

FEDERAL UNIVERSITY OF RIO GRANDE - FURG
CENTER OF COMPUTATIONAL SCIENCES
POST-GRADUATE PROGRAM IN COMPUTING
MASTER'S DEGREE IN COMPUTER ENGINEERING

Master's Degree Dissertation

**Fusion functions inspired on the Choquet integral in the
pooling layer of Deep Learning Networks**

Camila Alves Dias

Master's Dissertation presented to the Post-graduation Program in Computing of the Federal University of Rio Grande - FURG, as a partial requirement to obtain a Master's degree in Computer Engineering.

Advisor: Prof. Dr. Graçaliz Pereira Dimuro
Co-advisor: Prof. Dr. Silvia Silva da Costa Botelho

Rio Grande, 2019

Ficha catalográfica

D541f Dias, Camila Alves.

Funções de fusão inspiradas na integral de Choquet na camada de *pooling* de Redes de Aprendizagem Profunda / Camila Alves Dias. – 2019.

56 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2019.

Orientadora: Dra. Graçaliz Pereira Dimuro.

Coorientadora: Sílvia Silva da Costa Botelho.

1. Integral de Choquet 2. Funções de Agregação 3. Identificação de Capacidade 4. Processamento de Imagens 5. Rede de Aprendizagem Profunda I. Dimuro, Graçaliz Pereira II. Botelho, Sílvia Silva da Costa III. Título.

CDU 004

DISSERTAÇÃO DE MESTRADO

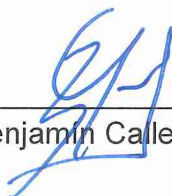
**Fusion functions inspired on the Choquet integral in the pooling layer of
Deep Learning Networks**

Camila Alves Dias

Banca examinadora:



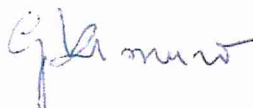
Prof. Dr. Humberto Bustince



Prof. Dr. Benjamin Callejas Bedregal



Prof. Dr. Eduardo Nunes Borges



Prof. Dr. Graçaliz Pereira Dimuro
Orientador(a)

I dedicate this to my family and friends; semper ad meliora.

ACKNOWLEDGMENT

To my family, for the ability to believe and invest in me. My mother and father, by the presence that meant the certainty that I was not alone, and the hope to follow this walk. Thanks to the teachers Dr. Graçaliz Dimuro, Dr. Eduardo Borges and Dr. Silvia Botelho for having participated with such dedication during the course of the present work. Thanks also to CAPES for the financial support received.

If they give you ruled paper, write the other way.
— RAY BRADBURY

ABSTRACT

DIAS, Camila Alves. **Fusion functions inspired on the Choquet integral in the pooling layer of Deep Learning Networks**. 2019. 56 f. Dissertation (MSc Degree) – Post-graduate Program in Computing. Federal University of Rio Grande - FURG, Rio Grande.

The image classification is one of the most studied problems in the area of computational vision. Some of the problems faced in this context are, for example, the characterization of images patterns to distinguish natural species, classification of collected data, which in general involves complex information to be identified, requiring resources as, e.g., machine learning tools, such as Convolutional Neural Networks (CNNs) and Deep Learning Networks (DLNs). This Master's dissertation explores the use of fusion functions inspired on the Choquet integral in the pooling layer of CNN architecture, presenting a two-folded general objective. First, we study the application of (pre) aggregation functions based on the generalizations of the Choquet integral in image dimensional reduction, simulating the pooling layer of a DLN, comparing such functions with the usual ones used in the literature (the maximum and arithmetic mean). A quantitative evaluation was done over an image dataset by using different image quality measures to compare the results. The second part of the dissertation is aimed to introduce a fusion function inspired in the Choquet integral for DLN pooling layer, defined by a capacity-like function which is learned by the own model. Using CifarNet (a simple architecture for classifying objects), we analyse the proposed approach in the image classification. The results are compared with the ones obtained when using the maximum in the pooling layer.

Keywords: Choquet integral, Aggregation functions, Capacity identification, Image processing, Deep Learning Network.

LIST OF FIGURES

1	MNIST Dataset and Number Classification	16
2	Architecture of a CNN (SCHERER; MULLER; BEHNKE, 2010) for NORB (an object recognition data set) experiments, consisting of alternating convolution and grouping layers. The values above the images represent the number of layers and the size in pixels of the images. Grouping layers can implement subsampling or grouping operations of maxima.	18
3	The process steps of maximum aggregation and resizing of the output image.	19
4	Simple example of a neural network architecture.	27
5	Example of a four-layer network has two hidden layers.	27
6	Representation of Gradient Descent (with the objective of minimizing the cost function).	28
7	Representation in graph of forward-mode differentiation or forward pass.	30
8	Representation in graph of reverse-mode differentiation (backpropagation algorithm).	30
9	Application of max pooling in a location divided into four quadrants.	31
10	Illustration of a disadvantage of mean function application (YU et al., 2014).	32
11	Illustration of a disadvantage of maximum function application (YU et al., 2014).	32
12	Illustration of image processing chain containing the diferent tasks.	33
13	Results of max, mean, Choquet integral and its generalizations obtained through the experiments applied to one of the images of the IIIT 5K-Word data set called 138_6, without resize. The parameters used were: window size = 4x4, stride = 2 and power fuzzy measure exponent = 0.7	37
14	Results of max, mean, Choquet integral and its generalizations obtained through the experiments applied to one of the images of the IIIT 5K-Word data set called 127_3, without resize. The parameters used were: window size = 2x2, stride = 2 and power fuzzy measure exponent = 0.1	38
15	Dataset used in CifarNet.	39
16	CifarNet Architecture used in this research.	40

17	Pool W exponents for layer 1 and 2	43
18	Pool W for layer 1 and 2	43
19	Losses during steps from NMCI and Max. (Max=red, NMCI=blue) .	44
20	Difference in losses for each pooling.	44
21	Table for different situations of the pooling window and the approxi- mate result of the pooling layer.	45
22	Convolution weights and biases for the max-pooling network.	46
23	Convolution weights and biases for the choquet-pooling network. network.	47
24	Activations for each convolution in a choquet-pooling network.	47
25	Activations for each convolution in a max-pooling network.	48
26	Comparison of each pool function for a random sample (N=1000) in [0, 1] ⁴	48

LIST OF TABLES

1	Cut of the results of the Choquet integral and maximum in the pooling layer obtained after performing 1407 iterations in CNN with the dataset MNIST	16
2	Average results obtained from the image quality measures for each aggregation function. The values in bold are the best results for the measurement indicated in table.	36
3	Best image quality measurement results from the combination of two pre-aggregation functions in the resizing of images for classification .	38
4	Accuracy comparison for max-pooling and Choquet integral pooling after 10000 iterations on CifarNet using CIFAR10 dataset.	49
5	Results clipping of image resizing using different combinations of functions, measures and parameters. Author: Cédric Marco-Dechart .	56

LIST OF ABBREVIATIONS AND ACRONYMS

DLN	Deep Learning Network
CNN	Convolutional Neural Network
AI	Artificial Intelligence
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
NCC	Normalized Cross-Correlation
AD	Average Difference
SC	Structural Content
MD	Maximum Difference
NAE	Normalized Absolute Error
PCA	Principal Component Analysis
NMCI	Non-Monotonic Choquet integral

SUMMARY

1	INTRODUCTION	14
1.1	Motivation and justification	15
1.2	General objective	16
1.2.1	Specific objectives	17
1.3	Methodology	17
1.4	Organization of text	21
2	PRELIMINARY CONCEPTS	22
2.1	Generalizations of the Choquet integral	22
2.2	Deep Learning Networks	26
2.2.1	Neural network architecture	26
2.2.2	Pooling layer	30
2.3	Neural networks in the context of image processing	32
2.3.1	Preprocessing	33
2.3.2	Data reduction	33
2.3.3	Image segmentation	33
2.3.4	Object recognition	33
2.3.5	Image understanding	33
2.3.6	Optimization	33
3	USING THE CHOQUET INTEGRAL IN THE POOLING LAYER IN DEEP LEARNING NETWORKS	34
3.1	Analyzes and results	36
3.2	Concluding remark of the first phase	37
4	IMAGE CLASSIFICATION USING A POOLING FUNCTION BASED ON THE DISCRETE CHOQUET INTEGRAL WITH THE LEARNING OF NON MONOTONIC FUZZY MEASURE	39
4.1	Theoretical developments	40
4.1.1	Motivation for working with non-monotonic fuzzy measures	40
4.1.2	The non-monotonic discrete Choquet integral	41
4.1.3	Modeling the inputs of the non-monotonic discrete Choquet integral in the learning process of the non-monotonic fuzzy measure	41
4.2	W-evolution	43
4.2.1	Comparison Non-Monotonic Choquet integral pooling against Max-pooling in training	44
4.2.2	Behaviour of the learnt weights in the 2×2 pool window	45
4.2.3	Final results of the classification	46

5	CONCLUSION	50
	REFERENCES	51
6	ANNEX 1	56

1 INTRODUCTION

The perform of algorithmic processes on image dataset is becoming possible with the use of high-speed computers. A very complex process used today is the classification of images, which can be influenced by several factors (LU; WENG, 2007) and applied in fields such as biomedical images (MARÉE et al., 2005), biometry (BRADY, 1999) and others. The steps followed by image classification are: pre-processing, segmentation, extraction of characteristics and classification. These steps are precisely best performed with an artificial neural network, which is an area where several image classification techniques are applied for better performance. Neural networks improves supervised and unsupervised image classification using methods such as Convolutional Neural Networks (CNNs) and Deep Convolutional Neural Networks (DNNs) (RAWAT; WANG, 2017).

The purpose of this research is to experiment with new approaches for image classification using DLN inspired on the concept of Choquet integral (CHOQUET, 1953–1954) with a nonadditive measure (DENNEBERG, 2013). For accomplish the research a two-folded general objective was established. Firstly some kind of (pre) aggregation functions, which are based on the generalizations of the Choquet integral, are applied in image dimensional reduction, simulating the pooling layer of a DLN. The results are compared with the usual ones normally used in the literature (the maximum and arithmetic mean). At the end the results are evaluated using qualitative (visually) and quantitative (using image quality measures) methods. To carry out the second phase of the work we propose a way to better refine the results using the standard Choquet integral, which obtained better results in the first phase of this research.

Thus a fusion function inspired in the Choquet integral for DLN pooling layer is presented, defined by a capacity-like function which is learned by the own model¹. This capacity-like fusion function, used in the place of the fuzzy measure of a Choquet integral is learned by the neural network using the backpropagation algorithm (LEONARD; KRAMER, 1990). Using CifarNet architecture (KRIZHEVSKY; HINTON, 2009), which is a simple model for classifying objects, we analyse the proposed approach in image classification. The results are compared with the ones obtained when using the maximum

¹A capacity (CHOQUET, 1953–1954) is also known as fuzzy measure (SUGENO, 1974).

in the pooling layer.

1.1 Motivation and justification

Deep structured learning, also known as Deep Learning Network (DLN) has gained greater attention, standing out as a new area of research for learning machines (DENG; YU et al., 2014). Deep learning can be broadly defined as: a class of machine learning techniques that analyze many layers of nonlinear data processing for extraction and modification of supervised or unsupervised aspects and for recognition and classification of paradigms (DEEPIKA JASWAL SOWMYA VISHVANATHAN, 2014).

The networks structure is modified according to the need of its use. There are several layers in its architecture, the pooling layer usually being used after the convolution and activation layers. The purpose of this layer is to decrease the dimensionality of the data in the network. This reduction step has a great importance, since the agility of the training will be expanded (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). The pooling layer acts by aggregating a group of data. For example: the input (an array, image, etc.) is divided into windows 4x4 and from each is assigned a value to represent it, defined by a function, such as max pooling or mean pooling.

This work presents the pre-aggregation function called Choquet integral with the objective of simulating the pooling layer used with the functions known in the literature by the Choquet integral and its generalizations, in order to improve the aggregation of meaningful information without degrading its discriminative power in image processing of the pooling layer in DLNs and to obtain better or closer results in image classification when compared to those using the usual pooling functions.

The main motivation is based on an experiment that was performed using the MNIST database in a simple CNN for classification of handwritten numbers (LECUN et al., 1998). MNIST dataset has 60,000 training images and 10,000 testing images taken from American Census Bureau employees and American high school students (KUSSL; BAIDYK, 2004).

The results obtained after changing the function of pooling from maximum to Choquet integral were higher, as shown in the Table 1.

Then, motivated by those results unlike using the power measure in the discrete Choquet integral equation, an alternative was created where the network itself will learn a new measure, called capacity-like fusion function. Such fusion function is not monotonous, so it is not recognized as a fuzzy measure. Learning this capacity-like fusion function is important, where the network at the time of the backpropagation function corrects the errors, making the result even more accurate.

