POLITECNICO DI TORINO

Master's Degree in Physics of Complex Systems

Master's Degree Thesis

A new metric for the interpretability of artificial neural networks in medical diagnosis applications

Supervisors

Prof. Alfredo BRAUNSTEIN

Prof. Antonio SIMEONE

Candidate

Niccolò MANFREDI SELVAGGI

October 2022

Summary

Machine learning (ML) is a powerful tool for automating some tasks that humans can easily accomplish but, at the same time, are difficult to implement using "classic" algorithms.

In recent years, thanks to the increase in the computing power of computers and the amount of data collected, algorithms have become more complex and efficient in solving increasingly difficult tasks.

The increase in the complexity of algorithms, such as Deep Learning, has made the models *black boxes*, i.e. machines capable of performing a task faster and sometimes even better than humans, but without a way of understanding to which criteria and which calculations the algorithm provided a certain output.

One of the most promising areas for these technologies is medical diagnosis [1], because the identification of pathologies is a simple classification of medical data that can be collected in databases and often made up of numerical values and images, formats that can be easily processed by modern machine learning.

In this sector, however, trust in the outcome of the diagnosis and the legal responsibility of the doctor as regards his professionalism and any errors made are of fundamental importance.

However much an algorithm can be tested and, however it can be often shown that it has a lower error rate than the human one, it is inevitable that it will make mistakes and misdiagnoses too, and there is also the problem of entrusting responsibility for the life of patients to a non-human agent using non-interpretable methods [2].

This thesis proposes an empirical mathematical analysis of artificial neural networks (ANN) from which it is possible to develop a new metric to evaluate the diagnostic models under the aspect of *interpretability*.

The purpose of this thesis is to extrapolate certain statistical information on the reliability of the output of ML models through which it is possible to implement rational decision making protocols.

Underlying the ideas of the proposed new metric is a new mathematical formalism developed by Professor Audun Josang called *Subjective Logic* [3] which provides a series of tools aimed at the problem of decision making under uncertainty.

Table of Contents

Chapter 1

Machine learning

This chapter is an introduction to the main definition and the aim of machine learning and ANNs in binary classification and to some of the most important concept used in the thesis .

1.1 Neural Networks for binary classification

Machine learning (ML) [4] is the scientific branch that studies a specific kind of algorithms that are able to perform statistical analysis of data by a system of continuous improvement that takes place through the observation of samples of data provided in input. The types of tasks that such an algorithm can accomplish are very difficult for any algorithm that doesn't fit into the ML category, but they are eventually very easy for humans to solve. For this reason these techniques are often associated with the term *Artificial Intelligence*.

Recently, thanks to the increase in the computational power of computers and the amount of data collected from the real world, machine learning is currently widely used in industry and has achieved remarkable results in many sectors such as text categorization [5], spam filter [6], artificial vision [7, p. 15] and search engines [8].

The most successful and famous models in ML are artificial neural networks. This family of models are inspired by the neural structure of living beings. The information is processed through signals that travel through neurons, which are simple computing units that work in parallel, forming a structure capable of performing very complex calculations.

One of the main tasks that ANN's can perform thanks to machine learning is binary classification, that is the task of classifying objects in two groups on the basis of classification rules. In the case of ANN's the classification rule is contained in the values of the many parameters of the model, and it is very difficult to interpret

because of the complexity and non-linearity of the structure of the neural network [9][10].

1.1.1 Neural network basic concepts

Mathematically speaking, an artificial neural network can be interpreted as a multidimensional parametric function. The input has the same shape of the instance of the database (typically a multidimensional array of real numbers), and the output has a shape depending on the type of the task that the network has to perform. In the case of binary classification, it is typically used an output composed by two values (y1, y2 in fig 1.1) ranging from 0 to 1, with the objective that each one will represent the prospect of belonging to the respective class.

In the fig 1.1 is represented the structure of the Dense NN (the most common and simple example of ANN) of which the algorithm is discussed in more detail.

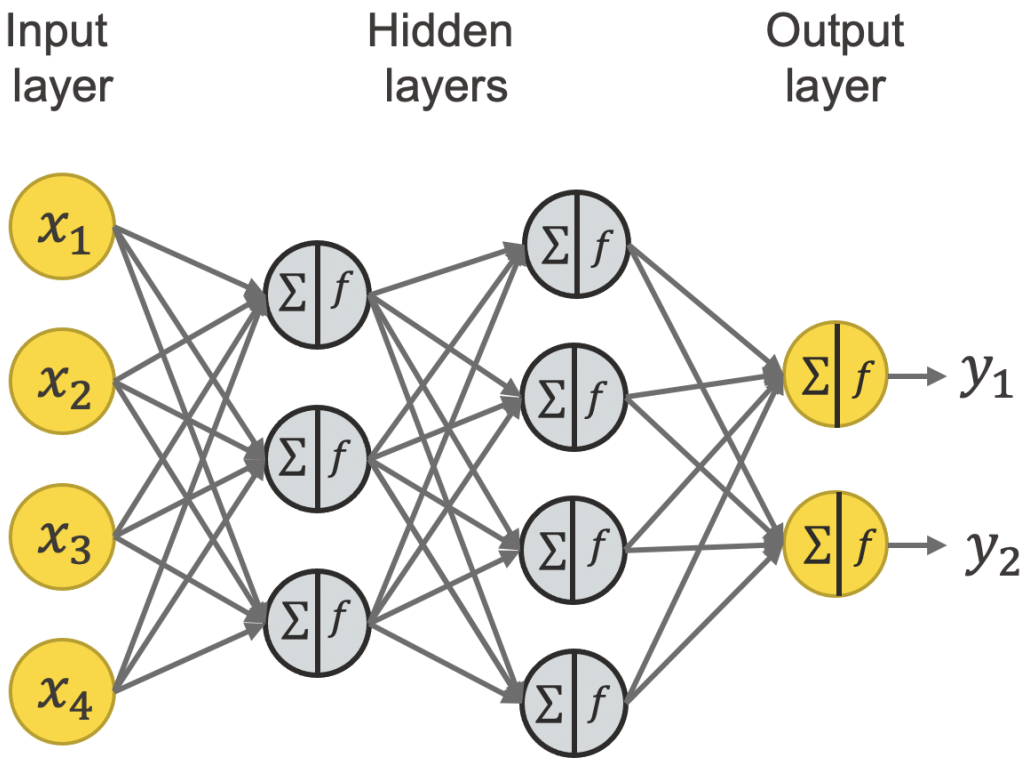

Figure 1.1: A Dense neural network architecture

The operation performed by the neural network can be summarized in this way:

- The input of every neuron of a layer is the linear combination of the output of the previous layer with a set of free parameters that will be adjusted during the training.

$$I_{i,j} = \sum_{k=1}^{n_i} b_{i-1,k,j} \cdot O_{i-1,k-1} \tag{1.1}$$

- Every neuron of each layer applies a non linear function (called the activation function) to its input. One of the most simple and used activation function is the ReLU (Rectified Linear Unit) [11] showed in fig 1.2.

$$O_{i,j} = f(I_{i,j}) \tag{1.2}$$

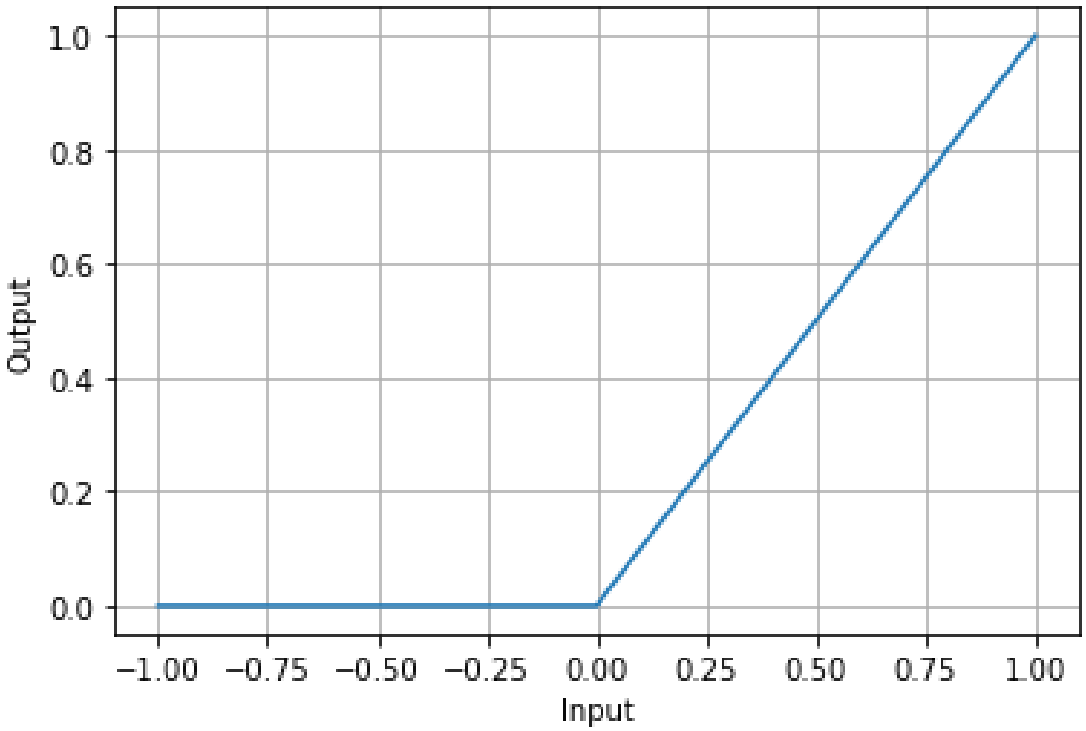

Figure 1.2: Plot of the ReLU activation function

Now that is clear how an ANN can compute the output starting from the input, it is possible to understand the basics of the learning algorithm by which all the parameters are adjusted in order to execute a classification or regression task. First we can notice that, by means of recursion of derivative of composite functions, it is possible to compute the derivative of each parameter with respect to the output neurons. So it is possible to apply any gradient descendent type of algorithms.

For each sample of the training set is performed the output and measured the error with respect to the real label of the sample. Then all the derivatives of the parameters with respect of the error are performed, and finally the gradient descendent slightly modify the values of all parameters. This method of parameter adjustment is called Back propagation [12], because the information propagates from the last layer backward along the network. These kind of algorithms usually converge to a local minimum, even if it's not yet generally demonstrated.

1.1.2 Overfitting

A very important concept in machine learning and which will have an important impact in this thesis, as will be illustrated in the chapter 5.4, is overfitting [13].

Overfitting is a training problem of machine learning models due to the limited size of the dataset with respect to the size of the model.

To explain the nature of this phenomenon, it is useful to imagine a neural network as a set of free parameters that can store information on datasets regarding the classification of elements.

The dataset used is a limited set of examples, but the algorithm is designed to work even in cases outside the dataset used for training. For this reason, datasets are usually divided into two, the training set and the test set. In this way the neural networks train by learning to classify the elements of the training set and, to verify that their ability is valid even for elements never seen by the network, we test on the test set.

It is easy to imagine that, if the model has too many parameters and the dataset is too small, it is possible to accumulate in the parameters the information of the training set with too high a level of detail, which is therefore not respected in the test set, in which the elements are distributed slightly differently.

To better visualize this phenomenon, it is very useful to view examples. The following image shows an example in the context of regression and one in the context of binary classification, which is of more interest in this thesis.

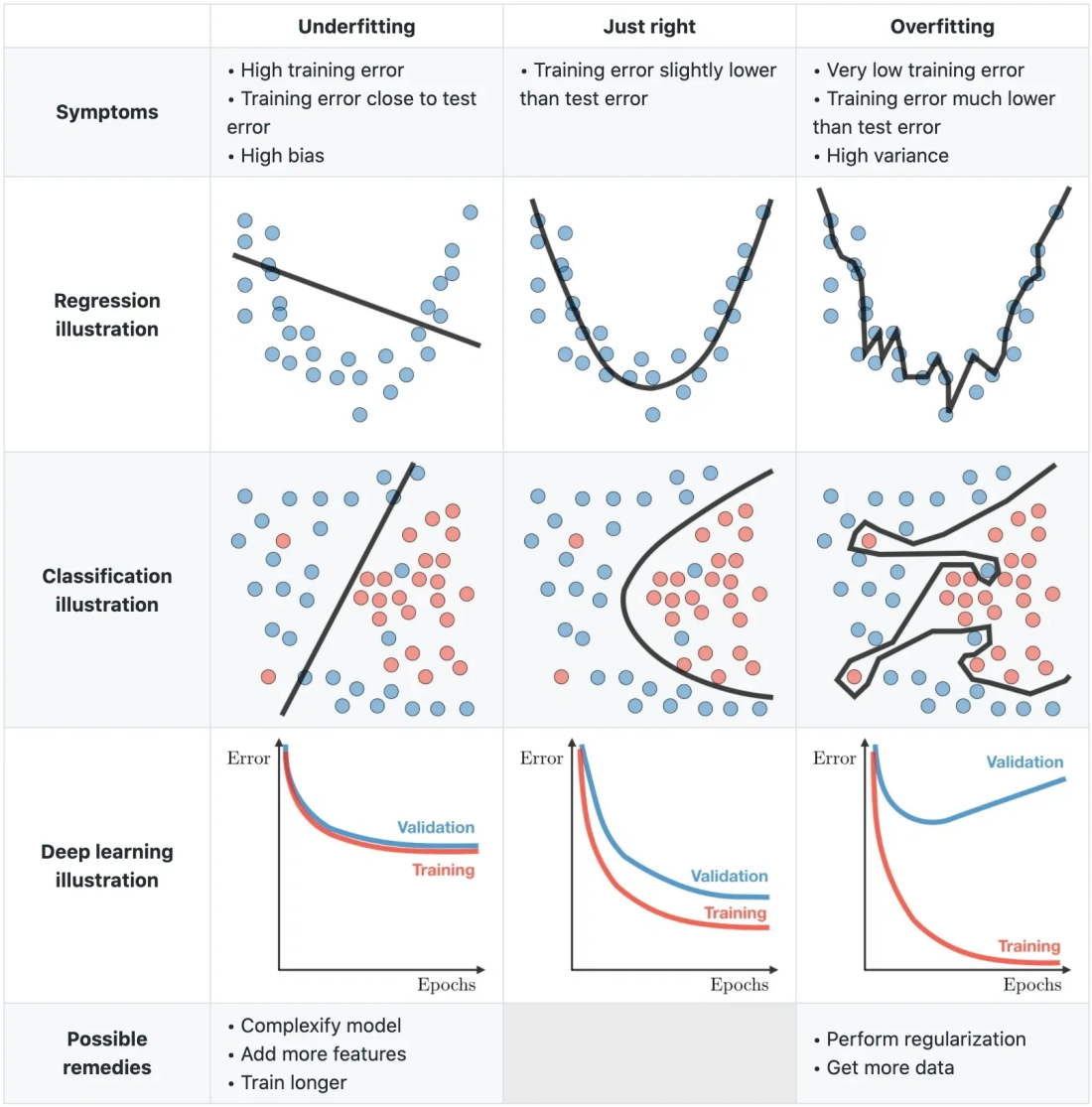

Figure 1.3: Overfitting representation scheme.

Apparently, looking only at the training set, overfitting does not lead to any problems, but a better classification. Only by looking at the test set does one realize that the model will no longer be able to classify well elements not present in the training set.

This difference between train and test set is the key to identifying the presence of overfitting.

