates that their evolution and convergence should also be taken into account when evaluating the adequacy (or not) of the bottleneck's size. This is, in turn, due to the fact that both quantities are intrinsically related with the amount of non-redundant information stored inside the FMs of the bottleneck layer, whose maximization should be one of the CAE's main goals. Furthermore, it suggests that the convergence of the bottleneck's WNRI can be used as a parameter for evaluation of the CAE's training process. Its evolution during it may indicate the CAE's effort to maximize the amount of non-redundant information, which indicates that its convergence can be an indicator of the interruption of this effort. However, more studies with different architectures and datasets are necessary to verify the possibility of using the WNRI to this end.

The results also show that CAE_1 clearly had a superior performance than CAE_2 from an information standpoint. This is shown by the fact that the entropy of CAE_1's codes converged to the MMI between input and output, while the entropy of CAE_2's converged to a value almost 1 bit lower. Also, the value reached by CAE_1's bottleneck layer's WNRI, whose convergence was not even achieved, is almost twice as large as that to which the bottleneck layer's WNRI of CAE_2 converged to. These results indicate that not only CAE_1 was able to capture the statistical structure of the dataset, but also that it did so while continuously increasing the amount of non-redundant information in its codes. Given the structures of CAE_1 and CAE_2, these results may suggest that, instead of using a larger number of FMs in the shallower layers (as is commonly done), using less FMs in these layers and more in deeper ones may yield more positive results from an information perspective. This observation is consistent with the results obtained in various repetitions of this experiment conducted by the authors, whose results cannot be shown in this work due to space limitations.

V. Conclusion and Future Work

This work presents a study of a CAE from an ITL perspective by using the efficient matrix-based information estimators proposed in [8]. The main scope of this study was to investigate if the method for automatic estimation of the optimal dimension of the bottleneck layer in densely connected SAEs proposed in [9] could be extended for the sizing of CAEs. In order to so, the evolution of different information quantities associated with multiple layers was regularly evaluated during their training. The results may suggest that the entropy of the codes cannot be used as the sole KPI for CAEs due to the fact that its convergence does not imply the convergence of both the RST and WNRI, whose evolution indicates an effort by the CAE to maximize the amount of non-redundant information stored in its bottleneck. Also, the WNRI's convergence may indicate the interruption of this effort, which could make it useful as a KPI for the training process quality, although further investigation is necessary to validate this hypothesis. Furthermore, it is worth mentioning that, by resizing the images to 128×128 pixels, some of their elements' variance (e.g., shape of objects) most likely increased, which probably influenced the information quantities estimated. In order to prevent this issue, we suggest that future works use CAEs capable of processing inputs with variable sizes, thereby eliminating the need for their resizing. Finally, the experiment also indicates that, by placing more FMs inside the deeper convolutional layers than in the shallower ones, the CAEs may be capable of yielding better results from an information perspective. We also leave further evaluation of this possibility as a future work.

References

[1] Principe, Jose C.: Information theoretic learning: Renyi's entropy and kernel perspectives. 1st edn. Springer Science & Business Media (2010)

[2] Alpaydin, Ethem: Introduction to Machine Learning. 4th edn. The MIT Press (2020)

[3] Thomas M. Cover, Joy A. Thomas: Elements of Information Theory. 2nd edn. John Wiley & Sons (2008)

[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning. MIT Press, `http://www.deeplearningbook.org` (2016). Last accessed 19 Mar 2022

[5] Yu, Shujian, Principe, Jose C.: Understanding autoencoders with information theoretic concepts. Neural Networks, vol. 117, 104–123 (2019)

[6] S. Yu, K. Wickstrøm, R. Jenssen, J. C. Príncipe: Understanding Convolutional Neural Networks With Information Theory: An Initial Exploration. IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, 435–442 (2021).

[7] Sanchez Giraldo, Luis Gonzalo, Murali Rao, Jose C. Principe: Measures of Entropy From Data Using Infinitely Divisible Kernels. IEEE Transactions on Information Theory, vol. 61, no. 1,, 535–548 (2015).

[8] Yu, Shujian, Giraldo, Luis Gonzalo Sánchez, Jenssen, Robert, Príncipe, José C.: Multivariate Extension of Matrix-Based Rényi's α-Order Entropy Functional. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 11, 2960–2966 (2020).

[9] G. Boquet, E. Macias, A. Morell, J. Serrano and J. L. Vicario: Theoretical Tuning of the Autoencoder Bottleneck Layer Dimension: A Mutual Information-based Algorithm. 2020 28th European Signal Processing Conference (EUSIPCO), 1512–1516 (2021).

[10] Siradjuddin, Indah Agustien, Wardana, Wrida Adi, Sophan, Mochammad Kautsar: Feature Extraction using Self-Supervised Convolutional Autoencoder for Content based Image Retrieval. 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 1–5 (2019).