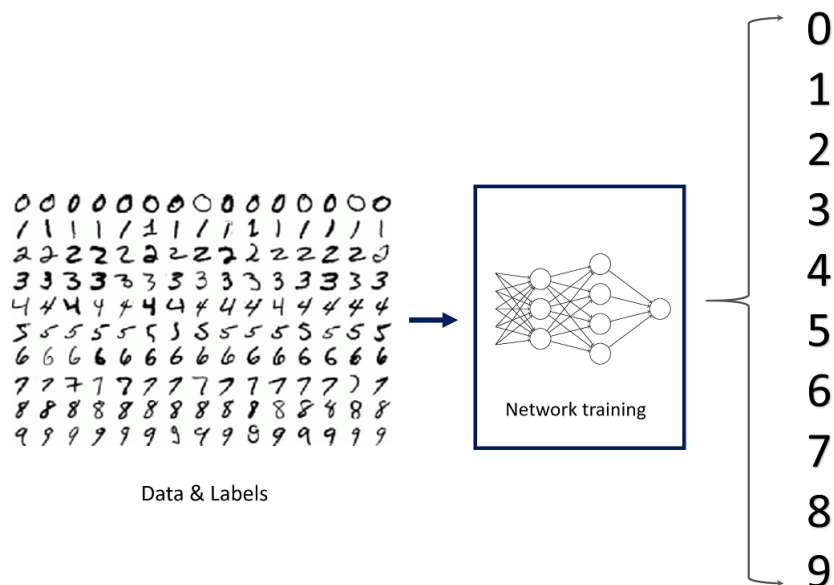


Figure 1: MNIST Dataset and Number Classification

Table 1: Cut of the results of the Choquet integral and maximum in the pooling layer obtained after performing 1407 iterations in CNN with the dataset MNIST

Iterations	Choquet pooling		Max pooling	
	Accuracy	Cross entropy	Accuracy	Cross entropy
469	0,892	0,299	0,888	0,304
938	0,896	0,297	0,896	0,287
1407	0,9	0,305	0,903	0,283
1876	0,902	0,306	0,902	0,299
2345	0,903	0,319	0,899	0,324
2814	0,904	0,325	0,9	0,349
3283	0,898	0,363	0,897	0,378
3752	0,898	0,398	0,899	0,381
4221	0,898	0,404	0,903	0,372
4690	0,903	0,403	0,894	0,432
5159	0,904	0,409	0,901	0,431
5628	0,904	0,417	0,907	0,418
6097	0,908	0,424	0,902	0,464
6566	0,909	0,433	0,903	0,444

1.2 General objective

The research general objective is to study the use of fusion functions inspired on the Choquet integral and apply in the pooling layer of DLN architecture, considering two directions:

1. to study the application of (pre) aggregation functions based on the generalizations of the Choquet integral in image dimensional reduction, simulating the pooling layer of a DLN;
2. to introduce a new fusion function inspired by the (pre) aggregation functions analysed in item 1 for DLN pooling layer, defined by a capacity-like function which is

learned by the own model.

1.2.1 Specific objectives

The specific objectives are separated into two phases, which were accomplished the two parts of the dissertation.

First phase:

1. To implement algorithms to calculate the Choquet integral and selected generalizations in MatLab using fuzzy power measure;
2. To search for a qualified image dataset for application of the algorithms;
3. To perform tests on multiple image datasets with the proposed functions;
4. To analyze the output results performed in MatLab using image quality measures;
5. To compare the results with the maximum and arithmetic mean.

Second phase:

1. To define a function inspired on the (pre) aggregation that performed better in the first phase, using a capacity-like fusion function in the place of the fuzzy measure, which is to be learned by the model;
2. To select a DLN architecture to replace the maximum function of the pooling layer by the proposed function;
3. To implement the function in the Python language to introduce in the DLN architecture;
4. To study the change of the fuzzy power measure learning by the capacity learning;
5. To realize the chosen network training;
6. To analyze network output data.

1.3 Methodology

A common technique for obtaining invariant means in object recognition paradigms is to aggregate several low-level features into a discrete neighborhood. However, the divergences between these models make a comparison of the characteristics of difficult distinct aggregation functions. The maximum and mean arithmetic functions are the most common aggregation functions used in the pooling layer of a Convolutional Neural Network (CNN) (SCHERER; MULLER; BEHNKE, 2010). To improve the aggregation of meaningful information without degrading its discriminative power for image processing, first

we propose the substitution of the aggregation function previously described by the (pre) aggregation functions called Choquet integral and its generalizations. However, we do not apply the functions directly in the network architecture, we just simulate the pooling behavior. This is a previous stage, in order to analyze which (pre) aggregation functions will have a better perform.

The pooling layer reduces the size of the input by synthesizing the neurons of a small neighborhood. Figure 2 demonstrates a example of a general a CNN architecture. The values above the images represent the amount and size (in pixels) of the images. For example: in the first network layer (input layer) there are 2 images with the size of 96x96 each. After the first convolution, 16 images with the size of 92x92 were generated. At the end, these layers will be fully connected creating a unique volume.

Related to the pooling layer there are some important parameters determined by the programmer: the window size and the stride. The window size agrupates the image pixels values in a matrix. The stride counts how many columns of the matrix (pixels of an image) will be jumped, which represents the next layer size of the processed input.

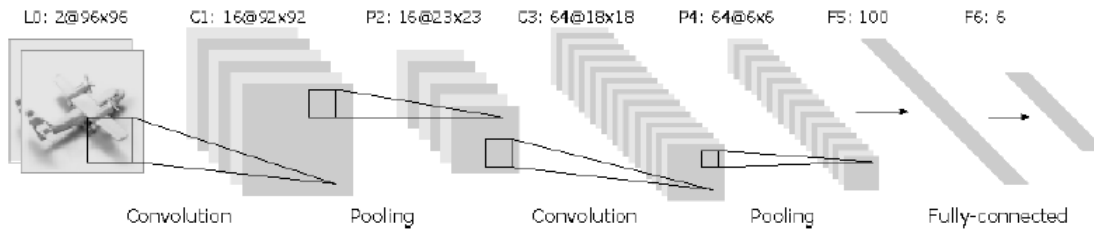


Figure 2: Architecture of a CNN (SCHERER; MULLER; BEHNKE, 2010) for NORB (an object recognition data set) experiments, consisting of alternating convolution and grouping layers. The values above the images represent the number of layers and the size in pixels of the images. Grouping layers can implement subsampling or grouping operations of maxima.

The input data set is composed by images that simulate the pooling layer of a CNN only with the filter application, i.e. without applying on the network architecture.

The domain of the Choquet integral are the pixels values of an input image, which can vary from 0 to 255. These values were stardalized between 0 and 1 to use the Choquet integral. In order to evaluate the resulting image quality, the quality measures (ESKICI-UGLU; FISHER, 1995) were applied. In addition, to facilitate the analysis of the results, the input images are converted to grayscale.

The resizing is then used to return the same size of the input image. To perform a resizing, the nearest function in Matlab software is applied. The technique is used after aggregation with the purpose of comparing the input image with the resulting image, since it is only possible to apply image quality measures once the input image has the same size as the output image. We observe that this resizing method is not the best one that may be found in the literature, such as image magnification using interval information (JURIO

et al., 2011). However, since the resizing is not the focus of this work, we decided to use it due to its simplicity.

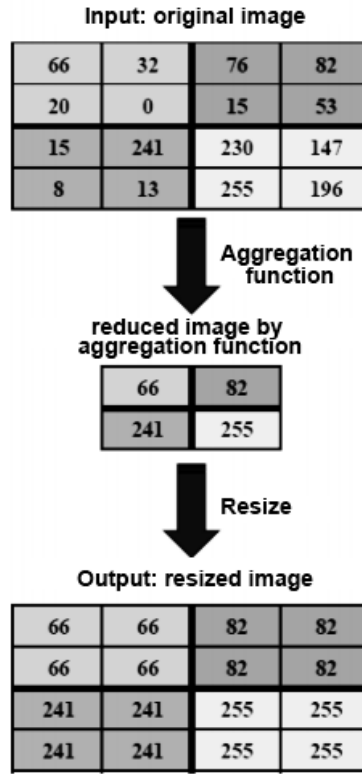


Figure 3: The process steps of maximum aggregation and resizing of the output image.

Aggregation functions were performed with several parameters (stride, window size and fuzzy power measure exponent value) chosen by experts.

After this process, image quality measurements are applied. To perform that, seven quality measures were adopted (ESKICIOGLU; FISHER, 1995): Average Difference (AD, must be zero), Structural Content (SC↓), Normalized Cross-Correlation (NK↑), Maximum Difference (MD↓), Normalized Absolute Error (NAE↓), Mean Squared Error (MSE↓) and Peak Signal to Noise Ratio (PSNR↑). These measures are intended to evaluate how much the output image matches the input image. The arrow ↑ denotes the higher the value the higher the quality, i.e. how similar the images are. In contrast, ↓ means the lower the value, the better the quality of the output image.

Let an image $M \times N$, where M is the number of rows and N the number of columns. Besides, $P(i, j)$ represents a pixel of the original image and $\hat{P}(i, j)$ represents resized modified image pixel.

These measures are defined below:

$$AD = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \left(P(i, j) - \hat{P}(i, j) \right), \quad (1)$$

$$SC = \frac{\sum_{i=1}^M \sum_{j=1}^N (P(i, j))^2}{\sum_{i=1}^M \sum_{j=1}^N (\hat{P}(i, j))^2}, \quad (2)$$

$$NK = \frac{\sum_{i=1}^M \sum_{j=1}^N (P(i, j) \times \hat{P}(i, j))}{\sum_{i=1}^M \sum_{j=1}^N (P(i, j))^2}, \quad (3)$$

$$MD = \text{Max}|P(i, j) - \hat{P}(i, j)|, \text{ for } i \in \{1, 2, \dots, M\} \text{ and } j \in \{1, 2, \dots, N\}. \quad (4)$$

$$NAE = \frac{\sum_{i=1}^M \sum_{j=1}^N |O((P(i, j)) - O(\hat{P}(i, j)))|}{\sum_{i=1}^M \sum_{j=1}^N |O(P(i, j))|}, \quad (5)$$

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (P(i, j) - \hat{P}(i, j))^2, \quad (6)$$

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{\sqrt{MSE}}, \quad (7)$$

When the quality measures are computed, hypothesis tests are applied for a statistical analysis of the results. In this work, non-parametric tests were applied, since there is no guarantee of data normality and homogeneity. The non-parametric Friedman test (HODGES; LEHMANN, 1962) is applied to point statistical differences between a group of results, that is, between the aggregation functions used.

After confirming the existence of differences between groups, a post hoc test is applied to verify between which pair of groups these differences exist. For this, Wilcoxon's non-parametric paired test was used (WILCOXON, 1945). The significance level considered for the hypothesis tests was 0.05.

The research second phase is performed applying the function that best stood out in the first phase, in this case the standard Choquet integral. After analyzing the first results only with the learning of the fuzzy power measure (which was performed by the network itself) we observed that instead of using the this old application we could adjust the network by putting a method where it learns a measure itself, in this case the capacity-like function.

To apply the function we use a classification network called Cifar-Net (KRIZHEVSKY; HINTON, 2009). This network classifies several images into ten classes. The DLN architecture used has two pooling layers, where it learns the weights (capacities) and when perform the backpropagation method corrects the weights making the output better.

Finally, the ouput results of the network using fusion functions inspired on Choquet integral are compared with the same network using the maximum function.

1.4 Organization of text

The present work is organized as follows: in Chapter 2 the theoretical reference on the functions of aggregation and pre-aggregation with the necessary definitions and Deep Learning Networks inside the current context for the understanding of the theme is presented. In Chapter 3 presents the development of the work. Chapter 4 presents the work schedule related to the specific objectives. Chapters 5 and 6 present the conclusion and the published paper as well as the award received by the paper. Chapter 7 presents the annex of the results clipping of the pre-aggregation functions combinations in the context of image resizing.

2 PRELIMINARY CONCEPTS

This chapter presents concepts essential to the understanding of work.

2.1 Generalizations of the Choquet integral

Aggregation is a process of combining different numeric values returning a single value. The operator that performs this task is called an aggregation function (GRABISCH et al., 2009).

Definition 2.1.1. A function $A : [0, 1]^n \rightarrow [0, 1]$ is said to be an n -ary aggregation function whenever the following conditions hold:

(A1) *Boundary Conditions:* $A(0, \dots, 0) = 0$ and $A(1, \dots, 1) = 1$;

(A2) *Monotonicity:* A is non-decreasing in each argument: $A(x_1, \dots, x_n) \leq A(y_1, \dots, y_n)$ whenever $x_i \leq y_i$, for all $i \in \{1, \dots, n\}$.

Sometimes a function need not to be increasing in all domain to perform well in applications:

Definition 2.1.2. (BUSTINCE et al., 2015) Let $\vec{r} = (r_1, \dots, r_n)$ be a real n -dimensional vector, $\vec{r} \neq \vec{0}$. A function $F A : [0, 1]^n \rightarrow [0, 1]$ is directionally increasing with respect to $\vec{r}$ ($\vec{r}$ -increasing, for short) if for all $(x_1, \dots, x_n) \in [0, 1]^n$ and $c > 0$ such that $(x_1 + cr_1, \dots, x_n + cr_n) \in [0, 1]^n$ it holds that $F(x_1 + cr_1, \dots, x_n + cr_n) \geq F(x_1, \dots, x_n)$. Similarly, one defines an $\vec{r}$ -decreasing function.

Definition 2.1.3. (LUCCA et al., 2016) Let $\vec{r} = (r_1, \dots, r_n)$ be a real n -dimensional vector, $\vec{r} \neq \vec{0}$. A function $PA : [0, 1]^n \rightarrow [0, 1]$ is said to be an n -ary $\vec{r}$ -pre-aggregation function if:

(PA1) F is $\vec{r}$ -increasing;

(PA2) F satisfies the boundary conditions: (A2) (i) and (ii).

Definition 2.1.4. (KLEMENT; MESIAR; PAP, 2000) An aggregation function $T : [0, 1]^2 \rightarrow [0, 1]$ is a t-norm if the following conditions hold, for all $x, y, z \in [0, 1]$:

(T1) *Commutativity:* $T(x, y) = T(y, x)$;

(T2) Associativity: $T(x, T(y, z)) = T(T(x, y), z)$;

(T3) Boundary condition: $T(x, 1) = x$.

Definition 2.1.5. (ALSINA; FRANK; SCHWEIZER, 2006) An aggregation function $C : [0, 1]^2 \rightarrow [0, 1]$ is a copula if it satisfies the following conditions, for all $x, x', y, y' \in [0, 1]$ with $x \leq x'$ and $y \leq y'$:

(C1) $C(x, y) + C(x', y') \geq C(x, y') + C(x', y)$;

(C2) $C(x, 0) = C(0, x) = 0$;

(C3) $C(x, 1) = C(1, x) = x$.

Definition 2.1.6. A function $F : [0, 1]^n \rightarrow [0, 1]$ is called idempotent if for any $x \in [0, 1]$, $F(x, x, \dots, x) = x$. F is said to be averaging, if $\min \mathbf{x} \leq F(\mathbf{x}) \leq \max \mathbf{x}$, for all $\mathbf{x} \in [0, 1]^n$.

In the context of aggregation functions, idempotency and averaging behavior are equivalent concepts. However, this is not true for pre-aggregation functions.