In fact, looking at how the loss function of the training set and of the test set varies during the training, we notice a point where, while in the training set the loss function continues to decrease, in the test set it begins to increase. This is

a clear sign of overfitting, shown in figure 1.3, but also observable in the cases addressed in this thesis, as in figure 5.5.

1.1.3 CNN and image processing

This session briefly describes the most important application of neural networks in image processing, a task of fundamental importance in the medical field.

The ability to recognize patterns in images is a very difficult operation to perform through deterministic algorithms and at the same time a task that some types of neural networks are able to perform very successfully.

Humans and other animals learn very easily to recognize the objects around them through sight. This is due to the structure of the neurons involved in vision, which are distributed on various layers which, one after the other, identify patterns of increasing complexity, up to the point of segmenting very complex and characteristic patterns of certain types of objects. allowing for easy classification. To develop this skill it is necessary to practice, in fact we need to see the same category of objects several times to become good at distinguishing it.

Indeed, it is precisely this biological characteristic that has inspired the computer structure of Convolutional Neural Networks (CNN) [14][15].

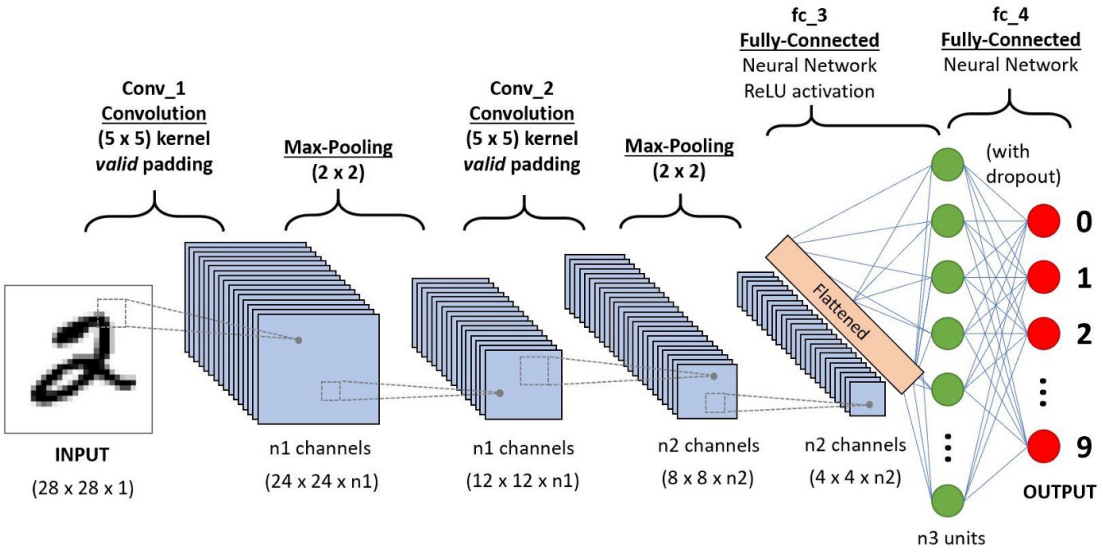

Figure 1.4: Structure representation of a deep CNN with final Dense layer for classification

As you can see in figure 1.4, this type of network uses filters that scan the whole image for a certain pattern. Each filter trains itself to recognize different patterns,

the choice of which patterns to look for is automated in the training process: the network itself learns which patterns to look for.

While the first layer is able to recognize very simple patterns (e.g. vertical and horizontal contours), through non-linearity each subsequent layer is able to construct more complex patterns (e.g. geomeric figures such as ovals or curves).

In the case of deep learning, where the number of hiden layers is quite high, the final layeers are able to adapt if they recognize patterns of arbitrary complexity (eg eyes, people, etc).

The output of these layers provides information on the presence of these patterns, which are easily processed by dense layers to obtain the final classification.

1.1.4 Softmax output layer

For the binary classification, we want the model to produce an output that gives information about what class the sample belong to. This is a binary information, but each neuron produce a real number. One possible strategy is to chose a 2 neuron output layer and add a Softmax function to the final output. The Softmax function [16] is often used for this kind of problems and it is very important for the main idea of the thesis. The function take two number as input and gives as output two positive numbers with unitary sum, according to the following formula:

$$O(I_1, I_2) = \left(\frac{e^{I_1}}{e^{I_1} + e^{I_2}}, \frac{e^{I_2}}{e^{I_1} + e^{I_2}} \right) \tag{1.3}$$

The final output has only one degree of freedom, because the two values are constrained by the fact that the sum is equal to one. The two values are greater then zero and less the one, which make them useful for representing probabilities. in fact the network is trained in a way that the two values refers to the two classes and value 0 and 1 will represent the state of truth of belonging to the relative class. So a well trained model will activate more the neuron corresponding to the more probable belonging class of the sample. This does not means that the output can be interpreted as the probability of belonging to the classes. This aspect is very important for the aim of the thesis and will be discussed in section 4.2

1.1.5 Cross entropy loss function

In section 1.1.1 is discussed the learning algorithm of the ANN and it is mentioned, without further investigation, the error between the output and the real label of a sample of the training set. In order to "teach" the network, it is needed to understand how far its prediction is from the real value that has to be predicted. This distance is called *loss function* and generally is an arbitrary choice. In this

section is presented one of the most used loss function in binary classification tasks with Softmax output layer: *the Cross entropy* [17, p. 82]. In section 4.2 will be clear why this loss function is very important for the aim of the thesis. Given (y_1, y_2) the output of the network and (l_1, l_2) the state of truth of belonging to the two classes, the cross entropy formula is given by:

$$Loss = -l_1 \cdot \log(y_1) - l_2 \cdot \log(y_2) \tag{1.4}$$

which can be also expressed by:

$$Loss = \begin{cases} -\log(y_1) & \text{if } l_1 = 1 \\ -\log(1 - y_1) & \text{if } l_1 = 0 \end{cases} \tag{1.5}$$

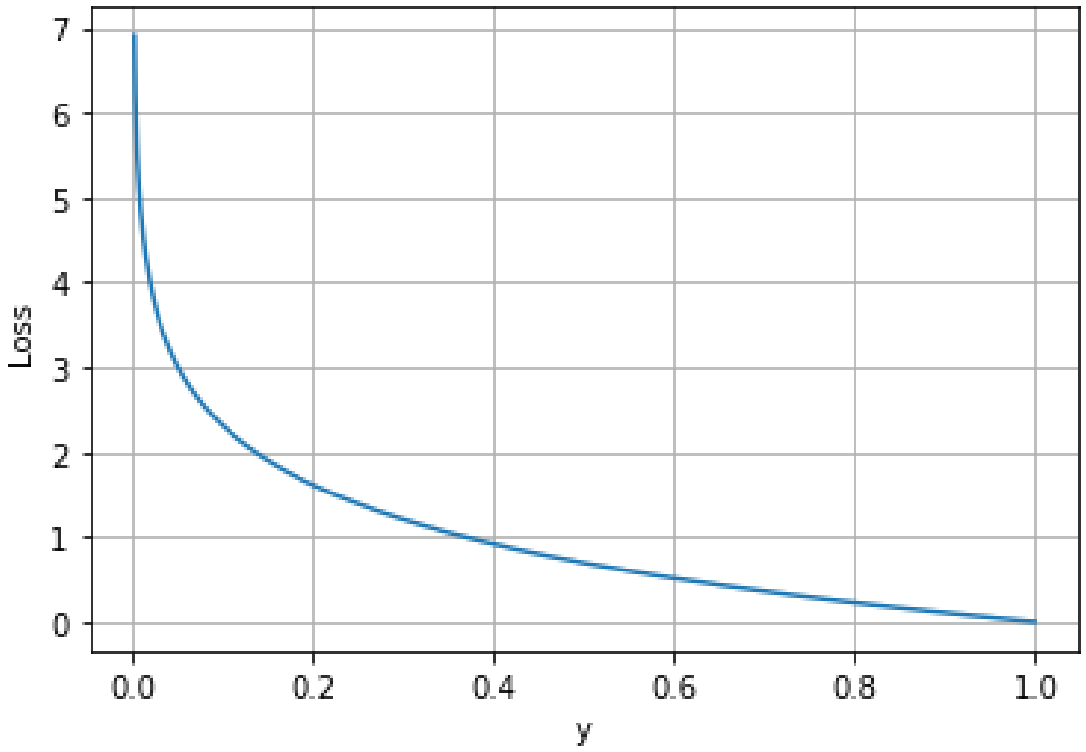

Figure 1.5: Plot of the cross entropy values when $l_1 = 1$

This function is very important because it derives from the *Kullback-Leibler divergence* between two distributions. If $p(x)$ is the empirical probability distribution of a sample to belong to one of the two classes and $q(x)$ is the model distribution (corresponding to the output value y_1) the cross entropy can be expressed as:

$$CE(p||q) = -\sum_{x\in\chi} p(x)\cdot \log q(x) = S(p) + D_{KL}(p||q) \tag{1.6}$$

Where:

- χ is the set of input samples
- $H(p)$ is the entropy of the distribution $p(x)$:

$$S(p) = -\sum_{x\in\chi} p(x)\log p(x) \tag{1.7}$$

- $D_{KL}(p||q)$ is the Kullback-Leibler divergence between $p(x)$ and $q(x)$:

$$D_{KL}(p||q) = \sum_{x\in\chi} p(x)\log\left(\frac{p(x)}{q(x)}\right) \tag{1.8}$$

All this concept are better explained in section 2.3.

1.2 Classical metrics

When a neural network for binary classification is trained, it's important to evaluate the performance in predicting the true class of a sample. A *metric* is a set of quantitative and computable parameters useful to evaluate and understand the performance of a machine learning model [18]. In this section are presented some of the most used metrics for binary classification All of them are based on a very simple interpretation of the output of the network. Section 1.1.4 shows that a network with Softmax output return a value (y_1) between zero and one representing the state of truth of belonging to one of the classes. The simplest interpretation of this value is to set a threshold (typically 0.5) and consider all the sample with y_1 greater then the threshold classified with class 1 (and vice-versa for class 2). Basically, this method force the output into a binary value, and does not care about the shade given by the variety of possible values. A 0.51 is equivalent to a 0.99. This is clearly a limitation of the classical metrics and one of the results of this thesis is overcome this "waste" of information. Anyway, this interpretation is very simple and gives rise to very practical and intuitive metrics.

Now it is possible to define four important values that compose the *confusion matrix*:

- **True positive (TP)**: The number of element belonging to class 1 and predicted with class 1

- **True negative (TN)**: The number of element belonging to class 2 and predicted with class 2
- **False positive (FP)**: The number of element belonging to class 2 and predicted with class 1
- **False negative (FN)**: The number of element belonging to class 1 and predicted with class 2

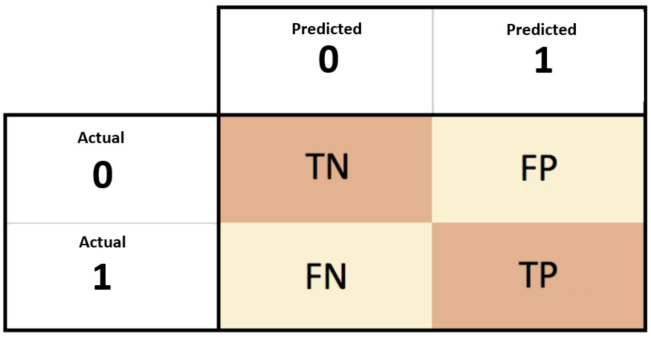

Figure 1.6: Representation of the confusion matrix

From this values, it is obtained the most important metric: the *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.9}$$

Which of course express the fraction of the samples well classified.
There are some other relevant metric values deriving from this values like:

- **Recall or true positive rate (TPR)**:

$$TPR = \frac{TP}{TP + FN} \tag{1.10}$$

- **Specificity or true negative rate (TNR)**:

$$TPR = \frac{TN}{TN + FP} \tag{1.11}$$

- **Precision or positive predictive value (PPV)**:

$$PPV = \frac{TP}{TP + FP} \tag{1.12}$$

- **Fall-out or false positive rate (FPR)**:

$$TPR = \frac{FP}{TN + FP} \tag{1.13}$$

The last metric covered by this section is the *ROC AUC* (Area under Receiver Operating characteristic Curve). This curve is obtained by varying the value of the threshold and putting on the x and y axis the false positive rate and the true positive rate, respectively. It can be shown that the area between this curve and the line $y = x$ is a good measure of the overall performance of the model.

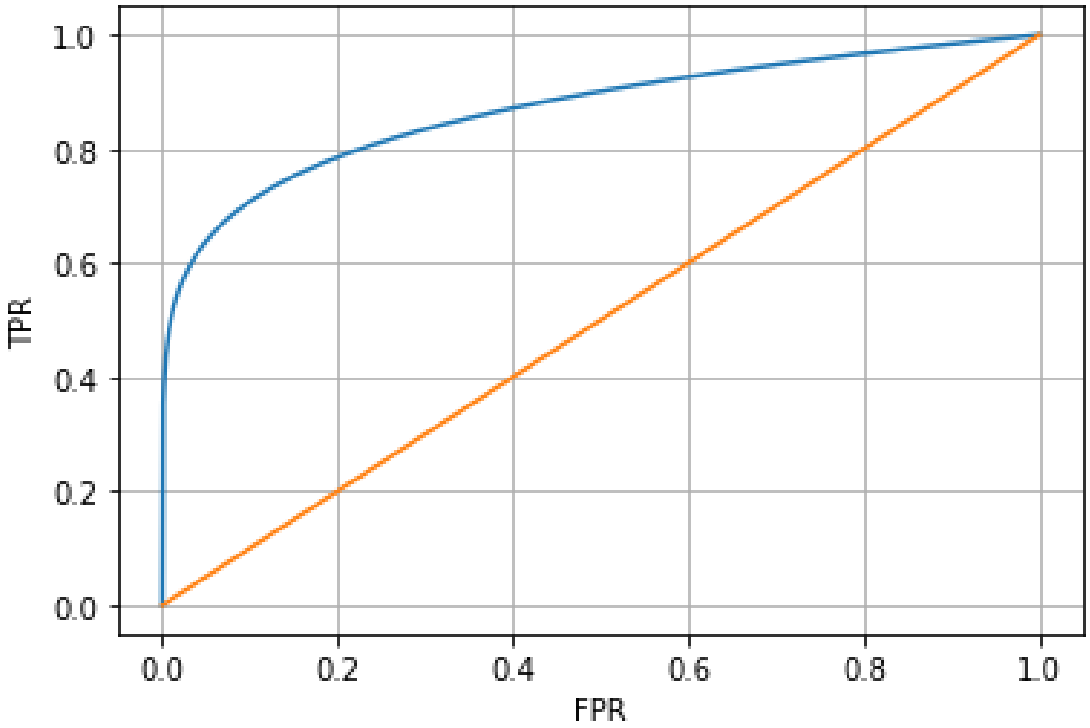

Figure 1.7: Plot of an instance of ROC

Chapter 2

Probability distributions

This chapter is an introduction to the basic concept of probability distributions [19] and a deepening of Beta distribution and Inference algorithms, which are the focus of the thesis.

2.1 Basic concepts

The probability of an event is a real number between 0 and 1 which express "how likely" the event is to occur in a way that 0 indicates impossibility and 1 indicates certainty.

$$0 < p(E) < 1 \tag{2.1}$$

In a purely theoretical setting, probabilities can be easy estimated by counting the number of positive and negative outcomes of the event that can be repeated indefinitely, but in practical application this is not possible and probability lose the definition of its fundamental nature.