[11] Wiki-Art: Visual Art Encyclopedia, `https://www.kaggle.com/datasets/ipythonx/wikiart-gangogh-creating-art-gan`. Last accessed 4 Apr 2022

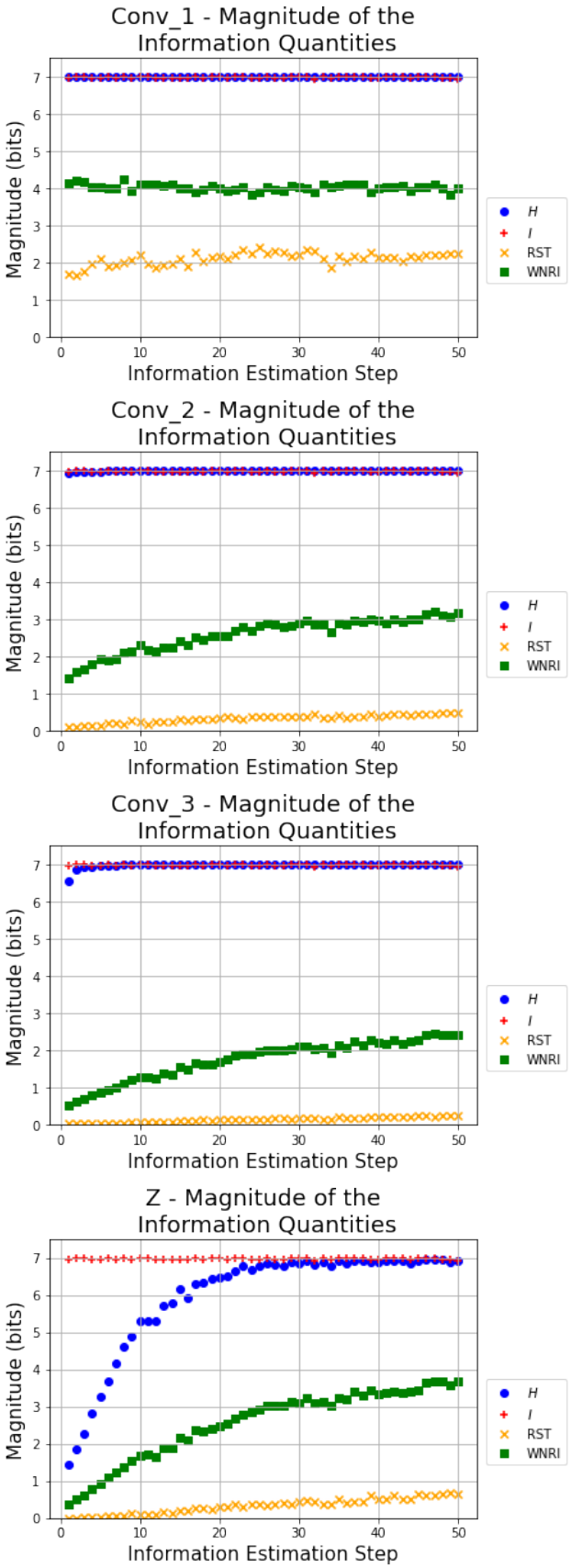

Fig. 1. Evolution of the CAE_1's information quantities' magnitudes.

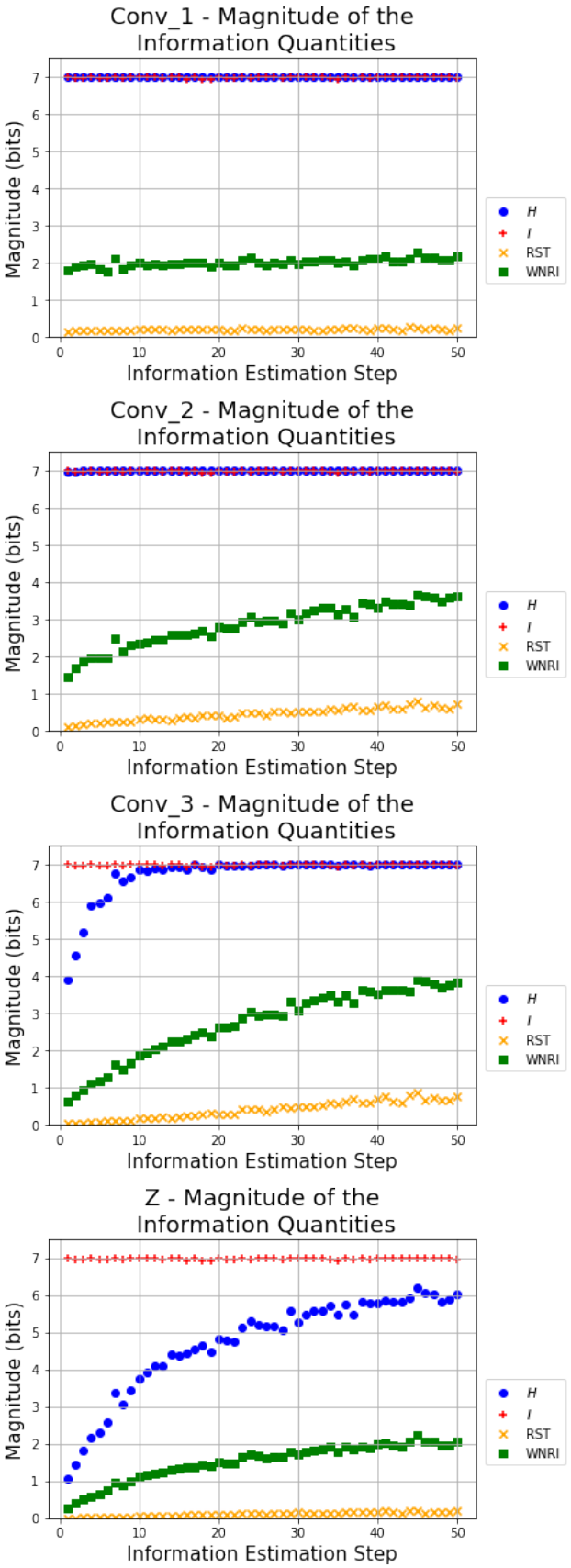

Fig. 2. Evolution of the CAE_2's information quantities' magnitudes.