Some other properties may be required for aggregation functions $F : [0, 1]^2 \rightarrow [0, 1]$:

(LAE) F is said to be left 0-absorbent: $\forall y \in [0, 1] : F(0, y) = 0$;

(RNE) Right Neutral Element: $\forall x \in [0, 1] : F(x, 1) = x$.

The most common aggregation functions used in applications such as classification and image processing are the maximum (KIM, 2014; LUCCA et al., 2016), the minimum t-norm (DIMURO et al., 2018; HAMEL et al., 2011), the product t-norm (LUCCA et al., 2016), the arithmetic mean (YU et al., 2014) and copulas in general (LUCCA et al., 2017).

In DLNs aggregation functions are used in the pooling layer. The data received at the pooling layer (FUKUSHIMA, 1981) of a DLN are sub-scaled from small regions to produce a map of smaller features as input to the next level of the network. Currently, the most used pooling aggregation function are the arithmetic mean and the maximum (GOODFELLOW; BENGIO; COURVILLE, 2016). The max pooling usually presents better results when compared to the mean pooling.

The max pooling function is described as:

$$Z_j = \frac{\max\{x_j\}}{j} \quad (8)$$

An alternative (pre) aggregation functions are proposed in the first phase of this research considering some generalizations of the Choquet integral (CHOQUET, 1953–1954). As the standard Choquet integral, such generalizations are defined based on a fuzzy measure, which represents the degree of relationship between the elements to be aggregated (BELIAKOV; SOLA; SANCHEZ, 2016). In this way, the Choquet integral's significance occurs due to its model to consider the importance of each attribute to be aggregated, as well as their interactions.

Definition 2.1.7. Let $N = \{1, \dots, n\}$ and 2^N be the set of subsets of N . The function $\mathfrak{m} : 2^N \rightarrow [0, 1]$ is a fuzzy measure if, for the whole set $A, B \subseteq N$, the following conditions hold:

(m1) Boundary Conditions: $\mathfrak{m}(\emptyset) = 0$ and $\mathfrak{m}(N) = 1$;

(m2) Monotonicity: $\mathfrak{m}(A) \leq \mathfrak{m}(B)$ whenever $A \subseteq B$.

The fuzzy measure adopted in the research first phase is called the power measure (BARRENECHEA et al., 2013a), defined by:

$$\mathfrak{m}_P(A) = \left(\frac{|A|}{n} \right)^q, \text{ where } q > 0. \quad (9)$$

At the beginning of the second phase of the research when applying the standard Choquet integral function in the network architecture, the DLN itself learned the value of the exponent q of the measure, which is able to determine the best value for it according to the problem being treated.

Consequently, in the second phase of the research the fuzzy power measure is replaced by the capacity-like function, which is also learned by the network itself in its architecture. A capacity (CHOQUET, 1953–1954) is also known as fuzzy measure (SUGENO, 1974). This capacity is a normalized monotone set function (GRABISCH et al., 2016; PAP, 1995) on the decision criteria set.

Definition 2.1.8. Let $\mathfrak{m} : 2^N \rightarrow [0, 1]$ be a fuzzy measure. The discrete Choquet integral of $\mathbf{x} \in [0, 1]^n$ with respect to a fuzzy measure $\mathfrak{m}$ is a function $\mathfrak{C}_{\mathfrak{m}} : [0, 1]^n \rightarrow [0, 1]$, defined by:

$$\mathfrak{C}_{\mathfrak{m}}(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)}) \cdot \mathfrak{m}(A_{(i)}), \quad (10)$$

where $\vec{x} = (x_{(1)}, \dots, x_{(n)})$ is a non-decreasing permutation of the input $\mathbf{x}$, that is, $0 \leq x_{(1)} \leq \dots \leq x_{(n)}$, $x_{(0)} = 0$, and $A_{(i)} = \{(i), \dots, (n)\}$ is a subset of the indices of $n - i + 1$ largest components of $\vec{x}$.

The Choquet integral (10) satisfies the conditions of Def. 2.1.1, then it is an aggregation function. The Choquet integral can be written in the expanded form:

$$\mathfrak{C}_{\mathfrak{m}}(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} \cdot \mathfrak{m}(A_{(i)}) - x_{(i-1)} \cdot \mathfrak{m}(A_{(i)})) . \quad (11)$$

The Choquet integral can be generalized by copulas C , generating a family of aggregation functions called CC-integrals (LUCCA et al., 2017). This generalization is constructed by changing the operator of the product of the classic Choquet integral in its expanded form (11) by a copula C .

Definition 2.1.9. (LUCCA et al., 2017) Take a fuzzy measure $\mathfrak{m} : 2^N \rightarrow [0, 1]$ and a copula $C : [0, 1]^2 \rightarrow [0, 1]$. The Choquet integral based on the copula C and with respect

to the fuzzy measure $\mathfrak{m}$ is the function $C_{\mathfrak{m}}^C : [0, 1]^n \rightarrow [0, 1]$, called CC-integral, defined by

$$C_{\mathfrak{m}}^C(\mathbf{x}) = \sum_{i=1}^n C(x_{(i)}, \mathfrak{m}(A_{(i)})) - C(x_{(i-1)}, \mathfrak{m}(A_{(i)})). \quad (12)$$

Applying the minimum (which is a t-norm and a copula) in Eq. (12), one obtains the $C_{\min}$ -integral (DIMURO et al., 2018):

$$C_{\mathfrak{m}}^{\min}(\mathbf{x}) = \sum_{i=1}^n (\min\{x_{(i)}, \mathfrak{m}(A_{(i)})\} - \min\{x_{(i-1)}, \mathfrak{m}(A_{(i)})\}). \quad (13)$$

Definition 2.1.10. Let $\mathfrak{m} : 2^N \rightarrow [0, 1]$ be a fuzzy measure and $M : [0, 1]^2 \rightarrow [0, 1]$ be a function such that $M(x, 0) = 0$ for all $x \in [0, 1]$. In this way, the classic Choquet integral extends for the function given by:

$$\mathfrak{C}_{\mathfrak{m}}^M(\mathbf{x}) = \sum_{i=1}^n M(x_{(i)} - x_{(i-1)}, \mathfrak{m}(A_{(i)})). \quad (14)$$

Theorem 2.1.1. (LUCCA et al., 2016) Let $M : [0, 1]^2 \rightarrow [0, 1]$ be a function such that for all $x, y \in [0, 1]$ satisfies $M(x, y) \leq x$, $M(x, 1) = x$, $M(0, y) = 0$ and M is $(0, 1)$ -non-decreasing. So, for some fuzzy measure $\mathfrak{m}$, classic Choquet integral is an idempotent and averaging pre-aggregation function.

As an example, consider the Hamacher product, which is a t-norm and also a copula $T_{Ham} : [0, 1]^2 \rightarrow [0, 1]$ defined as follows:

$$T_{Ham}(x, y) = \begin{cases} 0 & \text{if } x = y = 0 \\ \frac{xy}{x + y - xy} & \text{otherwise} \end{cases} \quad (15)$$

Then, the t-norm T_{Ham} , considering the Def. 2.1.10, can be applied to Theorem 2.1.1, where T_{Ham} satisfies such conditions and we obtain the following idempotent and averaging pre-aggregation function:

$$C_{\mathfrak{m}}^{T_{Ham}}(\vec{x}) = \sum_{i=1}^n \begin{cases} 0, & \text{if } x_{(i)} = x_{(i-1)} \text{ and } \mathfrak{m}(A_{(i)}) = 0 \\ \frac{(x_{(i)} - x_{(i-1)}) \cdot \mathfrak{m}(A_{(i)})}{x_{(i)} - x_{(i-1)} + \mathfrak{m}(A_{(i)}) - (x_{(i)} - x_{(i-1)}) \cdot \mathfrak{m}(A_{(i)})}, & \text{otherwise} \end{cases} \quad (16)$$

In the following, we present the method for constructing a family of pre-aggregation functions defined by generalizing the discrete Choquet Integral using left 0-absorbent functions $\mathfrak{F} : [0, 1]^2 \rightarrow [0, 1]$, obtaining the so-called $\mathfrak{C}_{\mathfrak{F}}$ -integrals.

Definition 2.1.11. Let $F : [0, 1]^2 \rightarrow [0, 1]$ be a bivariate function and $\mathfrak{m} : 2^N \rightarrow [0, 1]$ be a fuzzy measure. The Choquet-like integral based on F with respect to $\mathfrak{m}$, called C_F -

integral is the function $C_m^F : [0, 1]^n \rightarrow [0, 1]$, defined, for all $x \in [0, 1]^n$, by

$$C_m^F(\vec{x}) = \min \left\{ 1, \sum_{i=1}^n F(x_{(i)} - x_{(i-1)}, m(A_{(i)})) \right\}. \quad (17)$$

Theorem 2.1.2. (LUCCA et al., 2018) For any fuzzy measure $m : 2^N \rightarrow [0, 1]$ and left 0-absorbent (RNE)-function $F : [0, 1]^2 \rightarrow [0, 1]$, C_m^F is a $\vec{1}$ -pre-aggregation function.

For example, using the left 0-absorbent function

$$F_{NA}(x, y) = \begin{cases} x, & \text{if } x \leq y \\ \min \left\{ \frac{x}{2}, y \right\}, & \text{otherwise} \end{cases} \quad (18)$$

we obtain the following C_F -integral

$$C_m^{F_{NA}}(\vec{x}) = \min \left\{ 1, \sum_{i=1}^n \begin{cases} x_{(i)} - x_{(i-1)}, & \text{if } x_{(i)} - x_{(i-1)} \leq m(A_{(i)}) \\ \min \left\{ \frac{x_{(i)} - x_{(i-1)}}{2}, m(A_{(i)}) \right\}, & \text{otherwise} \end{cases} \right\} \quad (19)$$

2.2 Deep Learning Networks

The area of artificial intelligence has made a great advance in recent years, generating new concepts in its expansion. Deep learning with neural networks is one of the new concepts in the AI area, highlighting in the fields of visual recognition (KARPATHY et al., 2014; KAVUKCUOGLU et al., 2010; LEE et al., 2015) and speech (HINTON et al., 2012). Due to its success with state-of-the-art results for innumerable adversities, deep learning is a much-discussed topic in the field of Computer Vision.

After made available a huge proportion of training data and a high-performance computing infrastructure, DLN has achieved a vast improvement in visual perception across multiple categories (DENG et al., 2009).

The use of aggregation functions inside the area of DLNs is not common, being more likely to be found in subjects related to Convolutional Neural Networks (CNNs) (PAGOLA et al., 2017).

The application of aggregation functions in DLNs presented acceptable results with the combination of the two methods (HAN et al., 2017).

2.2.1 Neural network architecture

There are several convolutional neural network architectures (Cifar-Net (KRIZHEVSKY; HINTON, 2009), (ROTH et al., 2016), AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) and GoogLeNet (SZEGEDY et al., 2015). To develop

the model it is necessary to understand some points of how the network architecture is performed.

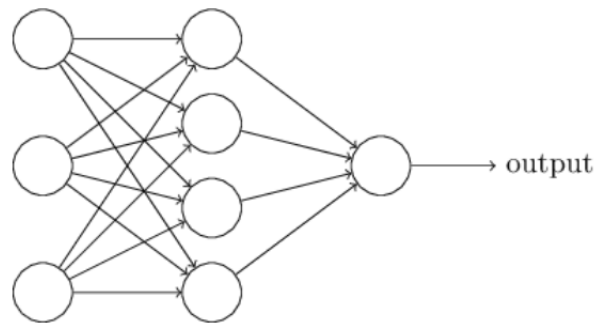


Figure 4: Simple example of a neural network architecture.

The leftmost layer in the network (Fig. 3) is called the input layer and the neurons within the layer are called input neurons. The right-most layer or the output contains the output neurons or, as in this case, a single output neuron. The middle layer is called the hidden layer, since the neurons in this layer are not inputs or outputs. The term "hidden" means a layer that is not input or output. The above network has only a single hidden layer, but some networks have multiple hidden layers. For example, the following four-layer network has two hidden layers:

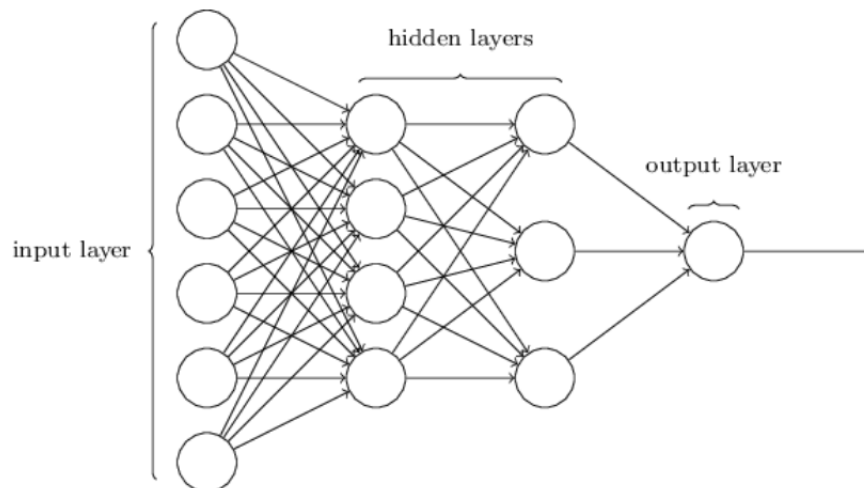


Figure 5: Example of a four-layer network has two hidden layers.

The design of input and output layers in a network is usually straightforward.

Although the design of the input and output layers of a neural network is often straightforward, there may be enough variation for the design of the hidden layers. In particular, it is not possible to summarize the design process of hidden layers with few simple rules. Instead, neural network researchers have developed many design heuristics for the hidden layers, which help people get the behavior they want from their networks. The hidden layers design is one of the crucial points in Deep Learning models.

Neural networks where the output of a layer is used as input to the next layer are called feedforward neural networks. This means that there are no loops on the network, information is always fed forward, never sent back. If we had loops, we would end up with situations where the input to the function σ (sigma) would depend on the output. This would be difficult to understand and therefore we do not allow such loops.