For this reason, currently exist two categories of interpretations [20] of the probabilities:

- **Objectivist interpretation**: This interpretation uses numbers to describe some objective or physical state. A very common example is the *frequentist probability*, in which the probability is interpreted by the relative frequency of occurrence of a random event experiment's outcome when the experiment is repeated many times.

- **Subjectivist interpretation**: In this case the probability is a measure of the degree of belief, and it is a subjective concept. The degree of belief can be interpreted as "the price at which a rational agent would make a bet with

jackpot equal to 1 if case of winning, and 0 in case of losing". The most famous is the *Bayesan probability*, according to which probability is interpreted as reasonable representation of a state of knowledge of a subjective belief.

Normally, the subjective probability converges to the frequentist probability in case where empirical data is avaliable, and it is possible to make statistics about the occurence of the event. [21]. These concepts are important for machine learning and for this thesis because, as discussed in section 1.1.4, artificial neural networks for binary classification are "entities" which, via complex non-interpretable calculation, return a value between 0 and 1 that we want to interpret as the belief of the network of the sample belonging to a certain class. Part of the idea of this thesis is based on the consequence of applying these concepts [22] to neural networks, as will be discussed in section 4.2.

So far, has not yet been fully discussed the concept of *stochastic variable*. Before in this section it was mentioned only the event E, which is a binary random variable. That means that the outcome of the event can numerically be represented as either 0 or 1. In general, a stochastic (or random) variable is an event whose outcome is unknown a priori and can be formalized by a mathematical object.

The simplest random variables are those which outcome is a real number belonging to some domain. For these variable, each possible value is an outcome of a random event so it must have a probability. However there is a uncountable number of real numbers (and so outcomes), so it is only possible to define the probability that the variable belong to an arbitrary small interval of possible values in the following way:

$$P(z < X < z + dz) = f(z)dz \tag{2.2}$$

Where $f(z)$ is called the *probability distribution* (PDF) associate with the variable X, and can be expressed as a real function defined on the domain of the variable. Since the outcome of the variable has to be one of the values of the domain, the probability of X belonging to the domain must be 1, imposing the following *normalization* property:

$$P(X \in D) = \int_{z \in D} f(z)\, dz = 1 \tag{2.3}$$

It is possible to define the probability that the variable is less then a value, this quantity is called the *cumulative distribution function*(CDF):

$$F(z) = P(X < z) = \int_{z_0}^{z} f(y)\, dy \tag{2.4}$$

So it is immediate that:

$$f(z) = \frac{dF(z)}{dz} \tag{2.5}$$

There are other two noteworthy quantities connected to probability distributions:

- **The expectation value**:

$$E[X] = \int_{z \in D} z f(z)\, dz \tag{2.6}$$

  Which can be interpreted as the average value obtained from a very large numbers of experiments.

- **The variance**:

$$VAR[X] = E[(X - E[X])^2] = \int_{z \in D} (z - E[X])^2 f(z)\, dz \tag{2.7}$$

  Expressible also in the following way:

$$VAR[X] = E[X^2] - E[X]^2 = \int_{z \in D} z^2 f(z)\, dz - \left(\int_{z \in D} z f(z)\, dz \right)^2 \tag{2.8}$$

  Which indicates how "sparse" around the average the outcomes of many experiments would be.

2.2 Beta distribution

The family of probability distributions discussed and deepened in this section is a very useful tool for the aim of this thesis. from now on there will be a slight change of notation, in particular x will be used to indicate what was previously indicated by z.

The *beta distribution*[23] is a family of probability distributions defined on interval [0,1]. It can be expressed by the following PDF:

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha - 1} (1 - x)^{\beta - 1} \tag{2.9}$$

Where $\alpha \geq 0$, $\beta \geq 0$ are the shape parameters,

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \tag{2.10}$$

is the normalization constant and Γ is the Gamma function.

The beta distribution model is often used for the description of the random behavior of fractions and can be used as probability distribution of probabilities. (see chapters 4 and 3)

Furthermore, beta distribution is the *maximum entropy* or *least informative* probability distribution with constraints $E[\log(x)]$ and $E[\log(1 - x)]$.

Following, some other useful properties of the Beta distribution:

- **The cumulative distribution function**:

$$F(x;\alpha,\beta) = \frac{B(x;\alpha,\beta)}{B(\alpha,\beta)} \tag{2.11}$$

Where

$$B(x;\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\,dx \tag{2.12}$$

Is the regularized incomplete beta function [24].

- **The expectation value**:

$$E[X] = \frac{\alpha}{\alpha+\beta} \tag{2.13}$$

- **The variance**:

$$VAR[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{2.14}$$

Lastly, a useful and intuitive graphic representation of the beta distribution family:

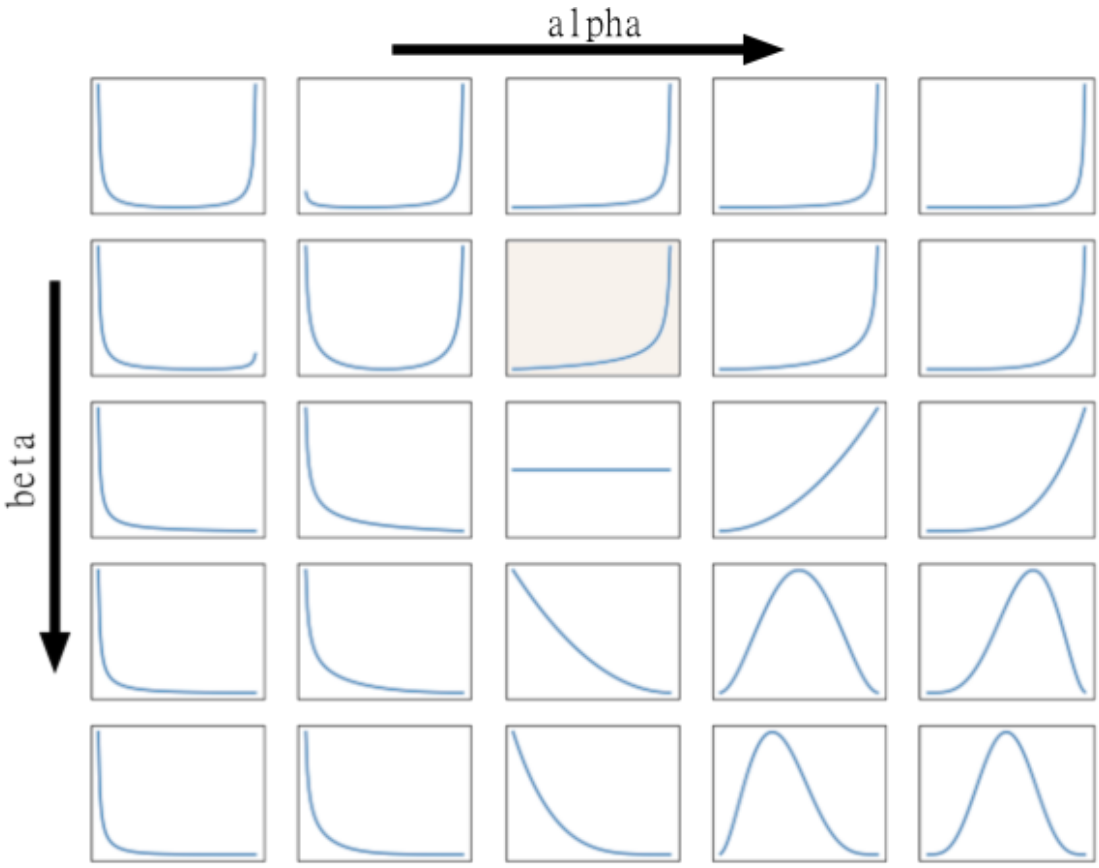

Figure 2.1: Plots of beta distribution for different parameter values

2.3 Inference

Inference[**Inference**] is the process by which a conclusion is inferred from multiple observations is called inductive reasoning. In statistics, inference means find the "best" probability distribution that explain a set of observation. In order to understand what "best" means, it is necessary to recall the *Bayes theorem*. Given two events A and B, the Bayes theorem states that:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.15}$$

Where $P(A|B)$ is the conditional probability of A given B, that is the probability of the event A when it is known that the event B is already happened.

Now consider some samples of a random variable x and a probability distribution q. In order to find the "best" distribution q that explain the data, it is natural

to look for the most likely distribution that may have generated that data. This correspond to maximize the conditional probability of the distribution given the data, also called the *posterior probability*:

$$P(q|x) = \frac{P(x|q)P(x)}{P(q)} \tag{2.16}$$

Where $P(x|q)$ is called the *likelihood* of the data given the distribution and $P(q)$ the *prior probability* of the distribution, which indicates how much it is likelihood that distribution without knowing any data.

Under the assumption of uniform prior probability, maximizing the posterior is equivalent to maximizing the likelihood.

Now consider a model, parameterized with a set of parameter θ, that return a probability distribution $q(x|\theta)$, and the *empirical distribution* defined as follows:

$$\tilde{p}(x) = \frac{1}{N}\sum_{\mu=1}^{N}\delta(x, x_\mu) \tag{2.17}$$

The following quantities can be defined:

- Shannon's Entropy [25] of the empiric distribution:

$$S(\tilde{p}) = -\sum_{x\in\chi}\tilde{p}(x)\log\tilde{p}(x) \tag{2.18}$$

- Cross entropy [17] between empirical and model distribution:

$$CE(\tilde{p}||q) = -\sum_{x\in\chi}\tilde{p}(x)\cdot\log q(x|\theta) \tag{2.19}$$

- Kullback-Leibler Divergence [26] between empirical and model distribution:

$$D_{KL}(\tilde{p}||q) = \sum_{x\in\chi}\tilde{p}(x)log\left(\frac{\tilde{p}(x)}{q(x|\theta)}\right) \tag{2.20}$$

It is possible to demonstrate that minimizing the Kullback-Leibler Divergence with respect to the model parameter θ is equivalent to minimizing the cross entropy and to maximizing the likelihood.

First, it is easy to notice that the difference between the cross entropy and the Kullback-Leibler Divergence is the empirical entropy, which does not depend on θ, so minimizing the Kullback-Leibler Divergence is equivalent to minimizing the cross entropy.

$$D_{KL}(\tilde{p}||q) = CE(\tilde{p}||q) - S(\tilde{p}) \tag{2.21}$$

Second, it is possible to demonstrate that the cross entropy is proportional to the opposite of the logarithm of the likelihood in the following way:

$$\begin{aligned} CE(\tilde{p}||q) &= -\sum_{x\in\chi} \tilde{p}(x)\cdot \log q(x|\theta) \\ &= -\frac{1}{N}\sum_{x\in\chi}\sum_{\mu=1}^{N} \delta(x,x_\mu)\log q(x|\theta) \\ &= -\frac{1}{N}\sum_{\mu=1}^{N} \log q(x_\mu|\theta) \\ &= -\frac{1}{N}\log\prod_{\mu=1}^{N} q(x_\mu|\theta) \\ &= -\frac{1}{N}\log q(x|\theta) \end{aligned} \tag{2.22}$$

The previous calculations show the importance of the cross entropy loss function discussed in section 1.1.5.

For Beta distributions, it is possible to calculate the likelihood with respect to the shape parameters. If the samples are statistically independent beta distributed variables, the expression of the log-likelihood of N observations is:

$$\begin{aligned} lnL(\alpha,\beta|X) &= \sum_{\mu=1}^{N} ln\left(\frac{x_\mu^{\alpha-1}(1-x_\mu)^{\beta-1}}{B(\alpha,\beta)}\right) \\ = (\alpha-1)\sum_{\mu=1}^{N} ln(x_\mu) &+ (\beta-1)\sum_{\mu=1}^{N} ln(1-x_\mu) - NlnB(\alpha,\beta) \end{aligned} \tag{2.23}$$

It is possible to find the maximum by setting the derivatives with respect to the parameters values equal to zero:

$$\begin{aligned} \frac{\delta ln(\alpha,\beta|X)}{\delta\alpha} &= \sum_{\mu=1}^{N} ln(x_\mu) - N\frac{\delta lnB(\alpha,\beta)}{\delta\alpha} = 0 \\ \frac{\delta ln(\alpha,\beta|X)}{\delta\beta} &= \sum_{\mu=1}^{N} ln(1-x_\mu) - N\frac{\delta lnB(\alpha,\beta)}{\delta\beta} = 0 \end{aligned} \tag{2.24}$$

From the previous condition, it is possible to obtain the value of the parameters by inverting, via numerical techniques, the following coupled equations:

$$\begin{aligned} E[ln(X)] &= \frac{1}{N}\sum_{\mu=1}^{N} ln(x_\mu) \\ E[ln(1-X)] &= \frac{1}{N}\sum_{\mu=1}^{N} ln(1-x_\mu) \end{aligned} \tag{2.25}$$

Chapter 3

Subjective logic

This chapter is an introduction to the original mathematical formalism developed by professor Audun Josang in 1996 [3] to better express concepts like uncertainty and trust. This topic is important for the thesis because, under some conditions, it is possible to perform a mapping between Subjective Logic and well trained ANN outputs.

Subjective logic gives a very useful approach to decision making under uncertainty, so by this mapping it is possible to apply to neural network all the tools developed in this formalism.

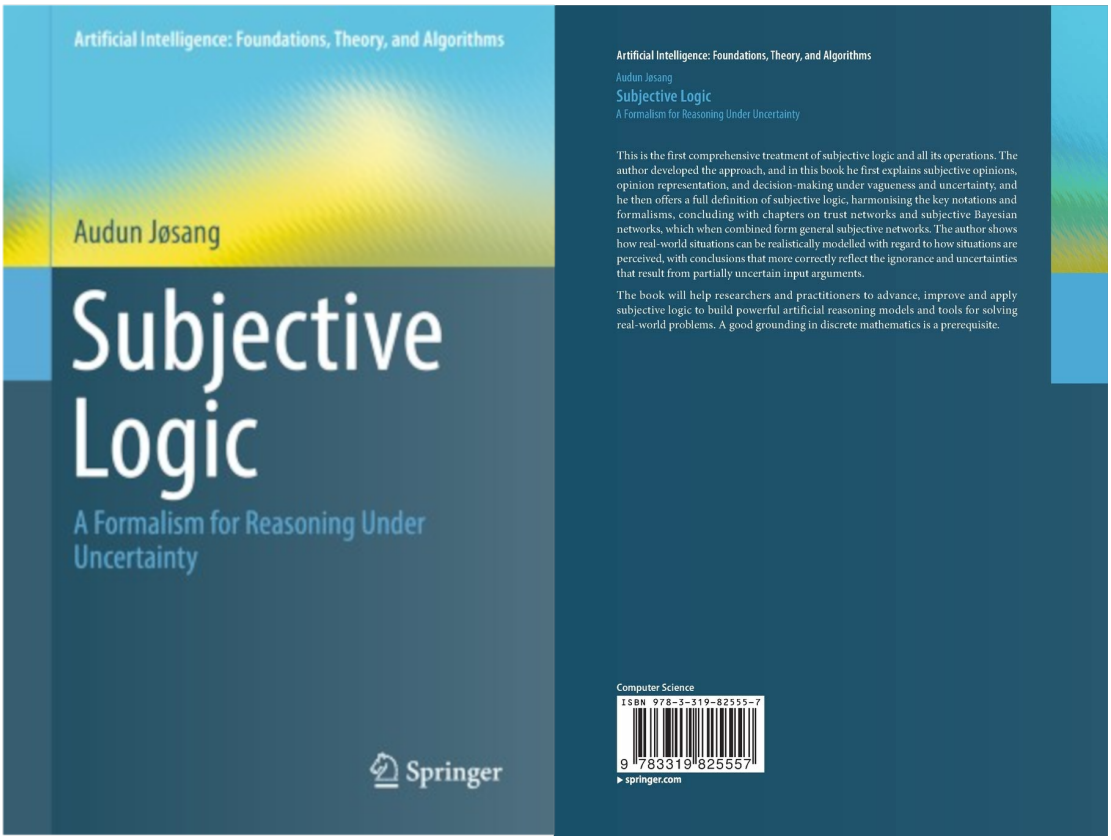

Figure 3.1: Subjective Logic book cover

3.1 Opinions

Subjective logic tries to formalize the description of stochastic phenomena in which the probability of an event is not even known.