Feed-Forward neural networks are the most common type of neural network in practical applications. The first layer is the input and the last layer is the output. If there is more than one hidden layer, we call them Deep Learning Neural Networks (DLNs). These types of neural networks calculate a series of transformations that alter the similarities between cases. The activities of the neurons in each layer are a non-linear function of the activities in the previous layer.

2.2.1.1 Gradient descent optimizer

Most tasks in machine learning are actually optimization problems and one of the most commonly used algorithms for this is the *gradient descent algorithm*. Gradient descent is a standard tool for optimizing complex functions iteratively within a computer program. The purpose is given some arbitrary function, find a minimum. For some small subsets of functions - those that are convex - there is only a single minimum that also happens to be global. For the most realistic functions, there can be many lows, so most of the lows are local.

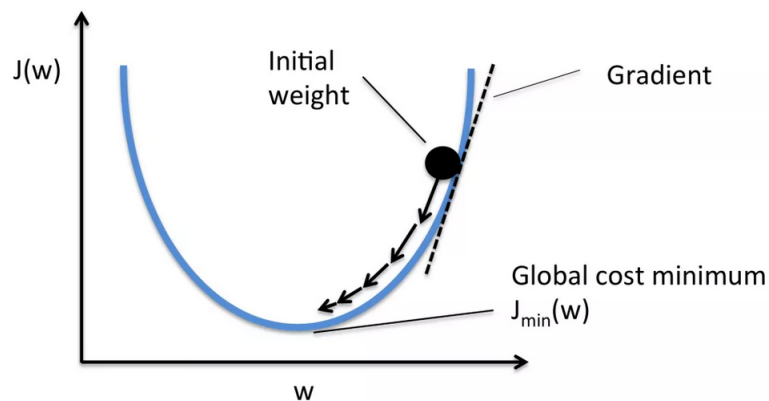


Figure 6: Representation of Gradient Descent (with the objective of minimizing the cost function).

Gradient Descent is an optimization algorithm used to find the parameter values (coefficients or if you prefer w and b - weight and bias) of a function that minimize a cost function. Gradient Descent is best used when the parameters can not be calculated analytically (for example, using linear algebra) and must be searched for by an optimization algorithm.

The procedure starts with initial values for the coefficient or coefficients of the func-

tion. These could be 0.0 or a small random value (initialization of the coefficient is a critical part of the process and several techniques can be used, with the choice being made by the Data Scientist and the problem to be solved with the model). We could thus start our coefficients (values of w and b):

$$coefficient = 0, 0 \quad (20)$$

The cost of the coefficients is evaluated by linking them to the function and calculating the cost (Eq. 21 and 22).

$$cost = f(coefficient) \quad (21)$$

$$cost = evaluate(f(coefficient)) \quad (22)$$

The derivative of the cost is calculated. The derivative is a concept from calculus and refers to the slope of the function at a given point. We need to know the slope so that we know the direction (sign) to move the coefficient values in order to get a lower cost on the next iteration.

$$\Delta = derivative(cost) \quad (23)$$

To update the coefficient values a learning rate parameter (α) must be specified that controls how much the coefficients can change on each update:

$$coefficient = coefficient - (\alpha * \Delta) \quad (24)$$

This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough.

2.2.1.2 Backpropagation method

Backpropagation is arguably the most important algorithm in the history of neural networks (without backpropagation, it would be nearly impossible to train deep learning networks in the way we see today). Backpropagation can be considered the cornerstone of modern neural networks and consequently Deep Learning.

Backpropagation is the key algorithm that makes deep modeling training something computationally treatable. For modern neural networks, it can make gradient training down to ten million times faster, compared to a naive implementation. This is the difference between a model that takes a few hours or days to train and another that could take years.

For any supervised learning problem, we select weights that provide the optimal estimate of a function that models our training data. In other words, we want to find a set

of weights W that minimizes the output of $J(W)$, where $J(W)$ is the loss function, or the network error.

The descending gradient algorithm updates each weight by some negative scalar reduction of the derivative of the error in relation to that weight. If we choose to use descending gradient (or almost any other convex optimization algorithm), we need to find the derivatives in numerical form.

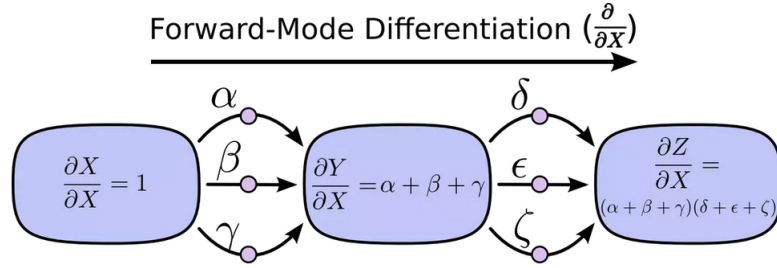


Figure 7: Representation in graph of forward-mode differentiation or forward pass.

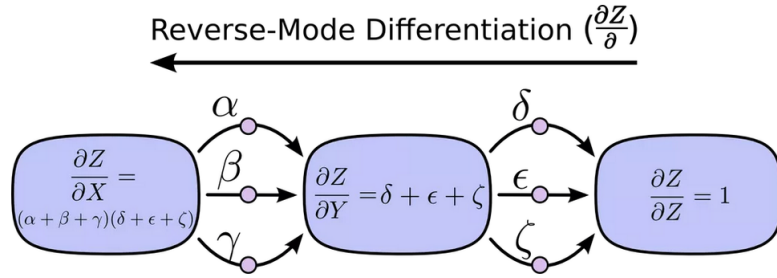


Figure 8: Representation in graph of reverse-mode differentiation (backpropagation algorithm).

2.2.2 Pooling layer

Convolutional Neural Networks are great tools for solving various problems (BOUVRIE, 2006; FUKUSHIMA, 1981; LECUN et al., 1998). The structure of a CNN is constituted by layers of convolution that have as objective the generation of maps of resources that have answers to different filters of kernel¹. Subsequently, the consequent activations of a convolutional layer are passed to the pooling layer (FUKUSHIMA, 1981), where data in small local regions are sub-magnified to produce a map of smaller features as input to the next level of the network.

The data received at the pooling layer (FUKUSHIMA, 1981) of DLN are sub-scaled from small regions to produce a map of smaller features as input to the next level of the network. The main purpose of the pooling operation is to reduce the perception of the network corresponding to small changes in the image.

¹Kernel is a small array used for blurring, sharpening, embossing, edge detection, etc. This is done by doing a convolution between a kernel and an image.

There are two ways to perform the pooling function currently: using the *arithmetic mean function* and the *maximum function*. The max pooling usually has better results when compared to the mean pooling. The max pooling function is described as:

$$Z_j = \max_j \{x_j\}$$

Considering the above function, the derivative of output Z_j related to input x_j will be 0 or 1, which depends on whether the value of x_j has been chosen or not. Simple derivatives speed network training time considerably, particularly in multi-layered neural networks, since they have a large amount of neural connections.

Figure 10 represents an example of the max pooling function using a simple matrix. The number of layers related to the image does not change after pooling, this means that the pooling regions are restricted only in the width and height dimensions of the image used.

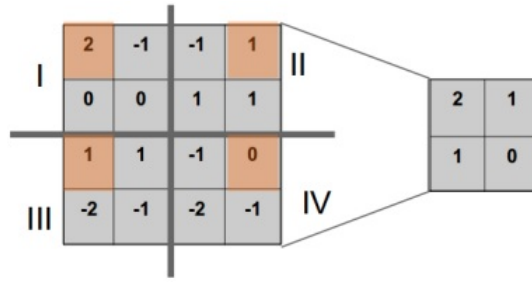


Figure 9: Application of max pooling in a location divided into four quadrants.

The result after applying pooling to an input image partitioned into $p \times p$ sections is described in the equation below:

$$Z_{(i,j)}^c = \max_{a \in [p.i, (p+1)i], b \in [p.j, (p+1)j]} \{x_{a,b}^c \forall c, i, j\} \quad (25)$$

The variable c is the index of the color channel of the image (RGB channel), and the variables i and j are the indexes of the partitions.

The *maximum function* draws the most important features (figure 10) and captures the strongest activation, disregarding all other values in the pooling area. Another function is the *arithmetic mean*. This function takes into account all activations (figure 11) in a pool area with equal contributions (YU et al., 2014).

The choice to use the Choquet integral and its generalizations as alternative methods that can be found in the literature is motivated by the difference of how the functions behave in the interaction question of each pixel of the image. The purpose is to obtain an image with the strongest activations and with the same or close quality to the input image.

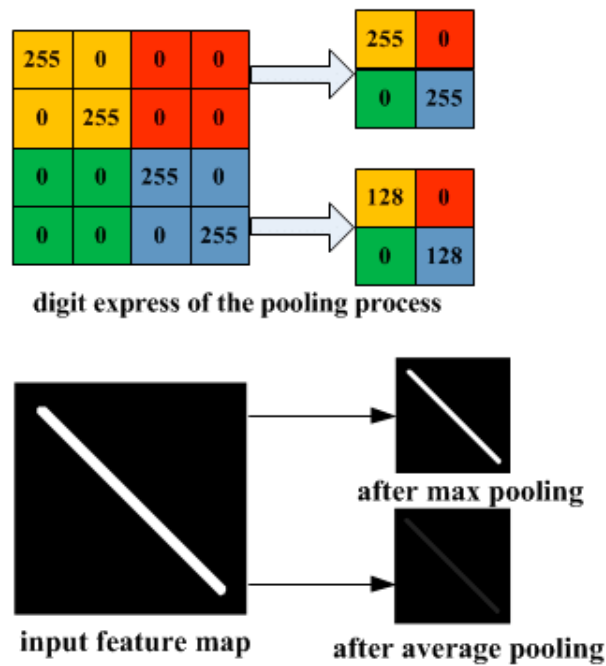


Figure 10: Illustration of a disadvantage of mean function application (YU et al., 2014).

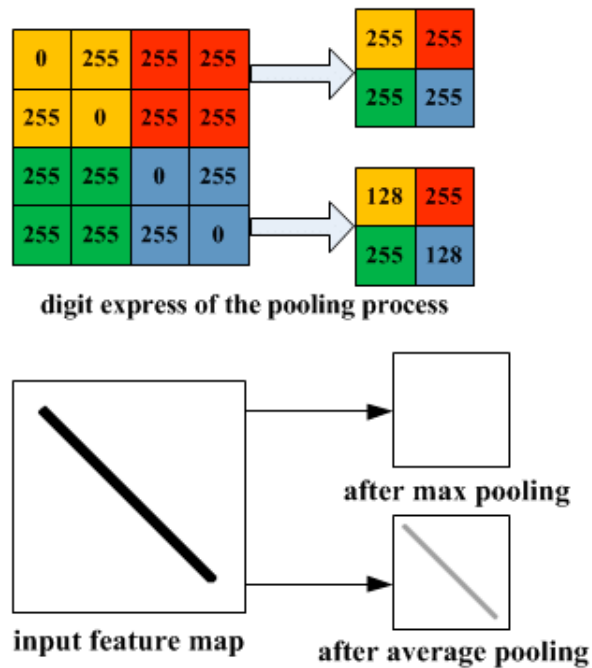


Figure 11: Illustration of a disadvantage of maximum function application (YU et al., 2014).

2.3 Neural networks in the context of image processing

In this section a brief demonstration of how image processing tasks (Fig. 11) are performed in DLN architecture is presented.

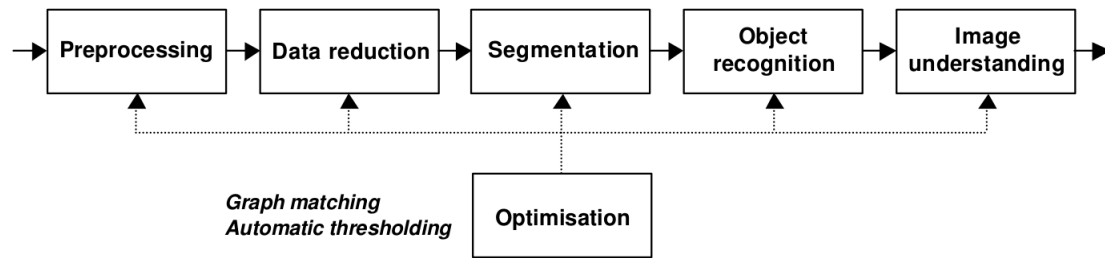


Figure 12: Illustration of image processing chain containing the different tasks.

2.3.1 Preprocessing

The first phase in the image processing consists of preprocessing the input image. The input consists of sensor data and of which output is a full image. Performing pre-processing operations, one of three categories are applied: image reconstruction, image restoration and image enhancement.

2.3.2 Data reduction

In the second step of image processing an image compression algorithm is applied. For subsequent segmentation or object recognition task a feature extraction is used. The type of method for extraction often corresponds to particular geometric or perceptual characteristics in an image (edges, corners, and joints) (EGMONT-PETERSEN; RIDDER; HANDELS, 2002).

2.3.3 Image segmentation

The segmentation step consists in partition an image into several parts that are in accordance with some coherent criterion. The literature on RGB image segmentation is not that rich as it is for grayscale images (PAL; PAL, 1993). The principal objective of segmentation is to assign labels to individual pixels.

2.3.4 Object recognition

The purpose of object recognition is to locate the positions, instances scales and orientations of some objects in an image, to assign a class label to the detected object.

2.3.5 Image understanding

A complicated area in image processing is the image understanding. This phase performs merging targeting object recognition techniques with knowledge of the expected image content.

2.3.6 Optimization

Tasks like stereo-matching and graph can be better formulated as optimization issues. This is a subtask in image processing.

3 USING THE CHOQUET INTEGRAL IN THE POOLING LAYER IN DEEP LEARNING NETWORKS

In order to improve the aggregation of significant information without degrading the image processing, it was proposed to replace the max pooling function by Choquet integral and its generalizations (DIAS et al., 2018). These functions will have as their domain the pixel values of an input image, ranging from 0 to 255. In addition, to facilitate analysis of results, the input image is converted from RGB to grayscale, reducing the number of layers from 3 to 1.