It is an extension of binary logic and probabilistic logic which are sometimes not sufficient to explain some human behavior, as demonstrated by the *Ellsberg Paradox* [27].

The Subjective Logic can be summarized in the following scheme:

- **Binary logic** [28]: Each event or proposition E is either **True** or **False**.
- **Probabilistic logic** [29]: The state of a sentence is not known and it is introduced the concept of probability, which assign a number that express "how likely" the statement is True.

$$0 < p(E) < 1 \tag{3.1}$$

- **Subjective logic** [3]: Also the probability is subject to uncertainty, so the statement of a sentence has to contain the information of the probability of the event and the confidence about the probability.

This is done by introducing the concept of **Opinion**.

Definition (Opinion). Let $\mathbb{X} = x, \bar{x}$ be a binary domain with stochastic variable $X \in \mathbb{X}$. A binomial opinion about x is the set of numbers $w = (b, d, u, a)$, where the constraint:

$$b + d + u = 1 \tag{3.2}$$

is satisfied, and where the respective parameters are defined as:

- b: belief mass in support of x being TRUE
- d: disbelief mass in support of x being FALSE
- u: uncertainty mass in support of x being TRUE
- a: base rate, i.e. prior probability of x without any evidence

Except for the base rate (that is usually set 0.5), which is a prior knowledge, an opinion can be represented by a point in a 3-axis Barycentric plot [30], as shown in fig 3.2.

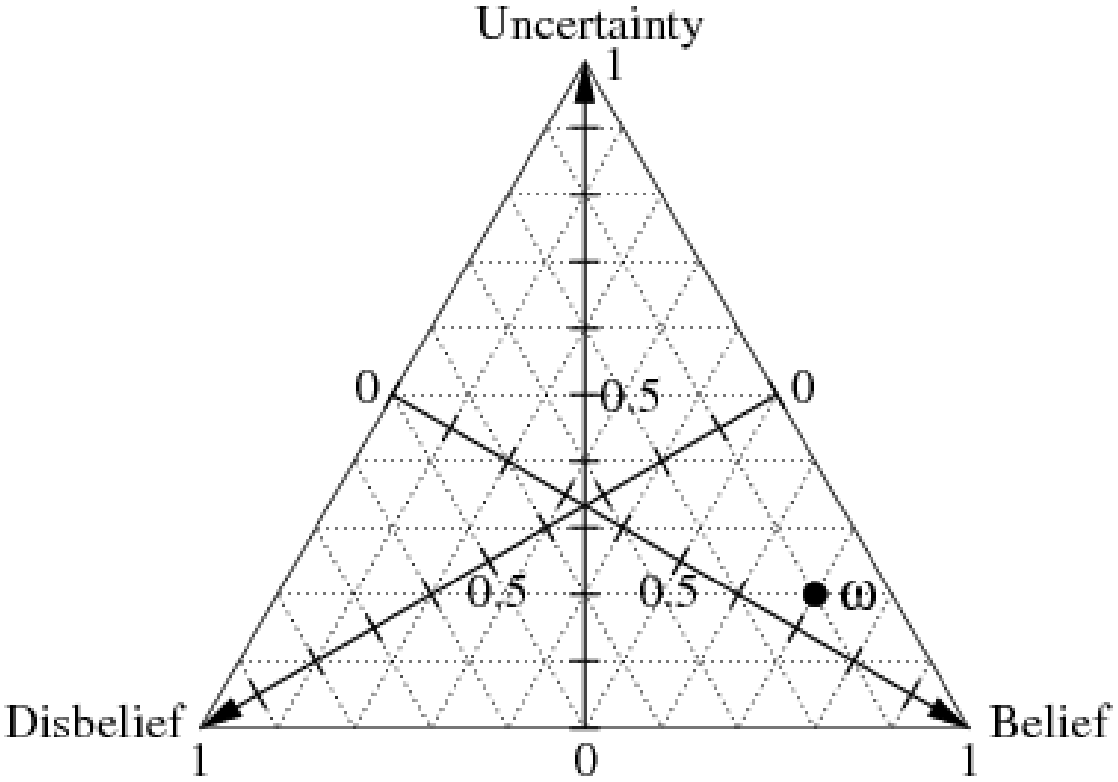

Figure 3.2: Barycentric triangle visualization of binomial opinion

Those just described are the binary opinions, which express the state of knowledge of a binary state. Opinions can be generalized in case the event can be multiclass.

3.2 Mapping with Beta distribution

As already mentioned, uncertainty represents ignorance about the probability of a stochastic event. The lack of information on probability can be formalized through a probability distribution of probabilities. These types of distributions must be defined on the probability domain, therefore in the range [0,1]. A distribution often used to describe probabilities or fractions is the Beta distribution (eq 2.9), due to its flexibility. Audun Josang showed how a objective mapping between an opinion and a Beta [3, p. 24] distribution is possible. The mapping is given by the following comparison between the expectation value and the variance of the distribution and the belief and the uncertainty of the opinion:

$$\begin{cases} E[x] = b + \frac{u}{2} \\ Var[x] = \frac{E[x](1-E[x])u}{w+u} \end{cases} \tag{3.3}$$

These relations, through the relations 2.13 and 2.14, lead to the following mapping with the parameters of the beta distribution:

$$\begin{cases} b = \frac{2\alpha - 1}{2\alpha + 2\beta} \\ d = \frac{2\beta - 1}{2\alpha + 2\beta} \\ u = \frac{1}{\alpha + \beta} \end{cases} \tag{3.4}$$

Notice that we have chosen the value $w = 1$, which is not the only possible choice, as explained in the book itself.

The figure 3.3 shows the mapping between the Beta parameters and the belief and uncertainty values of the associated opinion.

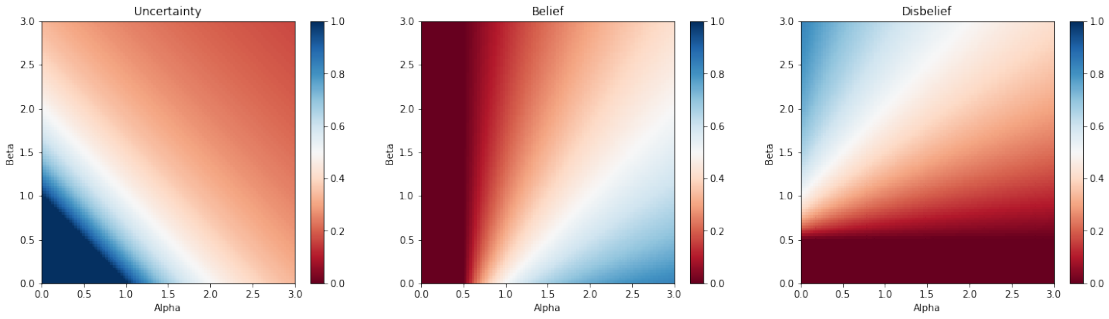

Figure 3.3: Mapping between Beta and opinion parameters

From the following figure (3.4) we notice that the belief value shifts the peak of the distribution towards the value one, while the uncertainty widens the distribution. This is consistent with the interpretation of the parameters of opinion.

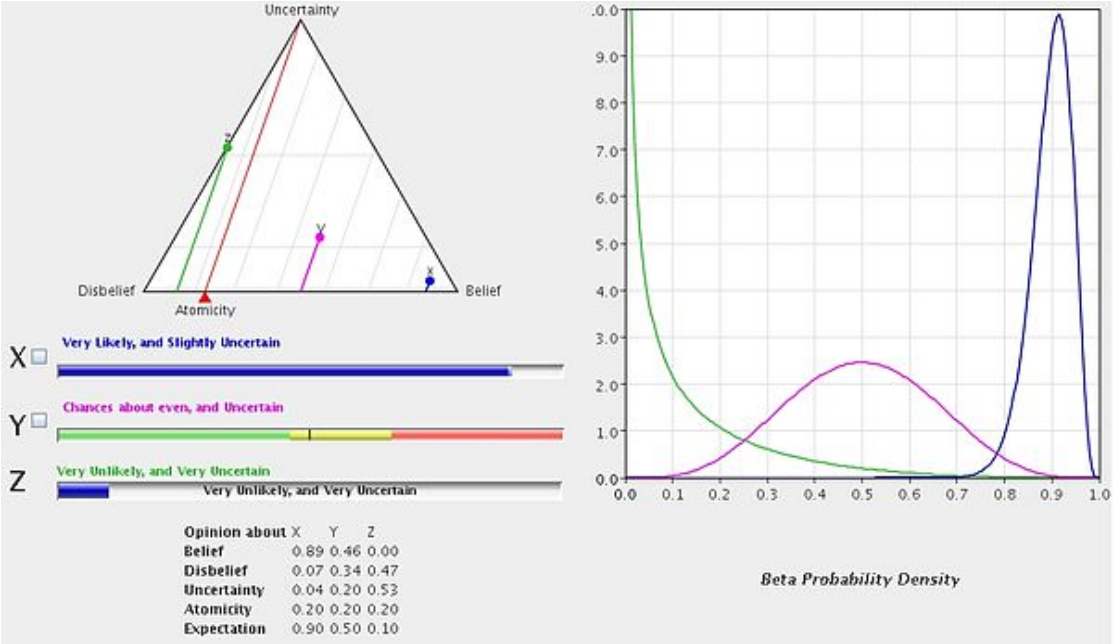

Figure 3.4: Beta distribution for 3 different opinions. In this case it is used also a base rate different from 0.5 and w=2.

This mapping falls into the domain of b and u only for $\alpha \geq 1$ and $beta \geq 1$, and therefore excludes part of the domain of the Beta distributions, as can be seen from the darker, and therefore negative, areas of figure 3.3). This is a problem when it will be applied to neural networks as shown in sect. 4.3.1.

Regions of space for which a Beta distribution exists that is mapped to an opinion that has parameters less than 0 and/or greater than 1 are called *Improper Opinions*.

The mapping between Beta Distributions and Opinions is not unique, you can use other mappings that actually extend the space of opinions, changing the interpretation of Belief, Disbelief and Uncertainty.

In any case, an Improper Opinion can be seen as a case of extreme certainty in which the Subjective Logic is no longer necessary and a discussion with Probabilistic Logic is sufficient.

Chapter 4

The new metric

This chapter contains the main ideas of the thesis. In particular some empirical mathematical properties observable in ANN's output are shown. For each observation some proves, clues or validity limits are provided. In this chapter are also discussed all the implications of the empirical assumptions made. The aim of this mathematical properties is to build a new metric very useful in task in which we don't just want a yes or no answer from the model, but instead a useful probability that we need to integrate to other information in order to make a decision.

4.1 Neural network as probability distributions

First it is needed to specify some assumption and notation for the concept that this chapter will discuss.

The two classes of the binary classification are called A and B and the number of element of the classes are, respectively, N_A and N_B. So the probability of an element X to belong to a class is given by:

$$P(X \in A) = 1 - P(X \in B) = \frac{N_A}{N_A + N_B} \tag{4.1}$$

Now consider the element X belonging to class A and B separately, they form two empirical distributions (eq 2.17), one for each class. So they can be considered as outcomes of two different random variables whose domain is the multidimensional sample space χ. In particular, the stochastic process assign to a sample X a position x in the space χ, extracted by the distributions $F_A(x)$ or $F_B(x)$.

Eq 1.3 shows that the output of the network ending with a Softmax layer is composed by two numbers relative to the two classes. In this chapter the notation $o(x)$ is used to indicate the class A output of the neural network to the input x. The value for class B is just $1 - o(x)$.

All the neural network does is a non linear transformation of a point x in a one-dimensional domain by the function $o(x)$.

It is very important to underline the distinction between the transformation $o(x)$, which transform every point in the multi-dimensional space χ in the one-dimensional space [0,1], and the stochastic variable $o(X)$ that assumes a random value in [0,1] because X assumes a random value $x \in \chi$. So $o(x)$ is completely deterministic, whereas $o(X)$ is stochastic because the position x of X has not yet been extracted.

It is possible to define the two probability distributions of the variable $o(X)$:

$$\begin{aligned} f_A(p)dp &= P\Big(p < o(X) < p + dp | X \in A\Big) \\ f_B(p)dp &= P\Big(p < o(X) < p + dp | X \in B\Big) \end{aligned} \tag{4.2}$$

By varying the parameter, the network modify the transformation $o(x)$ and so the distribution of $o(X)$. And this training process is finalized to minimizing the cross entropy (and so the likelihood thanks to eq 2.22) between empirical probability of the dataset and the value $o(x)$.

The output $o(x)$ of the network fulfill the property of a probability $0 \leq o(x) \leq 1$ and from figure 1.5 it is easy to understand that the network, in order to minimize the loss function, is incentivized to return value closer to 1 if it "thinks" that the sample belongs to class A, and closer to 0 for class B. However, from what has been discussed so far, there is no evidence that the value $o(x)$ can be interpreted as a probability of belonging to class A. As already said, $o(x)$ is deterministic and it is impossible to repeat the experiment x, which is just a tuple of coordinates. It is possible to "repeat" the sample X, but it will assume another position y and so we will get $o(y)$ instead of $o(x)$. The only probability that $o(x)$ can represent is the probability to find an element of class A in point x, given that a sample is found in that point.

In fact, assuming to extract enough ($N \to \infty$) samples from both classes, it is possible to calculate the probability of finding a sample of class A in a space dx around point x using the frequentist definition of probability (section 2.1):

$$P(X \in A | x) = \frac{F_A(x)dx}{F_A(x)dx + F_B(x)dx} = \frac{F_A(x)}{F_A(x) + F_B(x)} \tag{4.3}$$

4.2 Perfect calibration assumption

From the discussions in the previous section it might seem obvious that, during the training, the network try to fit the probability map expressed in formula 4.3 with $o(x)$, but it is not yet proven nor is it clear why it should.

If it were so, $o(x)$ could be interpreted as the frequentist probability that $X \in A$ and so the following equality would be true:

$$P(X \in A | o(x) = p) = p \tag{4.4}$$

This is the definition of *perfect calibration* [31].

4.2.1 Perfect calibration optimality

It is possible to show that in fact the perfect calibration condition is the optimal strategy for the network to minimize the expectation value of loss function.

Consider a binary stochastic variable:

$$X = \begin{cases} 1 & \text{with probability p} \\ 0 & \text{with probability 1-p} \end{cases} \tag{4.5}$$

And a penalty:

$$L(X, o) = \begin{cases} -\log o & \text{if } X = 1 \\ -\log(1 - o) & \text{if } X = 0 \end{cases} \tag{4.6}$$

Where o is the output of an agent that want to minimize the total penalty over multiple repetition of the experiment.