Inside the classic Choquet integral and its generalizations code, the stride and the size of the window are defined by the programmer in the code. For example: if the size of the window is defined as 2x2 and the parameter stride with value 2, the function will be executed generating several 2x2 matrices and 2 pixels will be skipped between each window, limited to the total size of the input image. That is, an input image with a total size of 500x500 pixels will output a 250x250 pixels image.

The following steps are directly related to the execution of the Choquet integral and its other generalized functions. Firstly, to perform the arithmetic function, it is necessary to transform the windows (which are matrices) into vectors. After this transformation, the vector permutation is performed, where the values of the vector will be ordered in pairs, together with the index of each value. After relating each value to its index, the organization of the vector is performed in an increasing way, because to perform the pre-aggregate functions it is necessary to organize the vector in a non-decreasing way. After this step, a vector will store the values without the indexes, processing the function, described in equations 10, 12, 13 and 16.

A fuzzy measure is applied in the described functions, which is calculated previously. The fuzzy measure used in this work is denominated as power fuzzy measure (Def. 2.1.7), where $N = \{1, \dots, n\}$, $A \subseteq N$ and $|A|$ is the cardinality of the set A . The value of the exponent of the fuzzy measure (q) is usually calculated through another optimization method, such as genetic algorithms, which will look for the best value for the fuzzy measure (BARRENECHEA et al., 2013b). In the case of this work, the value of q is chosen by a specialist user as a parameter of the proposed method.

In the measurement of image quality, which is very important for image processing systems, different image quality measures were used, which are calculated for each image resulting from the aggregation and pre-aggregation functions. Identifying image quality measures that are most sensitive to applications helps to systematically design coding, communication and image systems, as well as improving or optimizing image quality for a desired application quality at minimal cost.

In recent years, many efforts have been made to develop objective image quality metrics that correlate with perceived quality. Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are some objective measures and objective image quality most used (C.SASI VARNAN; MULLANA, 2011).

The measures used to measure the quality of the output images of this work are defined in equations 1, 2, 3, 4, 5, 6 and 7.

In order to compare the maximum, arithmetic mean, classic Choquet integral (Def. 2.1.8) and its generalizations: Average-CF (Def. 2.1.9), TM (Def. 2.1.10) and Hamacher (Def. 15), 12 images from the IIIT 5K-Word data set were used (MISHRA; ALAHARI; JAWAHAR, 2012).

For each of these 12 images, experiments were performed considering the parameters: window size (2x2, 3x3 and 4x4) and stride (2 and 3). Using different window size and stride allows to capture interaction characteristics between multiple windows. In addition, the parameter q of the fuzzy measure was varied, chosen by a specialist, applying the values 0.1, 0.3, 0.5 and 0.7.

Thus, each input image generate 24 output images, one for each distinct combination of the window, stride and q . Therefore, 288 images were generated for each of the four aggregation functions mentioned above.

Table 2 presents the average results obtained from the image quality measures for each aggregation function according to the best parameters: stride, window size and fuzzy power measure exponent (with the values 0.1 and 0.7).

As it can be seen in Table 2 the best results for AD, MD and MSE were mean, maximum and mean in this order. The second best results for the same image quality measures were classic Choquet integral (Eq. 10), TM (Eq. 13), Hamacher (Eq. 16) and CF (Eq. 19). In the case of MSE results the classic Choquet integral exceeded the maximum function, which is the most used in CNNs applications. In measurements of image quality NAE and NK the best results were arithmetic mean and classic Choquet integral and secondly the best results were the classic Choquet integral and the generalization that consider the Hamacher t-norm. In the case of the NAE measurement it takes the same comparison of the MSE measurement. Finally, in the last two functions used, PSNR and SC, the best results were arithmetic mean and in the case of SC image quality measure Choquet integral classic and Hamacher obtained the lowest results, standing out from the others.

Table 2: Average results obtained from the image quality measures for each aggregation function. The values in bold are the best results for the measurement indicated in table.

Quality measure	Choquet	Hamacher	TM	Average-CF	Max	Mean
AD ($q = 0.1$)	-28.19	-28.25	-18.54	-25.87	-28.17	1.49
↓ MD ($q = 0.1$)	150.69	149.35	188.47	162.94	134.31	182.67
↓ MSE ($q = 0.7$)	2390	3421	2581	3865	4104	1494
↓ NAE ($q = 0.7$)	0.31	0.35	0.34	0.37	0.35	0.20
↑ NK ($q = 0.1$)	1.14	1.13	1.06	1.12	1.12	0.91
↑ PSNR ($q = 0.7$)	14.95	13.71	14.53	13.13	13.23	17.22
↓ SC ($q = 0.1$)	0.70	0.70	0.83	0.72	0.72	1.10

3.1 Analyzes and results

A statistical analysis was performed on the means (Table 2) to verify if there is statistically difference between the results.

As it can be seen in Table 2 the best results for AD, MD and MSE were mean, maximum and mean in this order. The second best results for the same image quality measures were classic Choquet integral Eq. 10, TM Eq. 13, Hamacher Eq. 16 and CF Eq. 19. Statistically they did not tie. In the case of MSE results the classic Choquet integral exceeded the maximum function, which is the most used in CNNs applications. In measurements of image quality NAE and NK the best results were arithmetic mean and classic Choquet integral and secondly the best results were the classic Choquet integral and the generalization that consider the Hamacher t-norm. In the case of the NAE measurement it takes the same comparison of the MSE measurement.

There was no statistically significant difference among the classic Choquet integral, Hamacher and the Maximum functions. Finally, in the last two functions used, PSNR in dB and SC, the best results were arithmetic mean and in the case of SC image quality measure Choquet integral classic and Hamacher obtained the lowest results, standing out from the others. In the case of the PSNR measure there is a statistical difference and in SC's case the classic Choquet integral and Hamacher functions are equal statistically, jointly with the maximum function.

Although there is a technical tie in the statistical analyzes, the functions derived from the Choquet integral presented the best visual details in the experimental results (the Choquet integral function in figure 4 and the TM function in figure 5). It can be observed in the resulting images that the standard Choquet integral and its considered generalizations are more representative to identify the object (in this case the words described in the image), leaving the image less dirty, blurry or serrated.

3.2 Concluding remark of the first phase

A preliminary study of the implementation of the Choquet integral and its generalizations in the DLN pooling layer was presented, demonstrating the analysis performed without the direct application of the function in the network. The results of the application of the Choquet integral and its generalizations to reduce the image are shown to be promising, when compared the results of the usual aggregation functions.

The results presented show to be prosperous tending to refine the studies. The parameter called exponent of the fuzzy measure used in Choquet integral and its generalizations, in relation to the power fuzzy measure can be learned by the DLN itself, generating results superior to those demonstrated in this work.

Figure 13: Results of max, mean, Choquet integral and its generalizations obtained through the experiments applied to one of the images of the IIIT 5K-Word data set called 138_6, without resize. The parameters used were: window size = 4x4, stride = 2 and power fuzzy measure exponent = 0.7

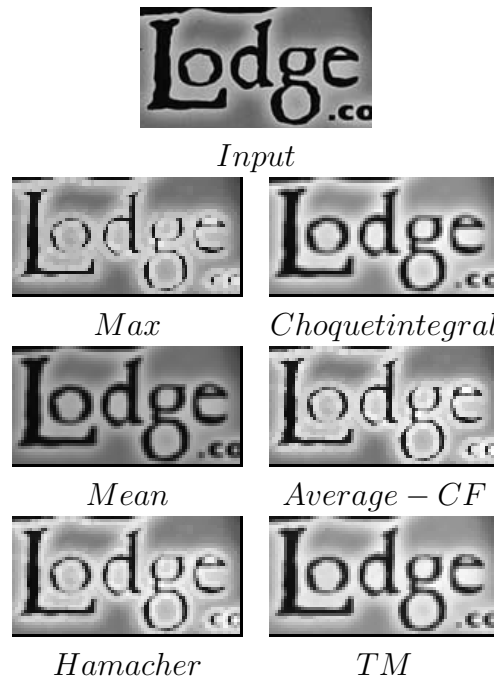
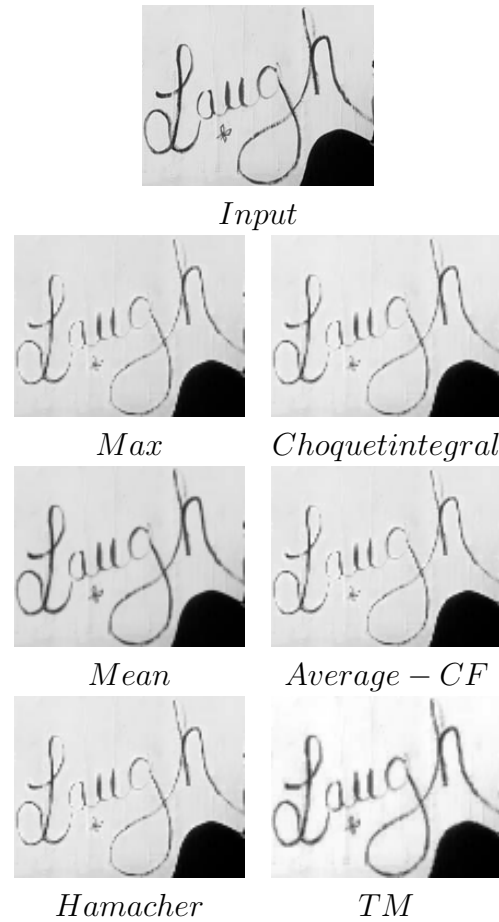


Figure 14: Results of max, mean, Choquet integral and its generalizations obtained through the experiments applied to one of the images of the IIIT 5K-Word data set called 127_3, without resize. The parameters used were: window size = 2x2, stride = 2 and power fuzzy measure exponent = 0.1



Previous studies have been with all the functions previously developed for image classification. In this specific study the application was set inside the resizing of the image using different functions, measures and parameters. The best results are shown below in table 3 and a snippet with the other experiments are in annex 1.

Table 3: Best image quality measurement results from the combination of two pre-aggregation functions in the resizing of images for classification

Quality measure	Function 1	Function 2	Fuzzy measure	Window	Stride	Image label	Best result
AD	CTM	prod	power-1-0000	2x2	2x2	14	0,000114
↓MD	CTM	AVG	power-1-0000	2x2	2x2	131	-0,04642
↓MSE	CTM	prod	power-1-0000	2x2	2x2	14	0,003145
↓NAE	CTM	CF	power-1-0000	2x2	2x2	105	0,032744
↑NK	CTM	AVG	power-1-0000	2x2	2x2	131	1,210775
↑PSNR	CTM	prod	power-1-0000	2x2	2x2	14	73,15506
↓SC	CTM	AVG	power-1-0000	2x2	2x2	131	0,669935

4 IMAGE CLASSIFICATION USING A POOLING FUNCTION BASED ON THE DISCRETE CHOQUET INTEGRAL WITH THE LEARNING OF NON MONOTONIC FUZZY MEASURE

CifarNet, introduced in *Learning multiple layers of features from tiny images* (KRIZHEVSKY; HINTON, 2009), was the state-of-the-art model for object recognition on the CIFAR-10 dataset, which consists of 32x32 images of 10 object classes. The objects are normally centered in the images. The example images and class categories from the Cifar10 dataset are shown in Fig. 15.

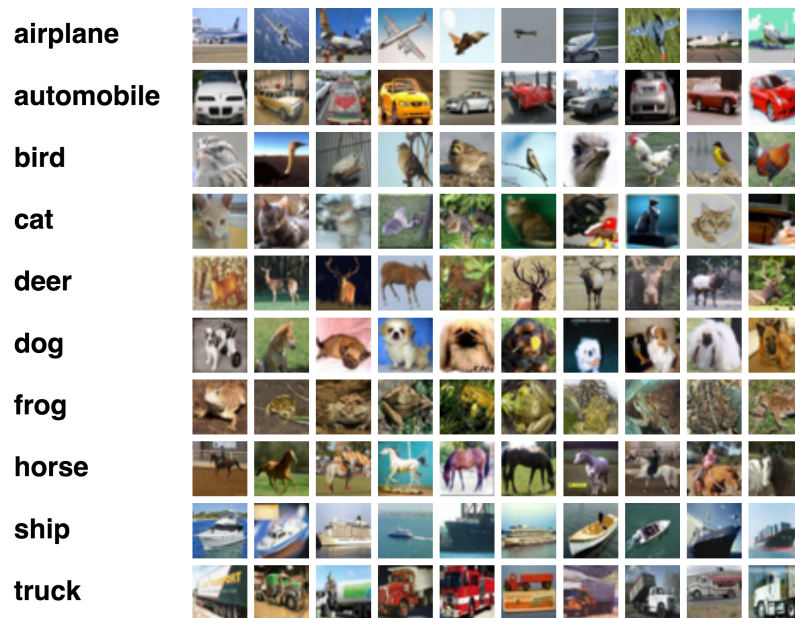


Figure 15: Dataset used in CifarNet.

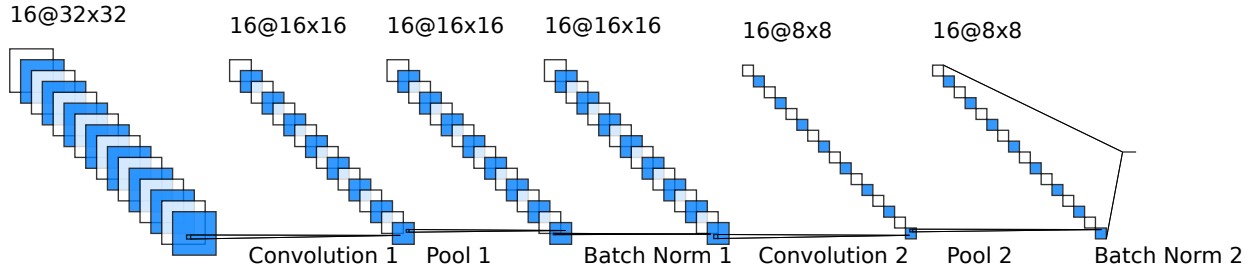


Figure 16: CifarNet Architecture used in this research.

4.1 Theoretical developments

In this section, we introduce some theoretical developments concerning defining the concept of discrete Choquet integral with non-monotonic fuzzy measures $\mu : \wp(N) \rightarrow [0, 1]$ such that $\mu m(N) = 1$ is not required for $N = \{1, \dots, n\}$.

4.1.1 Motivation for working with non-monotonic fuzzy measures

Murofushi et al. (MUROFUSHI; SUGENO; MACHIDA, 1994) discussed the importance of non-monotonic fuzzy measures and the Choquet integral with respect to non-monotonic fuzzy measures, given a complete characterization of such integrals.

Observe that fuzzy measure in the sense of Sugeno (SUGENO; MUROFUSHI, 1987) is a monotonic set function. Sugeno introduced this concept as an extension of the probability measure, that is, the fuzzy measure $m(A)$ of a subset A of the whole set X expresses the degree of belief/likeness/confidence of $x_0 \in A$, where x_0 is an unknown element of X . Then, according to this idea, the monotonicity of the fuzzy measure is essential: whenever $A \subset B$, then $x_0 \in A$ implies $x_0 \in B$, and, thus the degree of $x_0 \in A$ is less than or equal to that of $x_0 \in B$.

However, one can provide a different interpretation of the fuzzy measure, where instead, the monotonicity is inessential. For that, consider the abstract interpretation given by Murofushi and Sugeno (MUROFUSHI; SUGENO, 1989): "the non-additivity of the fuzzy measure expresses interaction among subsets". Now, let m be a fuzzy measure and A and B disjoint subsets. According to this interpretation, the inequality $m(A \cup B) > m(A) + m(B)$ shows a cooperative action or synergy between A and B , and the opposite inequality $m(A \cup B) < m(A) + m(B)$ shows a tendency of offset or incompatibility between A and B . Therefore, if the tendency of offset is very strong, there may be a case where $m(A \cup B) < m(A)$ and/or $m(A \cup B) < m(B)$, which is a violation of the monotonicity.

Murofushi et al. (MUROFUSHI; SUGENO; MACHIDA, 1994) also argued that, from the mathematical view point, the monotonicity is not necessary, since one can construct measure theory without monotonicity, showing that the Choquet integral with respect to non-monotonic fuzzy measures can be defined. See (MUROFUSHI; SUGENO; MACHIDA, 1994, Section 2). For other studies concerning non-monotonic fuzzy measures,

see also (RÉBILLÉ, 2005; AUMANN; SHAPLEY, 2015).

However, Murofushi et al. (MUROFUSHI; SUGENO; MACHIDA, 1994) worked with real-valued set function defining the non-monotonic fuzzy measure and the Choquet integral.

Then, in order to adapt such definitions to our context, we define the non-monotonic fuzzy measure $\mu : \wp(N) \rightarrow [0, 1]$ such that $\mu(N) = 1$ is not required for any $N = \{1, \dots, n\}$ and the concept of discrete Choquet integral concerning such fuzzy measures.

4.1.2 The non-monotonic discrete Choquet integral

Based on (MUROFUSHI; SUGENO; MACHIDA, 1994, Definition 2.2), we define, for $N = 1, \dots, n$:

Definition 4.1.1. A function $\mu : 2^N \rightarrow [0, 1]$ is said to be a non-monotonic fuzzy measure if $\mu(\emptyset) = 0$.

Definition 4.1.2. The discrete Choquet integral with respect to a non-monotonic fuzzy measure μ is the function $\mathfrak{C}_\mu : [0, 1]^n \rightarrow [0, 1]$, defined, for all of $\vec{x} = (x_1, \dots, x_n) \in [0, 1]^n$, by:

$$\mathfrak{C}_\mu(\vec{x}) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)}) \cdot \mu(A_{(i)}), \quad (26)$$

where $(x_{(1)}, \dots, x_{(n)})$ is an increasing permutation on the input $\vec{x}$, that is, $0 \leq x_{(1)} \leq \dots \leq x_{(n)}$, where $x_{(0)} = 0$ and $A_{(i)} = \{(i), \dots, (n)\}$ is the subset of indices corresponding to the $n - i + 1$ largest components of $\vec{x}$.

4.1.3 Modeling the inputs of the non-monotonic discrete Choquet integral in the learning process of the non-monotonic fuzzy measure

The fuzzy measure $\mu(A_{(i)})$, for any $A_{(i)} \neq \emptyset$ of Def. 4.1.2, is to be learned using an iterative algorithm, based on backpropagation process of the DLN.

Then, we started with an initial vector for the fuzzy measure values, namely: $\vec{W}^0 = (w_1^0, \dots, w_n^0)^T$, where $\mu(A_{(i)}) = w_i$, for simplicity, and T indicates that we are considering the transpose vector.

After the first iteration using backpropagation, the obtained values, denoted by $bp(\vec{W}^0) = (bp(w_1^0), \dots, bp(w_n^0))^T$, may not be in the unit interval $[0, 1]$. Then, for the next iteration, we proceed to their normalization in two steps by:

1. Transforming the values into non-negative values:

$$\vec{W}^1 = (e^{bp(w_1^0)}, \dots, e^{bp(w_n^0)})^T \quad (27)$$

2. Obtaining values less or equal to 1:

$$\vec{W}^{1*} = \frac{\overline{W^1}}{\max\{e^{bp(w_1^0)}, \dots, e^{bp(w_n^0)}\}}. \quad (28)$$

In fact, the new values of the fuzzy measure of the non-monotonic Choquet integral in any iteration i are calculated by:

$$\overline{W^i} = (e^{bp(w_1^{i-1})}, \dots, e^{bp(w_n^{i-1})})^T \quad (29)$$

$$\vec{W}^{i*} = \frac{\overline{W^i}}{\max\{e^{bp(w_1^{i-1})}, \dots, e^{bp(w_n^{i-1})}\}}. \quad (30)$$

The discrete non-monotonic Choquet integral for the pooling function, Pooling-Ch, in any iteration i , is then calculate in a matrix notation by:

$$Pooling - Ch^i(\vec{x}) = \vec{W}^{i*} \begin{pmatrix} x_{(n)} - x_{(n-1)} \\ \vdots \\ x_{(2)} - x_{(1)} \\ x_{(1)} \end{pmatrix} \quad (31)$$

for all $\vec{x} = (x_1, \dots, x_n) \in [0, 1]^n$, where $(x_{(1)}, \dots, x_{(n)})$ is an increasing permutation on the input $\vec{x}$, that is, $0 \leq x_{(1)} \leq \dots \leq x_{(n)}$, where $x_{(0)} = 0$.

Given 4 values in a 2×2 pooling window, in this dissertation we use the following expression:

$$Pooling - Ch^i(\vec{x}) = \vec{W}^{i*} \begin{pmatrix} x_{(4)} - x_{(3)} \\ x_{(3)} - x_{(2)} \\ x_{(2)} - x_{(1)} \\ x_{(1)} \end{pmatrix} = \vec{W}^{i*} \begin{pmatrix} A_4 - A_3 \\ A_3 - A_2 \\ A_2 - A_1 \\ A_1 \end{pmatrix} \quad (32)$$

The W vector learning is proceed using the backpropagation algorithm in the same fashion as the rest of the parameters in the network. We compare this new pooling using the architecture in Fig. 16, against an analogous net using the max-pooling.

A different number of architectures is necessary to train, using two convolution plus pooling layers. These architectures are smaller than the state-of-the-art networks, but are enough to study the results with our new function. We compare our results against the well-known max-pooling implement in all the deep learning libraries.

4.2 W-evolution

The first thing to notice is the difference between the two layers during training. While first shows a clear tendency to make W_4 as big as possible (Fig. 17), the other one, while making also W_4 the biggest one, does not suppress the other weights in the expression Eq. (30) (Fig. 18).

The difference between both layers is a bit of a mystery, but could possibly be related to the idea that each layer has its own optimal fuzzy measure. This may leave some room for improvement in mixed-pooling networks (using different aggregation functions in different layers), as it seems that there is no optimal pooling-function for all layers. The conditions under this could be a problem, remains unexplored.

When looking at the gradients, it seems they have converged, as they tend to decrease alongside the number of steps. It remains to be seen the learning of the measure is affected by the number of layers and how it evolves when the number of epochs is > 100 .

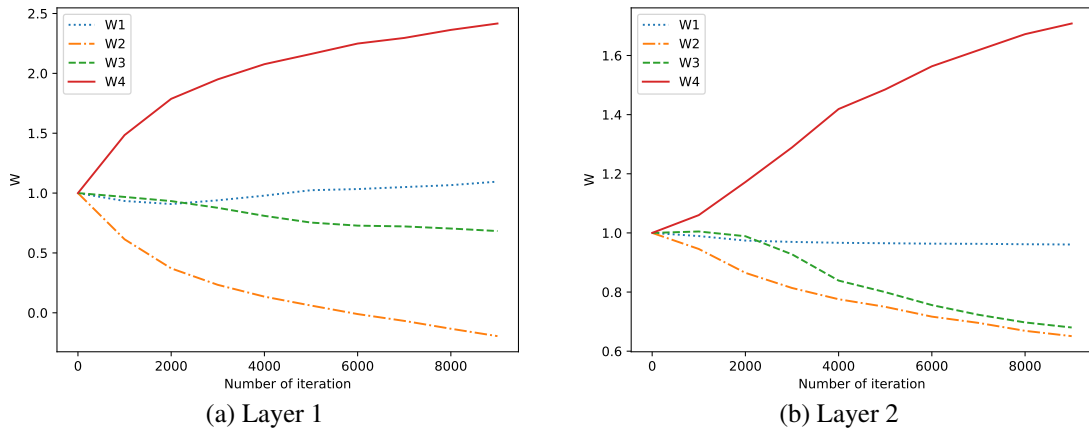


Figure 17: Pool W exponents for layer 1 and 2

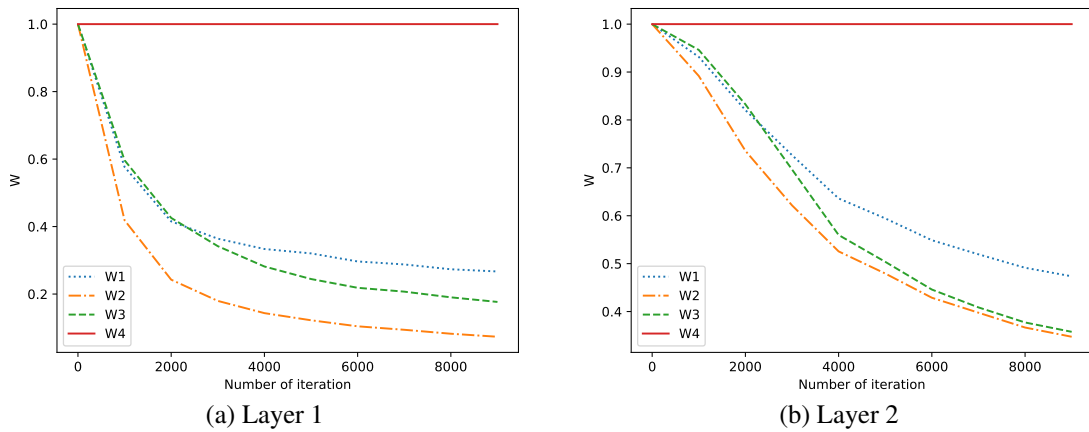


Figure 18: Pool W for layer 1 and 2

4.2.1 Comparison Non-Monotonic Choquet integral pooling against Max-pooling in training

This subsection presents how the non-monotonic Choquet integral (NMCI) pooling and the max-pooling progress during 10000 steps of training.

When observing at the evolution of the cross-entropy loss (Fig. 19), the both networks learn at a very similar rate. If we calculate the differences between each loss in every step, we get a normal population (Figure 20), with $\bar{x} = 0.03125$ and $s = 0.14765$. If we perform the Shapiro-Wilk normality test, we obtain the p-value: 0.17, meaning that this distribution is correctly identified as a normal. This means that the times where non-monotonic Choquet integral pooling loss is bigger than max-pooling loss and vice versa can be explained by the small fluctuations in the training process, making both statistically equivalents.

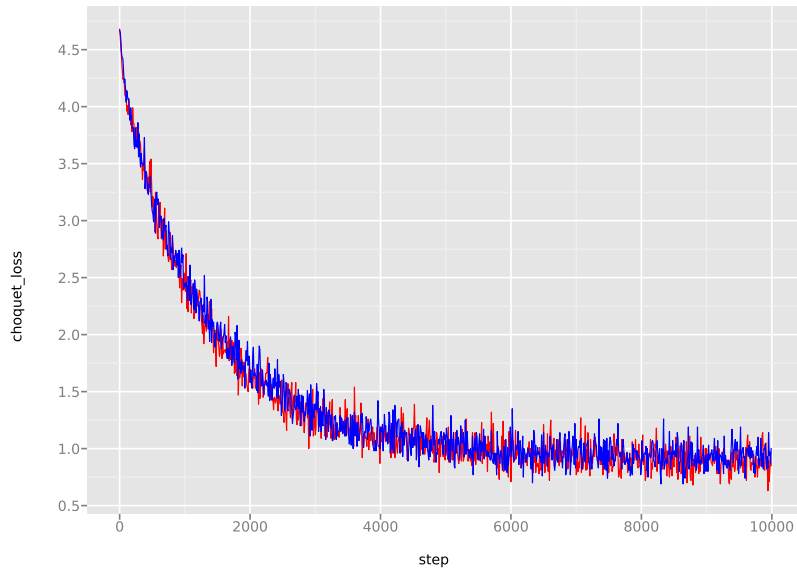
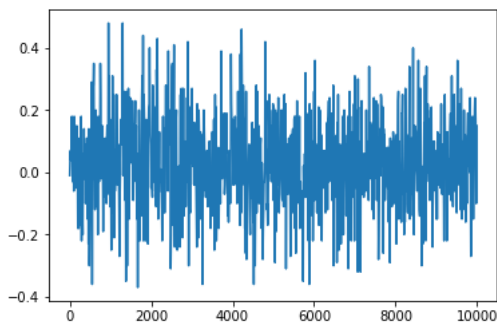
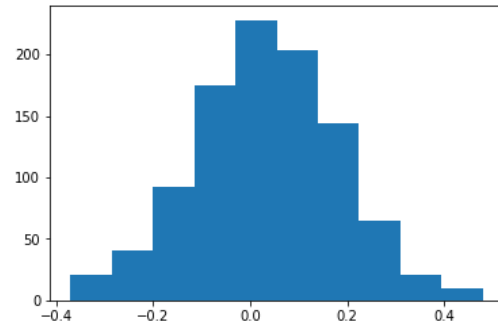


Figure 19: Losses during steps from NMCI and Max. (Max=red, NMCI=blue)



(a) Time series for NMCI losses - Max losses.