The rational strategy is to chose the value of o that minimize the expectation value of the penalty:

$$E[L(o)] = -p \log o - (1 - p) \log(1 - o) \tag{4.7}$$

Which is a convex function so is very easy to calculate the optimal value:

$$\begin{gathered} \frac{dE[L(o)]}{do} = -\frac{p}{o} - \frac{1-p}{1-o} \\ \frac{dE[L(o)]}{do} = 0 \Rightarrow o = p \end{gathered} \tag{4.8}$$

It is useful to emphasize that this result is only due to the penalty being equal to the cross entropy loss function. Indeed, using the more simple penalty given by the distance between the prediction and the stochastic value it would be obtained a different result:

$$L(X, o) = \begin{cases} 1 - o & \text{if } X = 1 \\ o & \text{if } X = 0 \end{cases} \tag{4.9}$$

$$E[L(o)] = p(1-o)^2 + (1-p)o^2 \tag{4.10}$$

$$\frac{dE[L(o)]}{do} = -p + (1-p) = 1 - 2p \tag{4.11}$$

$$\begin{cases} \frac{dE[L(o)]}{do} > 0 & \text{if } p < \frac{1}{2} \\ \frac{dE[L(o)]}{do} < 0 & \text{if } p > \frac{1}{2} \end{cases} \tag{4.12}$$

$$\frac{dE[L(o)]}{do} = 0 \Rightarrow \begin{cases} o = 0 & \text{if } p < \frac{1}{2} \\ o = 1 & \text{if } p > \frac{1}{2} \end{cases}$$

(4.13)

So in this case the network would learn to "shoot" 1 every time is slightly more likely that $X = 1$, and 0 otherwise, which is a very risky strategy. Instead a rational agent that wants to maximize the logarithm of the error (like in the cross-entropy penalty) tries to infer the probability of event X and "bets" an amount of risk equal to that probability. This situation recalls the definition of subjective probability mentioned in section 2.1. In *game theory* this behaviour is called *risk aversion*[32], because the agent prefers not to gain as much, rather than risk losing. This is also related to the concept of maximum entropy.

4.2.2 Perfect calibration output distributions

Assuming perfect calibration, it is possible to conclude an interesting property of the output distribution functions in formulas 4.2.

From Bayes theorem:

$$P(X \in A | o(X) = p) = P(o(X) = p | x \in A) \frac{P(x \in A)}{P(o(X) = p)} = \tag{4.14}$$

$$\frac{f_A(p)}{f_A(p)\frac{N_A}{N_A+N_B} + f_B(p)\frac{N_B}{N_A+N_B}} \frac{N_A}{N_A + N_B} == \frac{f_A(p)}{f_A(p)N_A + f_B(p)N_B} = p \tag{4.15}$$

This implies:

$$f_B(p) = \frac{N_A}{N_B} f_A(p) \frac{1-p}{p} \tag{4.16}$$

From now on, it is considered only the case of balanced classes, but it is possible to generalize to the non balanced case. So imposing $N_A = N_B$:

$$f_B(p) = f_A(p)\frac{1-p}{p} \tag{4.17}$$

Last equation implies that the two probability distribution are related and not independent. With the purpose of finding a family of distributions to which they belong, it is useful to extrapolate the infinitesimal (or infinite) part of the function for $p \to 0$ and $p \to 1$:

$$\begin{aligned} f_A(p) &= g_A(p)p^{\alpha_A - 1}(1-p)^{\beta_A - 1} \\ f_B(p) &= g_B(p)p^{\alpha_B - 1}(1-p)^{\beta_B - 1} \end{aligned} \tag{4.18}$$

From 4.17 it is easy to show that:

$$\begin{cases} g_A(p) = g_B(p) \\ \alpha_B = \alpha_A - 1 \\ \beta_B = \beta_A + 1 \end{cases} \tag{4.19}$$

Where new constraints are introduced: $\alpha_A > 1$, $\beta_B > 1$.

From now on, will be used the following notation: $\alpha_A \to \alpha$, $\beta_A \to \beta$.

In summary, this section showed that, under perfect calibration assumption, the probability distributions of the output of the network can be written in the form:

$$\begin{aligned} f_A(p) &= g(p)p^{\alpha - 1}(1-p)^{\beta - 1} \\ f_B(p) &= g(p)p^{\alpha - 2}(1-p)^{\beta} \end{aligned} \tag{4.20}$$

With:

$$\begin{cases} g(0) \neq 0 \\ g(1) \neq 0 \\ \lim_{p\to 0} g(p) \neq \infty \\ \lim_{p\to 1} g(p) \neq \infty \end{cases} \tag{4.21}$$

This family of distribution has infinite degree of freedom because it is not possible to make assumptions on $g(p)$ other the normalization of the distributions.

4.2.3 Calibration measure

A neural network in a real training case can never achieve perfect calibration. For this it is necessary to define and use parameters to evaluate how well the network is [31].

Calling up the 4.4 definition:

$$P(X \in A | o(x) = p) = p \tag{4.22}$$

Define the *expected calibration error* as follows:

$$CalErr = E\Big[\Big|P(X \in A | o(x) = p) - p\Big|\Big] \tag{4.23}$$

Or, alternatively, the *expected calibration squared error*:

$$CalSqErr = \sqrt{E\Big[\Big(P(X \in A | o(x) = p) - p\Big)^2\Big]} \tag{4.24}$$

Utilizzando le distribuzioni definite nell'eq 4.2 possiamo riscrivere i seguenti parametri come:

$$CalErr = \int_0^1 \Big|\frac{f_A(p)}{f_A(p) + f_B(p)} - p\Big| \frac{f_A(p) + f_B(p)}{2}\, dp = \frac{1}{2}\int_0^1 \Big|f_A(p)(1-p) - p f_B(p)\Big|\, dp \tag{4.25}$$

It is easy to show that the last equation vanishes in the case of the perfect calibration hypothesis described in section 4.2.2.

The study of the calibration error in the next section is very useful, where the output distributions are modeled by a few parameters.

From the application point of view, to calculate this integral it is sufficient to divide the probability interval from 0 to 1 into a certain number of bins, and to calculate the integral numerically as a summation, using the empirical distributions of the train set, or alternatively of the test set.

For the purpose of this thesis, the calibration error is very useful because it allows us to understand, with a good approximation, if the output of the neural network can be reasonably interpreted as a frequentist probability.

4.3 Beta distribution assumption

This section shows how the output of a neural network can be modeled via a Beta distribution. This modeling is very useful for constructing the new metric proposed by this thesis. Since the assumption is totally empirical, some considerations regarding its validity and limitations will also be indicated.

4.3.1 ANN output with Beta distribution assumption

Taking into consideration the models discussed in the numerical results (cap 5), it can be seen

Taking into consideration a well-trained artificial neural network for classification (avoiding overfitting) and with a good level of calibration we can note, in the examples considered in chap 5, that the distribution of the model's outputs on the database takes a form which is well suited to the Beta distribution.

The figure 4.1 shows one of the examples of the neural networks analyzed. These are the histograms constructed by taking the outputs of the various samples of the train and the test set. The samples form two clouds corresponding to the two classes. Class A has a label corresponding to 1, while class B to 0. This implies that the model's ability to successfully classify the samples is proportional to "how much" the two clouds move away from each other, decreasing their intersection surface.

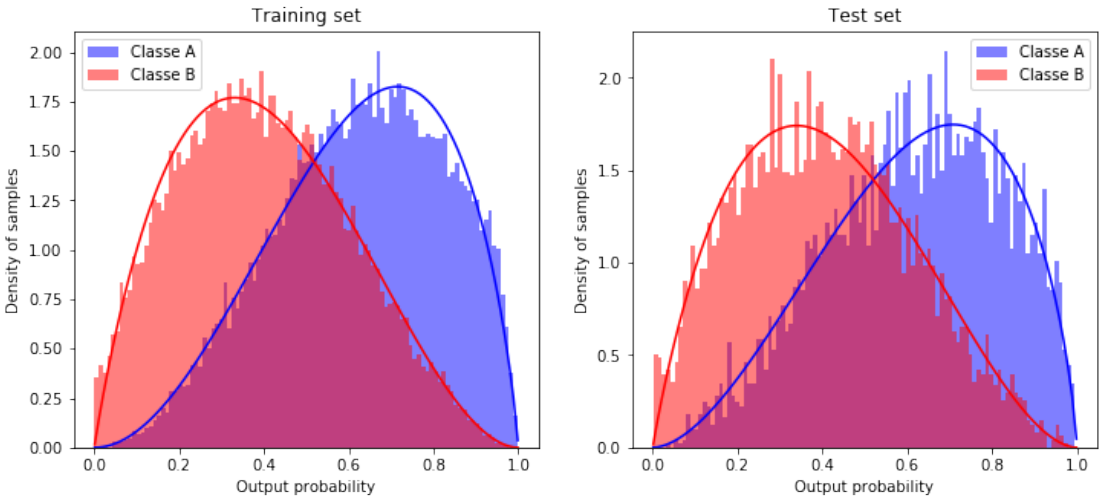

Figure 4.1: Histograms of the outputs (divided in two classes) of the neural network for train and test set, fitted by beta distributions.

As seen in the 2.3, given an empirical distribution and a parametric family of distributions, it is possible to find the distribution that best represents the data. In the case of the Beta family this can be done through the algorithm shown in the appendix **??**.

The result of the fitting is shown in figure 4.1, and more generally in the section 5.3.

In the cases taken into consideration, it is noted that the fitting manages to describe the data well.

$$
\begin{aligned}
f_A(p) &= \frac{1}{B(\alpha_A, \beta_A)} p^{\alpha_A - 1}(1-p)^{\beta_A - 1} \\
f_B(p) &= \frac{1}{B(\alpha_B, \beta_B)} p^{\alpha_B - 1}(1-p)^{\beta_B - 1}
\end{aligned}
\tag{4.26}
$$

From the parameters α and β of the two distributions, it is possible to trace the performance of the model. In particular, the model will be able to better classify the elements of the set the more the two distributions are separated, one pushed towards the value 0, and the other towards the value 1, which correspond to the two classes. This happens when the *alpha* parameter for class A and the *beta* parameter for class B decrease towards 0.

It is possible to note that the beta distribution is a particular case of the family of distributions obtained in the section 4.20. In particular, we add a degree of freedom because the parameters α and β are both free, but we impose $g(p) = 1$, which eliminates infinite degrees of freedom from the distribution. By freeing the α and β parameters, you give up the perfect calibration constraint. It is therefore possible to calculate, by applying the formula 4.25 applied to the beta distribution, the calibration error.

the error will depend on 4 parameters, which can be reduced to 2 by assuming the symmetry of the distributions of the two classes.

Let's assume $f_A(p) = f_B(1-p) \Rightarrow \alpha_A = \beta_B = \alpha, \beta_A = \alpha_B = \beta$

$$
CalErr = \frac{1}{2} \int_0^1 \left| f_A(p)(1-p) - p f_B(p) \right| dp \tag{4.27}
$$

Questo integrale può essere calcolato sdoppiando il modulo e usando le proprietà della funzione Beta incomplet [24], ottenendo:

$$
CalErr = \frac{\left| 2B_{\frac{1}{2}}(\alpha, \beta+1) - 1 \right|}{B(\alpha, \beta)} \tag{4.28}
$$

Below is the graph that shows how the calibration error varies according to the parameters α and β, from which the straight line also corresponds to the perfect calibration (error equal to zero).

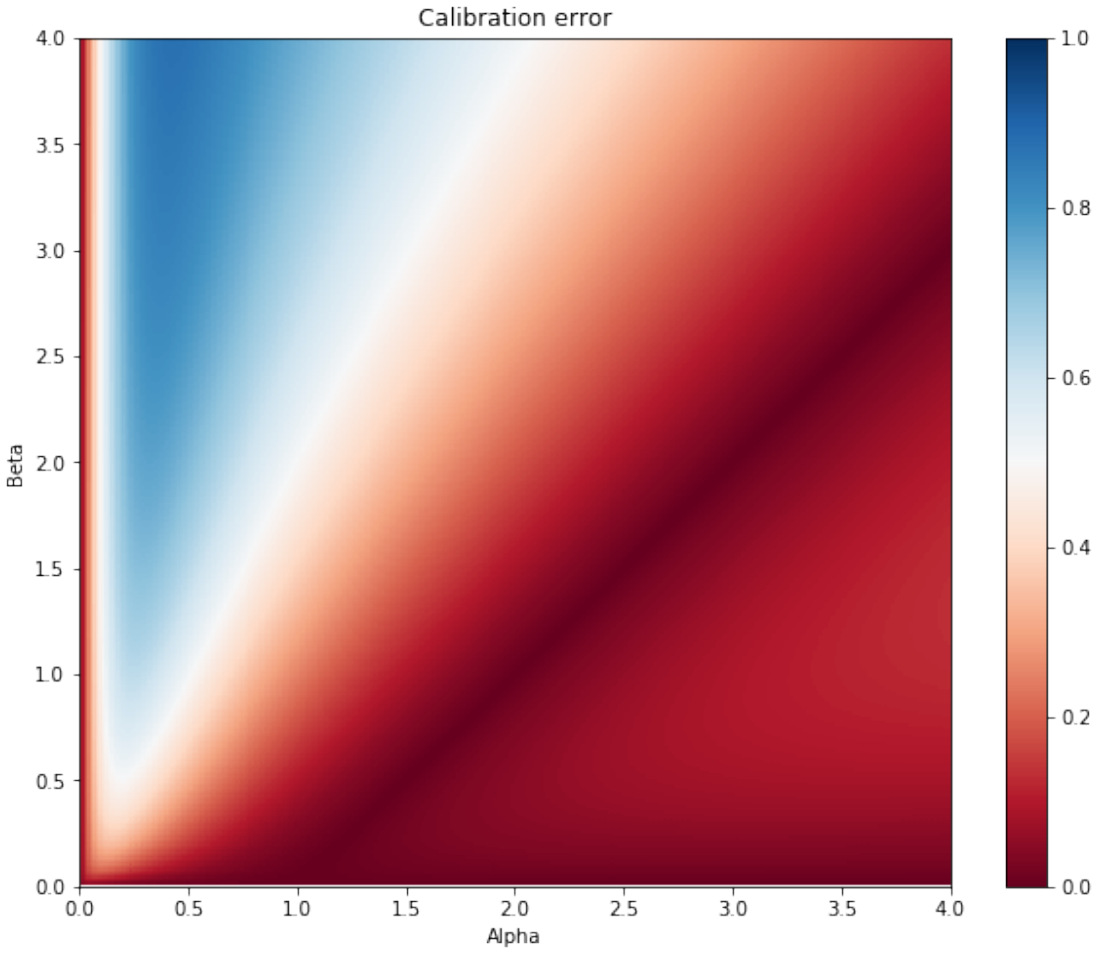

Figure 4.2: Heatmap of the calibration error with respect to α and β values..

Describing the output of the neural network with a Beta distribution is very useful for building a new metric because it allows to synthesize its performance in the 4 extracted parameters. Therefore it is useful to try to understand the validity and the limitations of the assumption that a well trained and calibrated neural network should distribute the samples according to a Beta distribution.

In fact, we have seen that, during training, the neural network adjusts its parameters in order to minimize the loss function in eq 1.4 and that the Beta distribution is the distribution that maximizes entropy for constant values of $log(x)$ and $log(1-x)$.

It is possible to make another observation starting from the link between the Beta and Gamma distribution families, namely that a random variable distributed according to a Beta distribution can be obtained from the following combination of random variables distributed according to a Gamma distribution:

$$\gamma(x) = \frac{x^{\alpha-1}e^{-\beta x}\beta^{\alpha}}{\Gamma(\alpha)} \tag{4.29}$$

To prove this, consider the joint PDF:

$$f_{X,Y}(x,y) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}y^{\beta-1}e^{-(x+y)} \tag{4.30}$$

Perform the transformation:

$$\begin{aligned} U &= \frac{X}{X+Y} \\ V &= X+Y \end{aligned} \tag{4.31}$$

$$\begin{aligned} X &= VU \\ Y &= V(1-U) \end{aligned} \tag{4.32}$$

The Jacobian of the transformation is equal to V so the joint probability of U and V is:

$$f_{U,V}(u,v) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)}v^{\alpha+\beta-1}e^{-v}u^{\alpha-1}(1-u)^{\beta-1} \tag{4.33}$$

and hence U and V are independent (because the pdf factors over u and v) with:

$$\begin{aligned} V &\sim Gamma(\alpha+\beta,1) \\ U &\sim Beta(\alpha,\beta) \end{aligned} \tag{4.34}$$

Now noticing that the last layer of the neural network consists of the softmax function (eq 1.3) and the similarity with the transformation in eq 4.31 it is possible to deduce that the last layer before the softmax must produce some random variables z_A and z_B must be distributed according to a log-gamma [33] distribution:

$$l\gamma(x) = Ke^{\alpha z - e^{z}} \tag{4.35}$$

Where K is the normalization constant. The Gamma distribution also maximizes entropy with two mean values being equal: x and $log(x)$.

It is difficult to rigidly demonstrate that the training of a neural network has as its optimal point a state in which the layers are distributed according to the log-gamma and the beta, but this is what is observed in many cases.

However, this behavior starts to be violated in case of strong overfitting, and loses its meaning in case the two classes are so easily separable that an accuracy very close to 100/100 is reached.

It is worth noting that the Beta distribution has two free parameters and is capable of representing a large spectrum of empirical distributions, which is why it is often used to describe probabilities of probabilities.

4.3.2 Mapping with classical metrics

As discussed in the 1.2 the main classical metrics can be calculated starting from the output of the neural network on each sample of the dataset.

By modeling the neural network with the Beta distribution, it is possible to calculate the classical metrics also through integrals of the Beta distribution, by making the limit to the continuum of the operations to calculate these values.

The following are the main metrics calculated using Beta distribution:

- True positive:

$$TP = \frac{N}{2} \int_{\frac{1}{2}}^{1} \beta_A(x)\, dx = \frac{N}{2}\Big(1 - I_A\Big(\frac{1}{2}\Big)\Big) \tag{4.36}$$

- True negative:

$$TN = \frac{N}{2} \int_{0}^{\frac{1}{2}} \beta_B(x)\, dx = \frac{N}{2} I_B\Big(\frac{1}{2}\Big) \tag{4.37}$$

- False positive:

$$FP = \frac{N}{2} \int_{\frac{1}{2}}^{1} \beta_B(x)\, dx = \frac{N}{2}\Big(1 - I_B\Big(\frac{1}{2}\Big)\Big) \tag{4.38}$$

- False negative:

$$FN = \frac{N}{2} \int_{0}^{\frac{1}{2}} \beta_A(x)\, dx = \frac{N}{2} I_A\Big(\frac{1}{2}\Big) \tag{4.39}$$

- Accuracy:

$$Acc = \frac{1}{2} \int_{\frac{1}{2}}^{1} \beta_A(x)\, dx + \frac{1}{2} \int_{0}^{\frac{1}{2}} \beta_B(x)\, dx = \frac{1}{2}\Big[1 - I_A\Big(\frac{1}{2}\Big) + I_B\Big(\frac{1}{2}\Big)\Big] \tag{4.40}$$

- ROC AUC area:

$$Roc = \int_{0}^{1} I_B\Big(I_A(x)\Big)\, dI_A(x) = \int_{0}^{1} I_B\Big(I_A(x)\Big)\beta_A(x)\, dx \tag{4.41}$$

Where $I_A(x)$ is the incomplete Beta function [24] calculated at point x with the parameters of the distribution relative to class A.

This mapping is very useful for evaluating fitting with the Beta distribution. In fact, to measure the quality of the fitting, the Kullback-Leibler Divergence is used, as shown in the 2.3 section but, in the context of neural networks, it is natural that an excellent indicator of the fitting can be given by the error on classic metrics, such as accuracy or ROC AUC curves. In the case of the figure 4.3, it is easy to see that the fitting reproduces the real ROC AUC Curve very well.

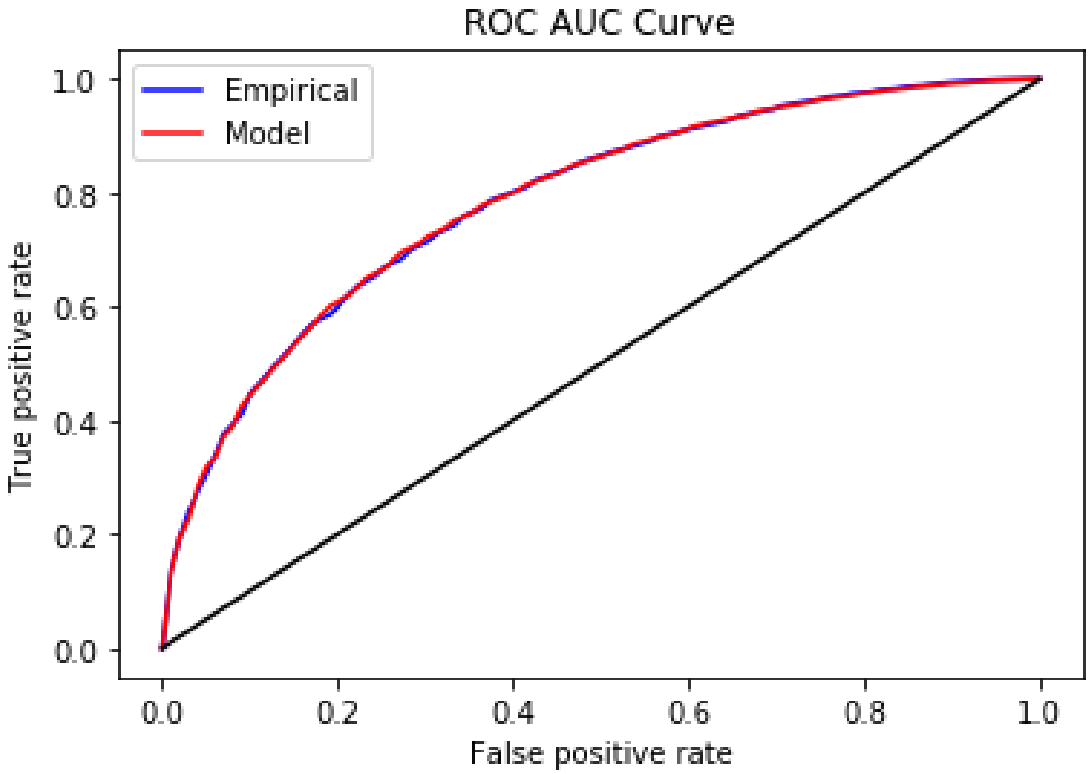

Figure 4.3: ROC AUC CURVE calculated with empirical distribution and Beta distribution distribution.

Assuming, for simplicity, that the two classes are distributed in a specular way, it is easy to show that the parameters of the two classes will be specular: $\alpha_A = \beta_B, \beta_A = \alpha_B$.

we can recalculate the accuracy from the eq 4.40 using the last approximation obtaining:

$$Acc = 1 - I\left(\frac{1}{2}\right) \tag{4.42}$$

In which it is recalled that $I(0.5)$ depends on the parameters α and β. This dependency can be visualized by the following heat map (fig. 4.4):

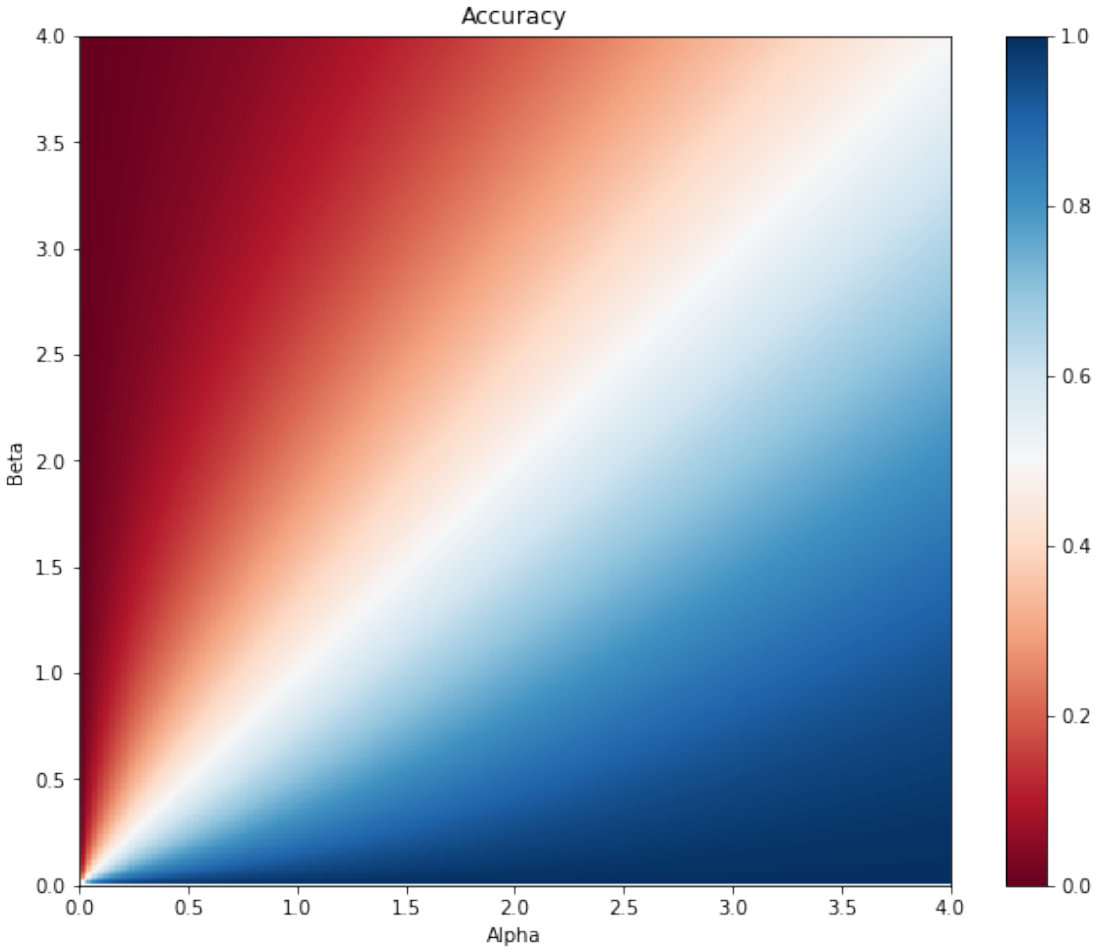

Figure 4.4: Accuracy calculated for each value of Beta parameters.

4.3.3 Interpretation with Subjective Logic

As we have just seen, the output of a neural network can be successfully described by a Beta distribution for each class. This creates the ability to map to subjective logic, whose views can equally be described by a Beta distribution.

In subjective logic, an opinion expresses the level of uncertain or subjective knowledge about a statement.

In the case of neural networks, the probability density modeled by Beta refers to the probability that the neural network returns a certain output value, knowing that the element belongs to a given class.

So it is an "a posteriori" opinion because it is necessary to know belonging to the class as a priori information.

The value of belief (b) tells us how likely it is that the output of the neural network is close to the true class, while the uncertainty gives us indications on the variability of this value.

It is possible to interpret these two values using the concepts of likelihood and confidence [3, p. 48].

To make these values more humanly interpretable, it is useful to identify discrete qualitative levels and, as will be shown in the chapter 6.1.2, it is possible to create a dashboard that describes in detail all the properties of a model, actually going to build a new, more complex and interpretable metric to add to the classic ones, which is the final aim of this thesis.

In addition to the definition of a new metric, the connection with the subjective logic allows to use all the decision criteria under uncertainty developed in the theory, that is, it allows to use a very precise protocol based on the data of the neural network metrics. These issues are not dealt with in this thesis.

Finally, it is emphasized that it is possible to apply this mapping also in the case of problems with non-binary classes and multiclass problems, using hyperopinions or multinomial opinions, which can be mapped with the distribution of Dirichlet [34], which is the generalization multidimensional Beta distribution.

4.4 Beta AND calibration assumptions case

In this last section, the case in which both the assumptions of the 4.2 and of the 4.3 are accepted.

It is emphasized that this configuration empirically does not occur, but is subject to an error. For this reason the content of this section has the sole purpose of completing the picture mathematically, but is not used for the application purpose of creating a new metric applicable in real cases such as medical diagnosis.

Let's summarize the two hypotheses we take into consideration (eq 4.20 and eq 4.26):

$$\begin{aligned} f_A(p) &= g(p)p^{\alpha-1}(1-p)^{\beta-1} = \frac{1}{B(\alpha_A,\beta_A)}p^{\alpha_A-1}(1-p)^{\beta_A-1} \\ f_B(p) &= g(p)p^{\alpha-2}(1-p)^{\beta} = \frac{1}{B(\alpha_B,\beta_B)}p^{\alpha_B-1}(1-p)^{\beta_B-1} \end{aligned} \tag{4.43}$$

Comparing the exponents we get:

$$\begin{aligned}
\alpha_A &= \alpha \\
\beta_A &= \beta \\
\alpha_B &= \alpha - 1 \\
\beta_B &= \beta + 1
\end{aligned} \tag{4.44}$$

While comparing the $g(p)$ function with the normalization constant $B(\alpha, \beta)$:

$$\begin{aligned}
&g(p) = B(\alpha_A, \beta_A) = B(\alpha_B, \beta_B) \\
&B(\alpha, \beta) = B(\alpha - 1, \beta + 1) \\
&\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} = \frac{\Gamma(\alpha - 1)\Gamma(\beta + 1)}{\Gamma(\alpha + \beta)} \\
&\Gamma(\alpha)\Gamma(\beta) = \Gamma(\alpha - 1)\Gamma(\beta + 1) \\
&(\alpha - 1)\Gamma(\alpha - 1)\Gamma(\beta) = \Gamma(\alpha - 1)\beta\Gamma(\beta) \\
&\beta = \alpha - 1
\end{aligned} \tag{4.45}$$

We note that from the 4 initial parameters we have introduced such restrictions that only one parametric degree of freedom is left. The final distributions are therefore:

$$\begin{aligned}
f_A(p) &= \frac{1}{B(\alpha, \alpha - 1)} p^{\alpha - 1}(1 - p)^{\alpha - 2} \\
f_B(p) &= \frac{1}{B(\alpha - 1, \alpha)} p^{\alpha - 2}(1 - p)^{\alpha - 1}
\end{aligned} \tag{4.46}$$

In which we also notice the $\alpha \geq 1$ constraint, similarly to what has already happened in section 4.2.2.

In this scenario, the metrics of a neural network boil down entirely to the single free parameter α.