(b) Histogram for NMCI losses - Max losses (Mean 0.03125, Std 0.14765).

Figure 20: Difference in losses for each pooling.

4.2.2 Behaviour of the learnt weights in the 2×2 pool window

Based on the weights learnt in the previous section, and showed in Figure 18, we study the behaviour of each pooling layer according to different situations in Table 21.

Situation	Most relevant term	Pool Layer 1	Pool Layer 2
All four numbers are approximately the same	$W_1 A_1$	≈ 0	$\approx 0.6 A_1$
There is one much bigger than the others	$W_4 (A_4 - A_3)$	$\approx A_4$	$\approx A_4$
There are two much bigger than the others	$W_3 (A_3 - A_2)$	≈ 0	$\approx 0.8 A_3$
There are three much bigger than the forth	$W_2 (A_2 - A_1)$	≈ 0	$\approx 0.78 A_2$

Figure 21: Table for different situations of the pooling window and the approximate result of the pooling layer.

The *pool1* works as some kind of "big difference detector" while the second one is more closely related to a typical weighted average of the differences.

In the first case, the four elements (A_1, A_2, A_3 and A_4) in the pooling window are practically the same. Observing the Table 21, the values $A_4 - A_3 \approx 0$ lead us to conclude that only the term A_1 is relevant, what is going to determine the final result.

The first pooling layer will obtain the result nearly to zero because $W_0 \approx 0$, where it is represented in figure 17 (a). In figure 17 (b), which represents the second execution of pooling in the network, the result will be approximately 0,6.

At the end it is possible to conclude that the first execution of the pooling layer as a sort of detector of great differences, while the second layer is related to a typical weighted average of the differences. Lastly, the second measure is increasing ($W_4 > W_3 > \dots$) while the second measure is not.

The topology of the whole network shows the changes in the convolution parameters are significant. Comparing them to an equivalent network (same architecture and training process) we observed that biases are bigger than in the conventional max pooling network and that convolutional weights are smaller in both convolution layer 1 and 2 (Fig. 22 and 23)

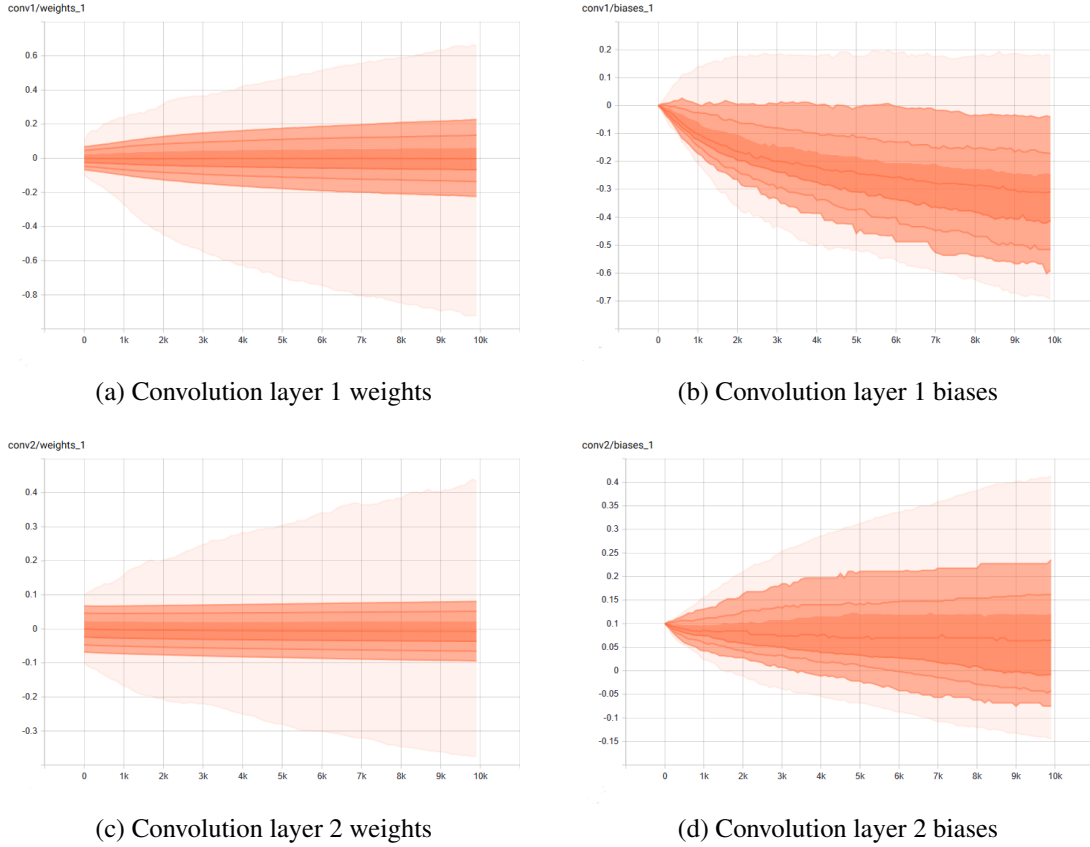


Figure 22: Convolution weights and biases for the max-pooling network.

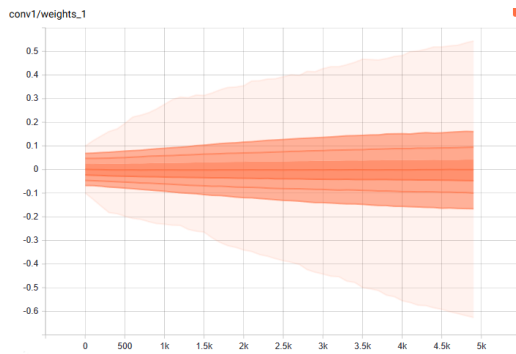
After obtaining the results we have also compared the results obtained by the traditional max-pooling function and the new Choquet integral pooling function for a random sample of size 1000 in the $[0, 1]^4$ space. The max-pooling shows a bigger value (around 0.5) than the Choquet integral pooling (Fig. 26).

Finally, when we compare results using both pooling functions (10000 iterations), we obtain similar results. This could be due to the extra parameters to train.

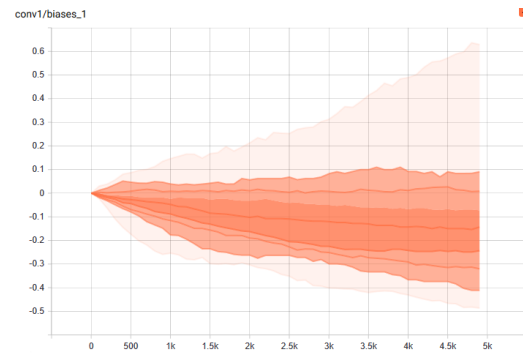
4.2.3 Final results of the classification

The comparison of final accuracy results can be observed in Table 4. It is noted that the fusion function inspired in standard Choquet integral, defined by a capacity-like function which is learned by the own model, obtained close results comparing with max-pooling accuracy results.

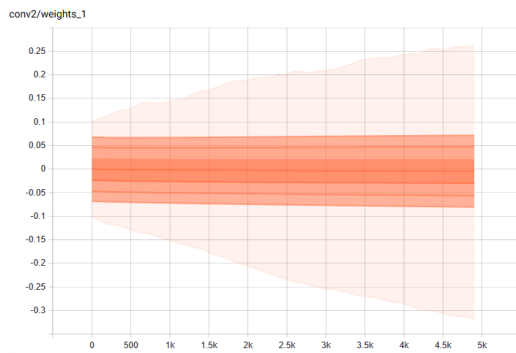
Thus, this research proves satisfactory since even with more parameters than the maximum pooling function, the new function presents acceptable results, which can be improved in the future.



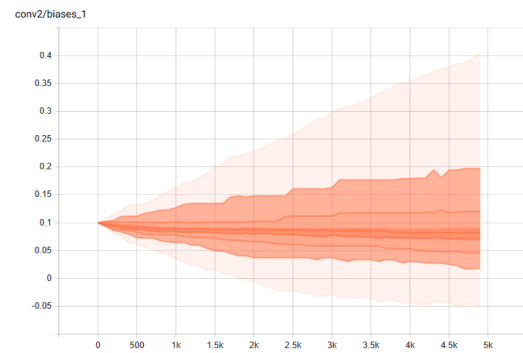
(a) Convolution layer 1 weights



(b) Convolution layer 1 biases

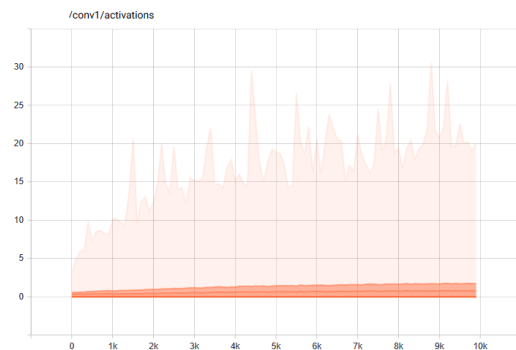


(c) Convolution layer 1 weights

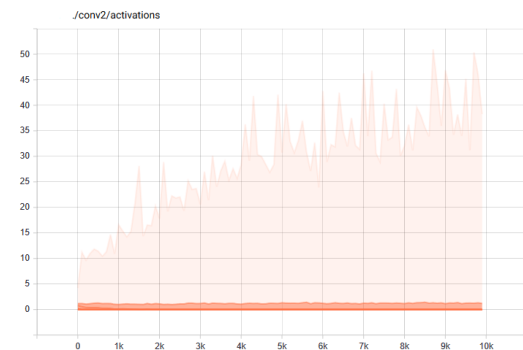


(d) Convolution layer 1 biases

Figure 23: Convolution weights and biases for the choquet-pooling network. network.

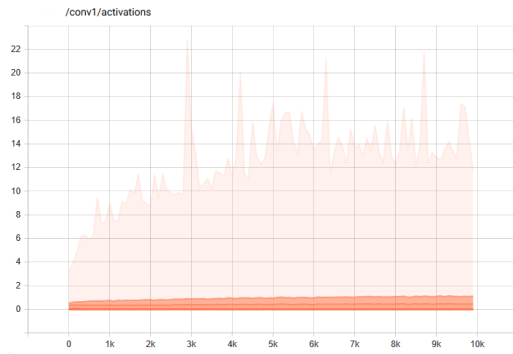


(a) Convolution layer 1 activations

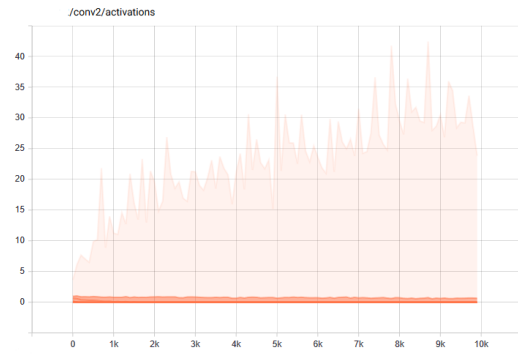


(b) Convolution layer 2 activations

Figure 24: Activations for each convolution in a choquet-pooling network.

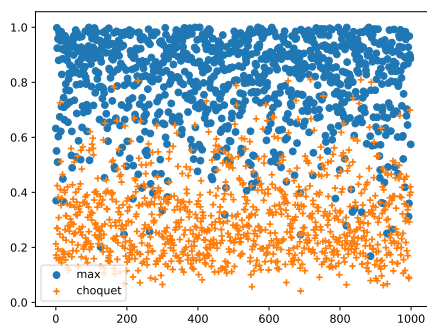


(a) Convolution layer 1 activations

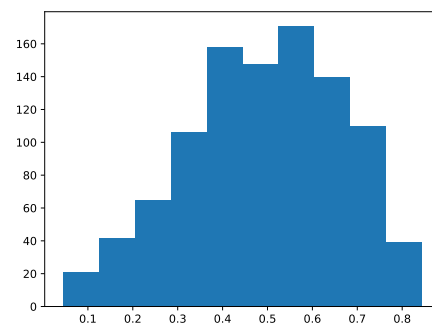


(b) Convolution layer 2 activations

Figure 25: Activations for each convolution in a max-pooling network.



(a) Results in each observation for each function.



(b) Histogram showing the differences between both functions for each observation in the sample.

Figure 26: Comparison of each pool function for a random sample ($N=1000$) in $[0, 1]^4$.

Class	Max	Choquet
Global	0.809	0.772
Airplane	0.794	0.753
Automobile	0.892	0.889
Bird	0.801	0.725
Cat	0.657	0.618
Deer	0.847	0.796
Dog	0.659	0.608
Frog	0.877	0.816
Horse	0.836	0.801
Ship	0.890	0.831
Truck	0.834	0.822

(a) Accuracy for each class for max pooling

Table 4: Accuracy comparison for max-pooling and Choquet integral pooling after 10000 iterations on CifarNet using CIFAR10 dataset.

5 CONCLUSION

This dissertation work presented a new form of performing the aggregation function inside the pooling layer of a DLN for image classification using the CIFAR10 dataset.

The research objective was to carry out the experiment in two phases: the realization of dimensional reduction of images using pre-aggregation functions based on generalizations of the Choquet integral simulating the pooling layer of a DLN and in the second phase is presented a discrete non monotonic Choquet integral, which is applied in DLN pooling layer, defined by a non monotonic fuzzy measure which is learned by the DLN architecture itself using backpropagation algorithm, obtained close results comparing with max-pooling accuracies.