The mapping with accuracy is shown below (using the 4.40 with a change of notation to highlight the parameters):

$$
\begin{aligned}
&Acc = \frac{1}{2}\Big[1 - I_A\Big(\frac{1}{2}\Big) + I_B\Big(\frac{1}{2}\Big)\Big] = \\
&\frac{1}{2}\Big[1 - I_{\frac{1}{2}}(\alpha, \alpha-1) + I_{\frac{1}{2}}(\alpha-1, \alpha)\Big] = \\
&\frac{1}{2}\Big[1 - I_{\frac{1}{2}}(\alpha, \alpha-1) + 1 - I_{\frac{1}{2}}(\alpha, \alpha-1)\Big] = \\
&1 - I_{\frac{1}{2}}(\alpha, \alpha-1) = \\
&1 - I_{\frac{1}{2}}(\alpha-1, \alpha-1) + \frac{0.5^{2\alpha-2}}{(\alpha-1)B(\alpha-1, \alpha-1)} = \\
&\frac{1}{2} + \frac{\Gamma(2\alpha-2)}{(\alpha-1)4^{\alpha-1}\Gamma(\alpha-1)^2}
\end{aligned}
\tag{4.47}
$$

Whose plot is shown below:

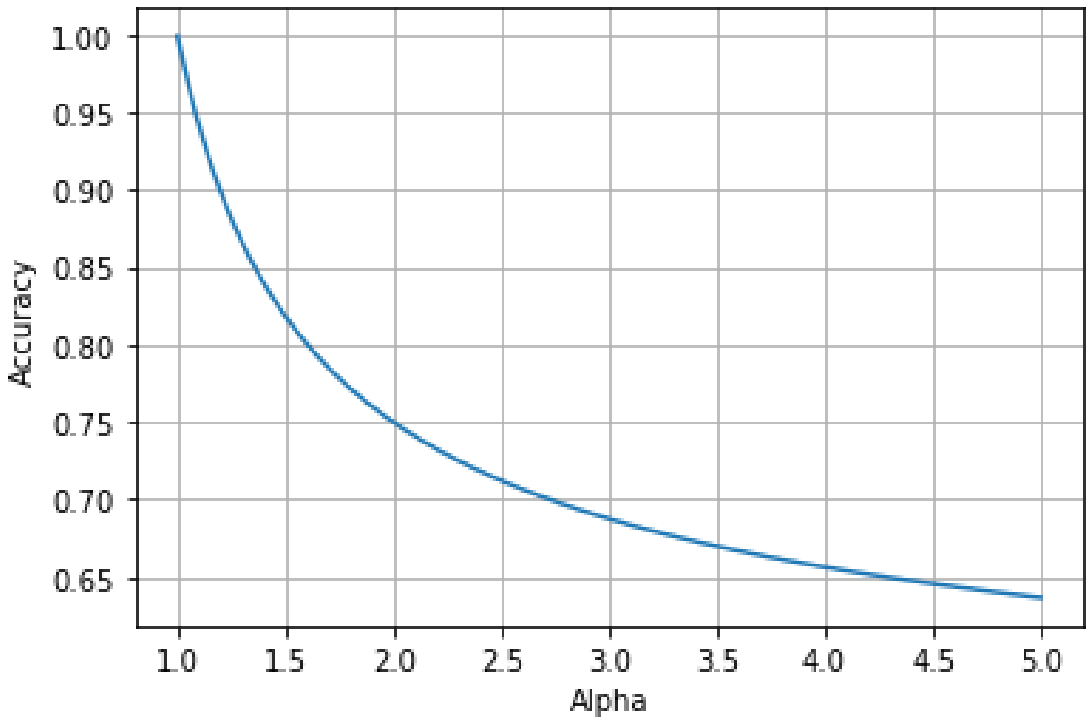

Figure 4.5: Relation between Accuracy and the Beta distribution parameter of the one-parameter metric.

Finally, as regards the mapping with Subjective Logic, we note that having only one parameter also restricts the field of action of the neural network in the opinion space, which has two degrees of freedom.

By inserting this restriction in the subjective logic mapping (eq. 3.4), we get (for class 1):

$$\begin{cases} u = \frac{1}{2\alpha - 1} \\ b = \frac{1}{2} \\ d = \frac{2\alpha - 3}{4\alpha - 2} \end{cases} \tag{4.48}$$

So the belief value will be confined to 0.5, and we note that the uncertainty is inversely proportional to alpha.

This means that a more accurate network decreases the value of the disbelief not in favor of belief, which remains constant, but of uncertainty.

It is therefore useful to plot the curve of the value $1 - d$ as a function of accuracy.

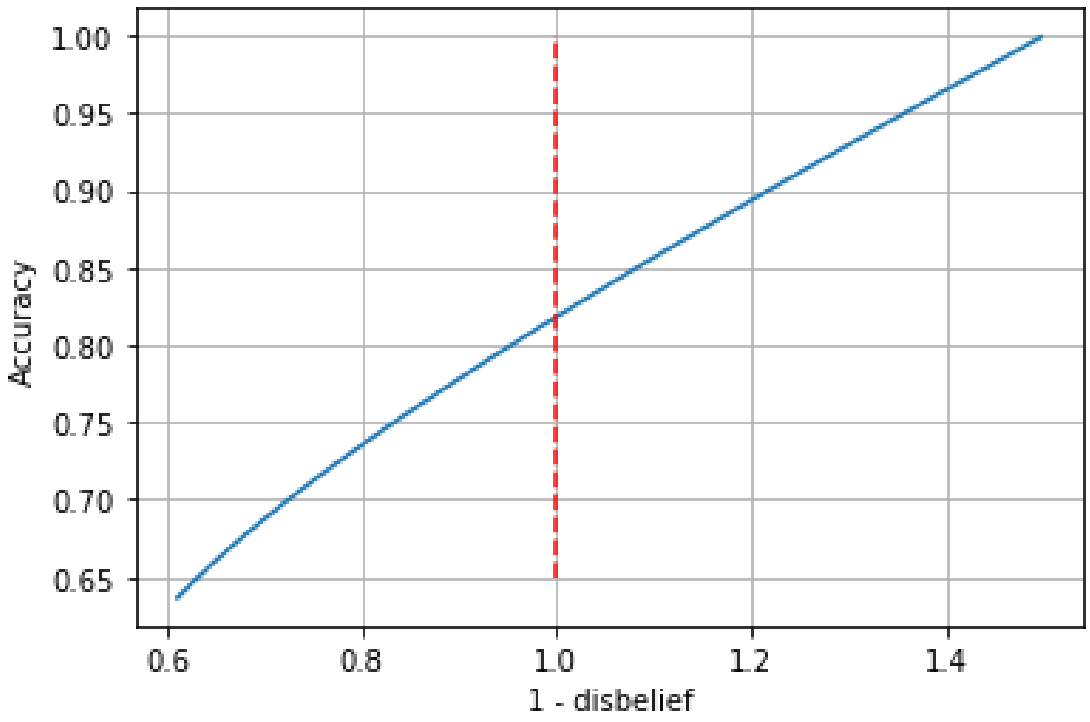

Figure 4.6: Relation between Accuracy and the Opinion parameter of the one-parameter metric.

From this last analysis shown in figure 4.6 one of the limits of this mapping emerges: the dashed line corresponds to the boundary of the opinion space, The part of the line to the right of the graph, which corresponds to all accuracy greater than about 83%, cannot be expressed by Beta distributions attributable to the Opinions of subjective logic and at the same time corresponding to perfect calibration, but

we enter the region of improper opinions, seen in the chapter 3.2. It is possible to use other mappings between Beta distributions and Subjective Logic that improve the problem, pushing beyond the value of 83%, but the problem persists in any case for very high accuracy. This means that one of the two hypotheses (Perfect Calibration and Beta Fitting) must fail. As will be shown in the results section 5.2 and 5.3.2, the calibration remains unchanged as statistical inference begins to be more inaccurate with the Beta distribution.

We conclude that, in any case, the new metric is less effective in the case of very high accuracy, a phenomenon that can be explained by the fact that if a model is able to identify distinctive patterns of the classes, it has no need to distribute the outputs according to a generic distribution of maximum entropy.

Chapter 5

Numerical results

This chapter provides all the empirical evidences of the assumptions made in the previous chapter.

These results testify that the new metric can be adopted in most application cases.

5.1 Setup

5.1.1 Database

The new metric described in this thesis aims to provide the end user of the predictive algorithm with more information on the meaning of the neural network output. This type of application is very useful in the field of medical diagnosis, where the analysis of the neural network is not the only information in the hands of the doctor and the decision-making process involves many responsibilities.

For this reason, the tests to verify the applicability of the hypotheses underlying the new metric have been applied to concrete cases similar to those that may arise in the medical field.

The problems solved concern image processing tasks, the most common field of application for neural networks and which is very frequently addressed in the medical field (x-rays, ultrasound scans, etc.)

2 MNIST databases were used, one containing images of digits 0-9 handwritten [35], and the other black and white images of photos of [36] clothes.

The reason for this choice lies in the fact that in the medical field one often has to deal with images of various types, but at the same time there is no need or possibility to test this new metric on the many and difficult real medical tasks, because the purpose of the discussion is to demonstrate that the new metric is applicable for a certain range of models, not to demonstrate its universal applicability in every

possible case.

Since the difficulty of the tasks depends on many variables, including the database, the "recognizability" of the classes and the size of the network, it was decided to carry out the study at different levels of difficulty, adjustable by inserting noise in the images.

In fact, square "spots" have been inserted that cover part of the images. The analysis was done on 5 different noise levels. Level 1 contains few spots, while level 5 contains a large number of randomly placed spots in the image, which often make it difficult to recognize the object even with the naked eye.

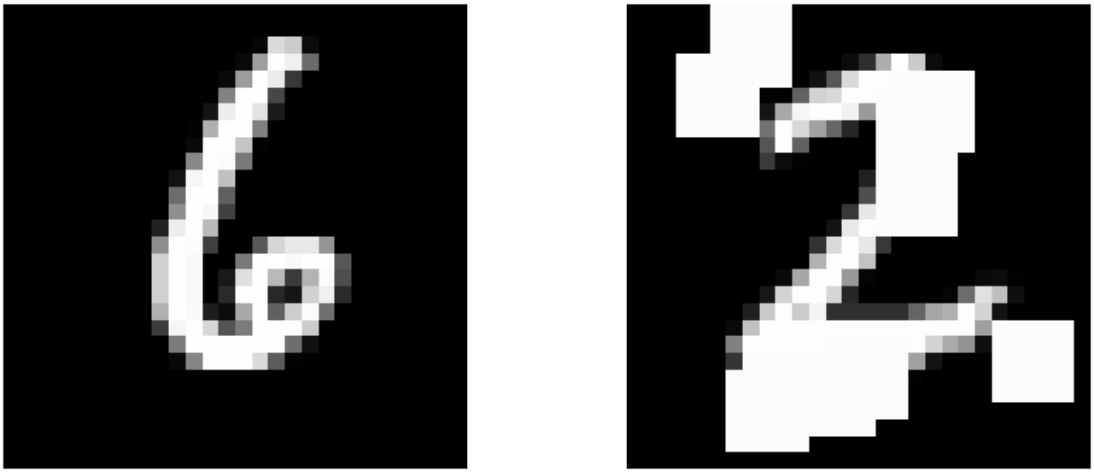

Figure 5.1: Example of a dataset image with (right) and without (left) artificial noise.

The type of classification for which this metric applies is binary classification, so the database has been divided into 2 classes.

Regarding the figures, the state of the art on the classification of this database has reached a very high accuracy [37][38]. From this it is clear that the classification of the figures is a problem that can be easily solved by neural networks, even with few parameters. This means that we can simulate any level of difficulty, inserting enough noise.

Two types of classifications have been made: the odd / even digit division and the greater / less than 5 digit division.

In this way the black network is forced to recognize complex patterns of different nature.

Instead, as far as clothes are concerned, they are divided into 9 categories, to make the task binary simply 2 classes have been created, each consisting of 5 categories.

In summary, we have 2 databases for 4 noise levels, for a total of 8 classification tasks on different databases.

5.1.2 Models

As regards the models used, it was decided to use two very common types of neural networks. The first model is the dense neural network, which is the simplest and most generic model in this category. In this network all the neurons of a layer are connected with the following ones, and therefore has a greater computing capacity, at the expense of a greater number of parameters and therefore greater inefficiency.

The second model is the CNN convolutional neural network, already discussed in the 1.1.3 chapter. This model is designed for use in image processing. It is based on the identification of local patterns in images, and on the construction, layer by layer, of increasingly complex patterns, which are finally used to distinguish two categories. It has the strong advantage of optimizing the number of parameters thanks to the ability to "delocalize" the patterns, ie recognize the same pattern regardless of the position in the image.

In practical applications, every problem requires the development of a model specifically designed for the type of task and the type of database. In any case, the most common neural networks used are the two taken into consideration, often also used together to compose a larger model.

Neural networks are also characterized by the number of parameters and the structure of the layers, for this reason, to simulate more possibilities, each model was used 3 times, with 3 levels of complexity, proportional to the size and number of layers, and therefore of total parameters.

In total we have 2 networks with 3 levels of complexity, for a total of 6 models.

6 models that solve 8 tasks correspond to 48 experiments, for each of which the parameters of the new proposed metric were studied.

The analyzes must be performed on the models at the end of the training, but care must be taken to recognize overfitting. In the event of overfitting, in fact, some hypotheses underlying the validity of the new metric could be lacking, such as the probabilistic interpretation of the elements of the datasets. In fact, overfitting is a condition in which the model is able to capture patterns with a level of detail greater than the average distance between two neighboring elements of a dataset, therefore it ignores the uncertainty due to the fact that the training set is only a sample of the distribution from which it is extracted.

For this reason the values discussed in this chapter refer to the point of the training where the overfitting does not yet occur, measured as the minimum point of the loss function of the test set.

Finally, in the 1.1.2 section, the variation of the parameters discussed during the training of the models will also be analyzed, in order to better identify the

validity limits of the hypotheses underlying this new metric.

5.2 Calibration

This section relates to the verification of the hypothesis of perfect calibration. As demonstrated in the 4.2.1 chapter, perfect calibration is an optimal condition for an agent that, in its training process, tries to optimize a cross entropy type loss function.

On the other hand, it is not certain that, on a practical level, a neural network will be able to reach this optimal point. the causes can be linked to an inability of the model to converge to this optimal state, to problems intrinsic to the task or to overfitting phenomena, which invalidate the hypothesis of being able to consider the various elements of the database as aleratory variables.

From a practical point of view, one can measure the calibration error 4.24, which is the content of this section.

The calibration error, in the real case, must be calculated by dividing the interval [0,1] into a certain number of bins, the error will measure the uncertainty of the probability of belonging to a class in that bin.

It is therefore necessary to compare the calibration error with the size of the bin, and consider the calibration as maximum if the two dimensions correspond.

If the calibration error is much larger than the bin size, it means that the model is not well calibrated.

This parameter is useful, in the new metric, to ensure interpretability of the network output. In the event that the network is not well calibrated, it is possible to use it anyway, but with the awareness that its output cannot be considered as a frequentist probability.

Note that the calibration error may be due to the accuracy of the model. In particular, if the model is extremely accurate, the calibration error could increase, as suspected from the considerations made on the 4.6 graph. This phenomenon can be explained by the fact that, if the two classes are easily separable, the probabilities of belonging to the classes are around 100% and also the division into bins becomes less sensible.