The concepts of discrete non monotonic Choquet integrals, whose definition consider non monotonic fuzzy measures μ such that $\mu(N)$ may not be necessarily equal to 1 are both contributions of this dissertation.

Based on the results of the first phase, where the experiment was carried out without the use of CNN, the best result was chosen (the standard Choquet integral) through quantitative comparisons (using image quality measures) and qualitative, that is, visually. Posteriorly the second phase, which was performed using CifarNet for image classification of the CIFAR10 dataset, the final accuracies of the maximum aggregation function and discrete non monotonic Choquet integral were compared.

The discrete non monotonic Choquet integral presented 77,2% of accuracy after 10000 iterations. Due to computational problems it was necessary to stop the experiment, but further training would surely improve the result. Training with Max pooling using the same DLN architecture and number of iterations, the final accuracy is 81%.

The results are interesting even if the function used is not necessarily a Choquet integral function (since the measure is not always increasing).

The research shows to be prosperous tending to refine the studies. In the future, it is expected to study more deeply the functions presented by Lucca et al. (LUCCA et al., 2018), such as the functions shown in Annex 1, being able to apply them in other DLNs architectures and for other purposes, as image restoration in mixed media or identification of moving objects in real time.

REFERENCES

- ALSINA, C.; FRANK, M. J.; SCHWEIZER, B. **Associative Functions: triangular Norms and Copulas**. World Scientific, 2006.
- AUMANN, R. J.; SHAPLEY, L. S. **Values of non-atomic games**. Princeton University Press, 2015.
- BARRENECHEA, E.; BUSTINCE, H.; FERNANDEZ, J.; PATERNAIN, D.; SANZ, J. A. Using the Choquet Integral in the Fuzzy Reasoning Method of Fuzzy Rule-Based Classification Systems. **Axioms**, v.2, n.2, p.208–223, 2013.
- BARRENECHEA, E.; BUSTINCE, H.; FERNANDEZ, J.; PATERNAIN, D.; SANZ, J. Using the Choquet integral in the fuzzy reasoning method of fuzzy rule-based classification systems. **Axioms**, v.2, n.2, p.208–223, 2013.
- BELIAKOV, G.; SOLA, H. B.; SANCHEZ, T. C. **A Practical Guide to Averaging Functions**. Springer, 2016.
- BOUVRIE, J. Notes on convolutional neural networks. , 2006.
- BRADY, M. J. **Biometric recognition using a classification neural network**. Google Patents, 1999. US Patent 5,892,838.
- BUSTINCE, H.; FERNÁNDEZ, J.; KOLESÁROVÁ, A.; MESIAR, R. Directional monotonicity of fusion functions. **European Journal of Operational Research**, v.244, n.1, p.300–308, 2015.
- CHOQUET, G. Theory of Capacities. **Annales de l’Institut Fourier**, v.5, p.131–295, 1953–1954.
- C.SASI VARNAN, C.; MULLANA, M. Image quality assessment techniques pn spatial domain. **Int. J. Comput. Sci. Technol**, v.2, n.3, 2011.
- DEEPIKA JASWAL SOWMYA VISHVANATHAN, S. K. Image Classification Using Convolutional Neural Networks. , 2014.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: COMPUTER VISION AND PATTERN RECOGNITION, 2009. CVPR 2009. IEEE CONFERENCE ON, 2009. **Anais...** 2009. p.248–255.

DENG, L.; YU, D. et al. Deep learning: methods and applications. **Foundations and Trends® in Signal Processing**, v.7, n.3–4, p.197–387, 2014.

DENNEBERG, D. **Non-additive measure and integral**. Springer Science & Business Media, 2013. v.27.

DIAS, C. A.; BUENO, J. C.; BORGES, E. N.; BOTELHO, S. S.; DIMURO, G. P.; LUCCA, G.; FERNANDÉZ, J.; BUSTINCE, H.; JUNIOR, P. L. J. D. Using the Choquet Integral in the Pooling Layer in Deep Learning Networks. In: NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY ANNUAL CONFERENCE, 2018. **Anais...** 2018. p.144–154.

DIMURO, G. P.; LUCCA, G.; SANZ, J. A.; BUSTINCE, H.; BEDREGAL, B. CMin-Integral: A Choquet-Like Aggregation Function Based on the Minimum t-Norm for Applications to Fuzzy Rule-Based Classification Systems. In: AGGREGATION FUNCTIONS IN THEORY AND IN PRACTICE, 2018, Cham. **Anais...** Springer International Publishing, 2018. p.83–95.

EGMONT-PETERSEN, M.; RIDDER, D. de; HANDELS, H. Image processing with neural networks—a review. **Pattern recognition**, v.35, n.10, p.2279–2301, 2002.

ESKICIOGLU, A. M.; FISHER, P. S. Image Quality Measures And Their Performance. **IEEE Transactions on communications**, v.43, n.12, p.2959–2965, 1995.

FUKUSHIMA, K. Neocognitron—A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. **Biological cybernetics**, n.15, p.p106–115, 1981.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. MIT press, 2016.

GRABISCH, M. et al. **Set functions, games and capacities in decision making**. Springer, 2016.

GRABISCH, M.; MARICHAL, J.; MESIAR, R.; PAP, E. **Aggregation Functions**. Cambridge: Cambridge University Press, 2009.

HAMEL, P.; LEMIEUX, S.; BENGIO, Y.; ECK, D. Temporal Pooling and Multiscale Learning for Automatic Annotation and Ranking of Music Audio. In: ISMIR, 2011. **Anais...** 2011. p.729–734.

HAN, Y.; KIM, J.; LEE, K.; HAN, Y.; KIM, J.; LEE, K. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. **IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)**, v.25, n.1, p.208–221, 2017.

HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-r.; JAITLY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. **IEEE Signal Processing Magazine**, v.29, n.6, p.82–97, 2012.

HODGES, J. L.; LEHMANN, E. L. Rank methods for combination of independent experiments in analysis of variance. **The Annals of Mathematical Statistics**, v.33, n.2, p.482–497, 1962.

JURIO, A.; PAGOLA, M.; MESIAR, R.; BELIAKOV, G.; BUSTINCE, H. Image magnification using interval information. **IEEE Transactions on Image Processing**, v.20, n.11, p.3112–3123, 2011.

KARPATHY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; FEI-
FEI, L. Large-scale video classification with convolutional neural networks. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2014. **Proceedings...** 2014. p.1725–1732.

KAVUKCUOGLU, K.; SERMANET, P.; BOUREAU, Y.-L.; GREGOR, K.; MATHIEU, M.; CUN, Y. L. Learning convolutional feature hierarchies for visual recognition. In: **ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS**, 2010. **Anais...** 2010. p.1090–1098.

KIM, Y. Convolutional neural networks for sentence classification. **arXiv preprint arXiv:1408.5882**, 2014.

KLEMENT, E. P.; MESIAR, R.; PAP, E. **Triangular Norms**. Dordrecht: Kluwer Academic Publisher, 2000.

KRIZHEVSKY, A.; HINTON, G. **Learning multiple layers of features from tiny images**. Citeseer, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS**, 2012. **Anais...** 2012. p.1097–1105.

KUSSUL, E.; BAIDYK, T. Improved method of handwritten digit recognition tested on MNIST database. **Image and Vision Computing**, v.22, n.12, p.971–981, 2004.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v.86, n.11, p.2278–2324, 1998.

LEE, C.-Y.; XIE, S.; GALLAGHER, P.; ZHANG, Z.; TU, Z. Deeply-supervised nets. In: **ARTIFICIAL INTELLIGENCE AND STATISTICS**, 2015. **Anais...** 2015. p.562–570.

LEONARD, J.; KRAMER, M. Improvement of the backpropagation algorithm for training neural networks. **Computers & Chemical Engineering**, v.14, n.3, p.337–341, 1990.

LU, D.; WENG, Q. A survey of image classification methods and techniques for improving classification performance. **International journal of Remote sensing**, v.28, n.5, p.823–870, 2007.

LUCCA, G.; SANZ, J. A.; DIMURO, G. P.; BEDREGAL, B.; ASIAIN, M. J.; ELKANO, M.; BUSTINCE, H. CC-integrals: Choquet-like copula-based aggregation functions and its application in fuzzy rule-based classification systems. **Knowledge-Based Systems**, v.119, p.32–43, 2017.

LUCCA, G.; SANZ, J. A.; DIMURO, G. P.; BEDREGAL, B.; BUSTINCE, H.; MESIAR, R. C_F -integrals: A new family of pre-aggregation functions with application to fuzzy rule-based classification systems. **Information Sciences**, v.435, p.94–110, 2018.

LUCCA, G.; SANZ, J. A.; DIMURO, G. P.; BEDREGAL, B.; MESIAR, R.; BUSTINCE, A. K.-R. H. Preaggregation functions: Construction and an application. **IEEE Transactions on Fuzzy Systems**, v.24, n.2, p.260–272, 2016.

MARÉE, R.; GEURTS, P.; PIATER, J.; WEHENKEL, L. Biomedical image classification with random subwindows and decision trees. In: **INTERNATIONAL WORKSHOP ON COMPUTER VISION FOR BIOMEDICAL IMAGE APPLICATIONS**, 2005. **Anais...** 2005. p.220–229.

MISHRA, A.; ALAHARI, K.; JAWAHAR, C. Scene text recognition using higher order language priors. In: **BMVC 2012-23RD BRITISH MACHINE VISION CONFERENCE**, 2012. **Anais...** 2012.

MUROFUSHI, T.; SUGENO, M. An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure. **Fuzzy sets and Systems**, v.29, n.2, p.201–227, 1989.

MUROFUSHI, T.; SUGENO, M.; MACHIDA, M. Non-monotonic fuzzy measures and the Choquet integral. **Fuzzy sets and Systems**, v.64, n.1, p.73–86, 1994.

PAGOLA, M.; FORCEN, J. I.; BARRENECHEA, E.; LOPEZ-MOLINA, C.; BUSTINCE, H. Use of OWA operators for feature aggregation in image classification. In:

FUZZY SYSTEMS (FUZZ-IEEE), 2017 IEEE INTERNATIONAL CONFERENCE ON, 2017. **Anais...** 2017. p.1–6.

PAL, N. R.; PAL, S. K. A review on image segmentation techniques. **Pattern recognition**, v.26, n.9, p.1277–1294, 1993.

PAP, E. Regular null-additive monotone set functions. **Novi Sad. J. Math**, v.25, n.2, p.93–101, 1995.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. **Neural computation**, v.29, n.9, p.2352–2449, 2017.

RÉBILLÉ, Y. Sequentially continuous non-monotonic Choquet integrals. **Fuzzy Sets and Systems**, v.153, n.1, p.79–94, 2005.

ROTH, H. R.; LU, L.; LIU, J.; YAO, J.; SEFF, A.; CHERRY, K.; KIM, L.; SUMMERS, R. M. Improving computer-aided detection using convolutional neural networks and random view aggregation. **IEEE transactions on medical imaging**, v.35, n.5, p.1170–1181, 2016.

SCHERER, D.; MULLER, A.; BEHNKE, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. **International Conference on Artificial Neural Networks**, p.92–101, 2010.

SUGENO, M. Theory of fuzzy integrals and its applications. **Doct. Thesis, Tokyo Institute of technology**, 1974.

SUGENO, M.; MUROFUSHI, T. Pseudo-additive measures and integrals. **Journal of Mathematical Analysis and Applications**, v.122, n.1, p.197–222, 1987.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015. **Proceedings...** 2015. p.1–9.

WILCOXON, F. Individual comparisons by ranking methods. **Biometrics bulletin**, v.1, n.6, p.80–83, 1945.

YU, D.; WANG, H.; CHEN, P.; WEI, Z. Mixed pooling for convolutional neural networks. In: INTERNATIONAL CONFERENCE ON ROUGH SETS AND KNOWLEDGE TECHNOLOGY, 2014. **Anais...** 2014. p.364–375.

6 ANNEX 1

Table 5: Results clipping of image resizing using different combinations of functions, measures and parameters. Author: Cédric Marco-Dechart

Function 1	Function 2	Measure	Window	Stride	AD	MD	MSE	NAE	NK	PSNR	SC
CFIF2	sine-FBPC	owa	2x2	2x2	-0,18886	-0,04642	0,047654	0,232832	1,210775	61,34983	0,669935
CFIF2	sine-FBPC	wmean	2x2	2x2	-0,18886	-0,04642	0,047654	0,232832	1,210775	61,34983	0,669935
CFIF2	GM-lukasiewicz	power-1-0000	2x2	2x2	-0,18886	-0,04642	0,047654	0,232832	1,210775	61,34983	0,669935
CTM	min	wmean	2x2	2x2	0,645002	0,768627	0,448131	0,796755	0,192392	51,61676	18,62213
CTM	prod	power-1-0000	2x2	2x2	-0,00011	0,313725	0,003145	0,032779	0,995452	73,15506	1,004421
CTM	prod	power-0-1000	2x2	2x2	-0,02925	0,05098	0,006244	0,039347	1,025547	70,17621	0,943028
CC	hamacher	power-1-0000	2x2	2x2	-0,02553	0,2	0,005088	0,03852	1,022312	71,06508	0,950373
CC	hamacher	power-0-1000	2x2	2x2	-0,03341	0,019608	0,007147	0,041693	1,029874	69,58949	0,934217
CC	hamacher	power-0-2000	2x2	2x2	-0,0327	0,039216	0,006915	0,041301	1,029193	69,73261	0,935709
CC	hamacher	power-0-3000	2x2	2x2	-0,03191	0,062745	0,006674	0,040927	1,028439	69,88691	0,937348
CTM	OB	power-0-4000	3x3	2x2	-0,02285	0,463217	0,008558	0,048095	1,016356	68,80733	0,956494
CTM	OB	power-0-5000	3x3	2x2	-0,02091	0,463217	0,008282	0,047521	1,014335	68,94957	0,960585
CTM	OB	power-0-6000	3x3	2x2	-0,01919	0,467139	0,008049	0,04708	1,012565	69,07335	0,964186
CTM	OB	power-0-7000	3x3	2x2	-0,01732	0,467139	0,00785	0,046673	1,010591	69,18183	0,968149