For this reason, it was decided to use a scatter plot in which this dependence is highlighted:

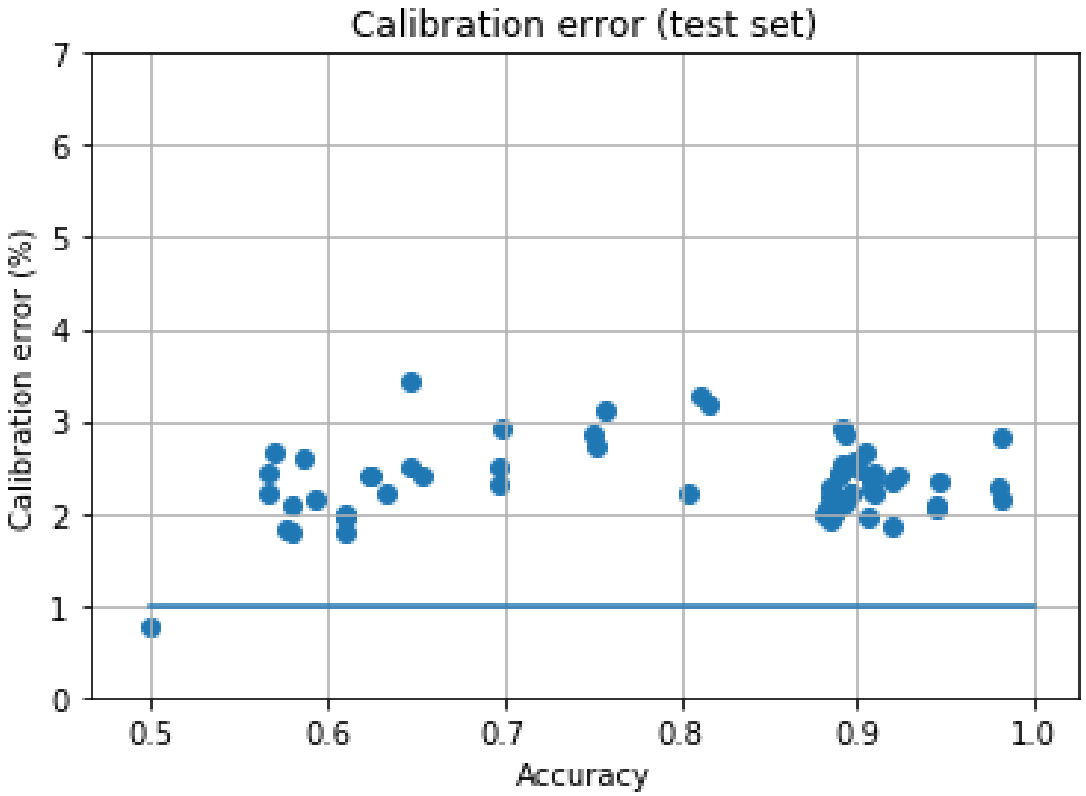

Figure 5.2: Calibration error VS Accuracy in all the experimental cases

We note that the average value of the ratio between calibration error and bin size is around 2.5%, and testifies that all networks reach a good level of calibration.

The perfect calibration level is only consistent with errors around the value of 1%, which corresponds to the size of the bins. This condition does not occur in the experiments, but it must be said that the tested neural networks were subjected to a standardized training with the same hyperparameters. If each network were subjected to specific training, a better local minimum of the parameters would be achieved, which would also allow to improve the calibration.

The most important observation is that there is no correlation with accuracy, this is a strong indication that the optimality of perfect calibration is a property guaranteed by the gradient descendent process and by the universal expressiveness of neural networks.

5.3 Beta distribution assumption

In this chapter, we analyze the hypothesis according to which a "well trained" neural network converts the distribution of the elements of a class into a distribution belonging to the family of Beta distributions as output.

This hypothesis is very useful because it allows to create a mapping with subjective logic, but its validity must be supported by empirical and experimental data.

5.3.1 Inference error measure

A first method to evaluate this hypothesis is to find the Beta distribution that best represents the output distributions and to calculate a measure of distance between the empirical distribution and that of the model.

This can be done through the inference algorithm which allows to find the theoretical distribution that minimizes the Kullback-Leibler divergence 2.20 with the empirical distribution, illustrated in the appendix **??**.

Please note that, as discussed in chap. 2.3, the Kullback-Leibler divergence gives an indication of the probability that the empirical distribution can be obtained from a sampling of the theoretical distribution, and therefore is an excellent metric for the inference error between the two distributions.

Also in this case, the results depend on the accuracy, similar to the calibration error, and it is therefore convenient to view them in the form of a scatter plot.

Figure 5.3: Kullback-Leibler Divergence VS Accuracy in all the experimental cases

It is noted that the value of the KL divergences increases a lot with accuracy. This is probably partly due to the shape of the distribution, which by its nature involves higher values of this parameter than an empiric.

In any case, it is an indication that the new metric has a limit of applicability proportional to the accuracy: for easy tasks in which the model is able to classify with extreme precision, the metric partially loses its meaning. This makes sense because the underlying hypothesis is to consider the network as a probability distribution, but if the task is easy the network does not need probabilities, but it is safe in its classification.

These values have the defect of not being easily interpretable, because they do not give useful indications on the validity of the hypothesis according to which the Beta distribution represents the data well. On a visual level, it is very useful to represent the two distributions as done in the figure 4.1, because it is possible to realize the similarity between the distributions.

The next section discusses the second method for evaluating the fitting, which is much more intuitive and therefore useful on a practical level.

5.3.2 Classical metric mapping error

The second method to evaluate the accuracy of the Beta distribution in representing data is obtained starting from the final objective of the discussion: to create a new metric, more in-depth than the classic ones.

A necessary but not sufficient condition to achieve this result is that the new metric is able to reproduce all the results of the classic metrics.

Classic metrics are parameters calculated on empirical output distributions. These parameters can be similarly calculated from the theoretical Beta distributions, as shown in Eq. 4.3.2.

Therefore, the relative errors of some classical metrics are taken into consideration: accuracy, secondary diagonal of the confusion matrix and ROC AUC. In the case of accuracy, we used the relative error of $1 - accuracy$, to point out the error of high accuracy.

Again, it pays to show data in relation to accuracy.

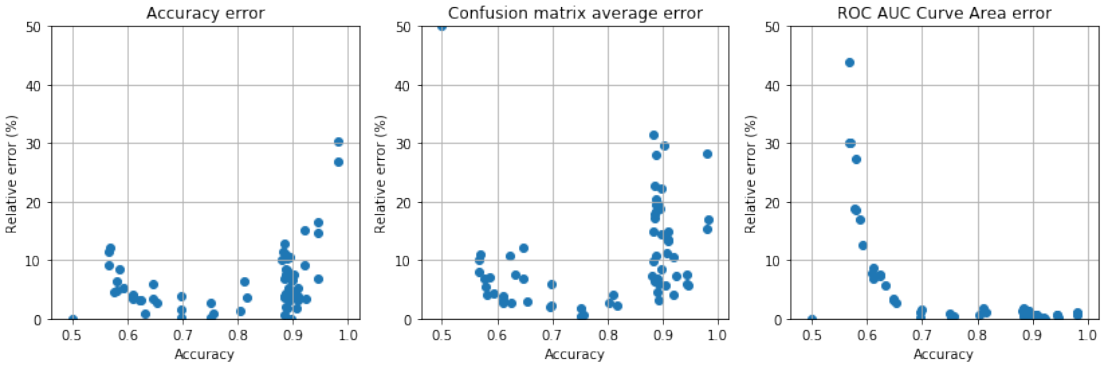

Figure 5.4: Classical metrics errors VS Accuracy in all the experimental cases

The first graph on the left represents the accuracy of $1 - accuracy$, so its correlation with accuracy is that small percentages of difference are amplified in the relative error calculation. In any case, it is noted that the error remains contained and is accentuated only for very high accuracy.

The last graph on the right, relating to the ROC AUC, shows that the Beta distribution is able to reproduce this classic metric very well.

The central graph shows the same phenomenon, only partly caused by accuracy. It is deduced that the Beta distribution could differ with the empirical distribution as regards the calculation of false positives and false negatives. This is only a problem if you don't look at the network output for false positives. Indeed, if the output is analyzed, it turns out that it is around 0.5 and therefore, even if the misclassification model of the elements, however, gives an indication of uncertainty in the classification. Put simply, an output of 0.47 is a false negative only based on

the 0.5 threshold, but if we interpret the output as a probability then we do not classify it as negative but as an uncertain case.

Based on the considerations made regarding the 4.6 graph and on the results shown in the 5.2 graph, it is concluded that the mapping with the Beta distribution begins to be inaccurate for accuracy around 90% . For very easy tasks, the assumptions of the hypothesis that the output of a neural network can be distributed according to a Beta distribution are lacking.

From this it is concluded that the new metric is valid in cases in which the model is not in any case able, due to lack of information, to solve the classification by itself, and therefore it makes much more sense to reason in a probabilistic way, using the tools of Logic Subjective.

5.4 Training and overfitting

Finally, it is interesting to observe the behavior of the neural network during training through the use of the new parameters discussed. In particular, the analysis is interesting as regards the overfitting behavior and the mapping with the opinions of subjective logic.

Below is an analysis of all the parameters referred to one of the cases tested.

First of all, we analyze the training process from the point of view of the loss function. In this way we can already observe the overfitting from the point of separation of the graph between train and test set.

We observe that overfitting begins to arrive around epoch 20.

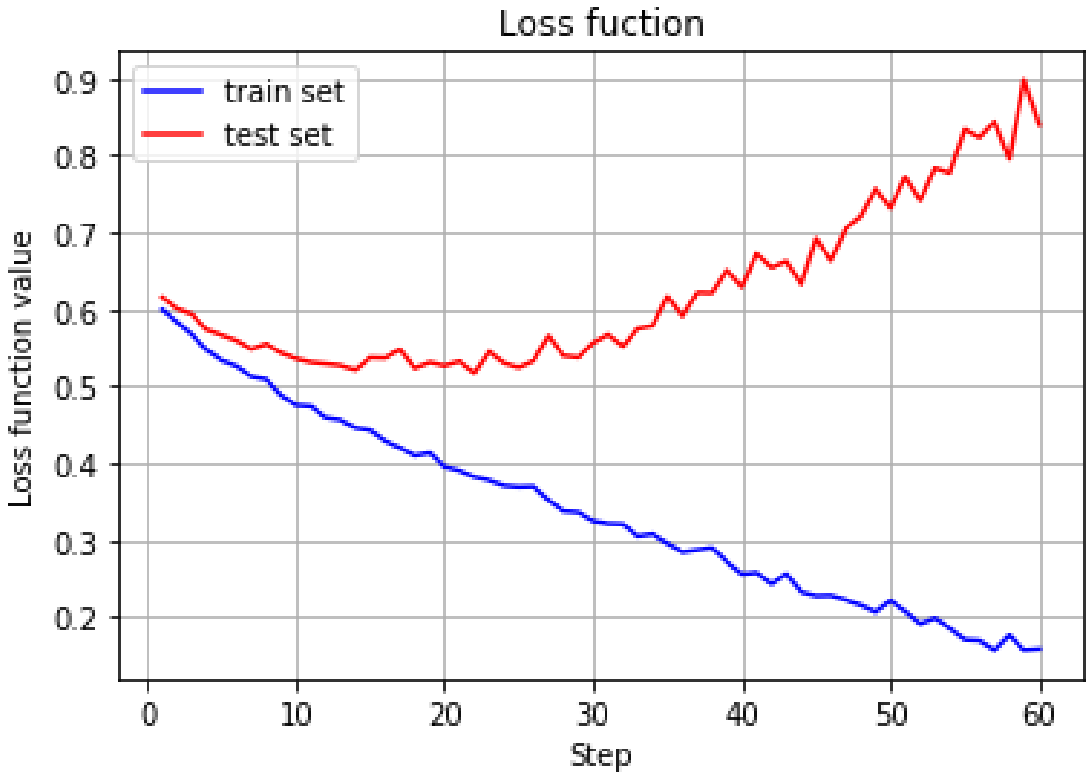

Figure 5.5: Train and test set loss function in each epoch

We proceed with a look at the classic metrics. Accuracy and ROC AUC were chosen.

It is observed that in the test set the ROC AUC begins to decrease in overfitting, while the accuracy remains the same. This means that the outputs change, but not significantly around the 0.5 value, which corresponds to the threshold on the basis of which accuracy is calculated.

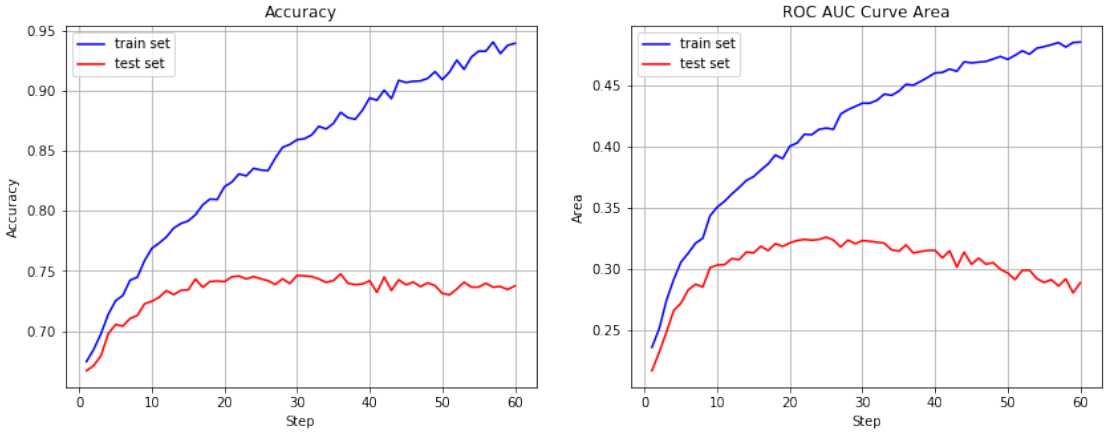

Figure 5.6: Accuracy and ROC AUC in each epoch

Now we analyze the errors related to the two assumptions underlying the new metric: calibration and fitting with the Beta distribution.

As for the calibration, there is a substantial increase in error with overfitting only in the test set, while the training set remains well calibrated. This means that overfitting does not undermine the optimality of the calibration per se, but, of course, today it offsets reliability as regards the test set.

As for the Beta fitting, we notice both an increase in the error with overfitting in both sets, but also a basic difference between the two classes even before overfitting.

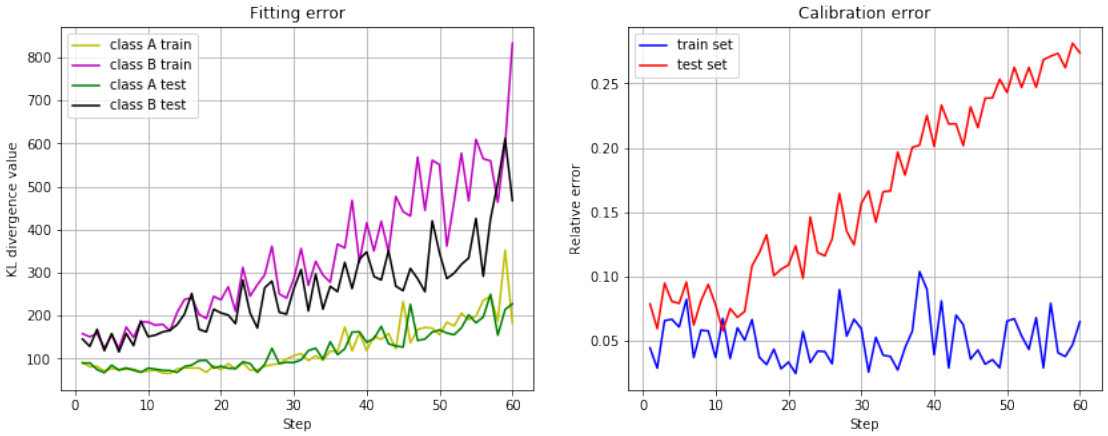

Figure 5.7: Calibration and Beta fitting error in each epoch

The fitting error can also be measured by the relative errors of the classic metrics calculated with the model.

From the graph we observe that overfitting leads to an increase in the error in

both sets, but much more accentuated in the test set.

It can be concluded that in overfitting the hypothesis of fitting with the Beta distribution is no longer valid, which instead remains valid in the absence of overfitting.

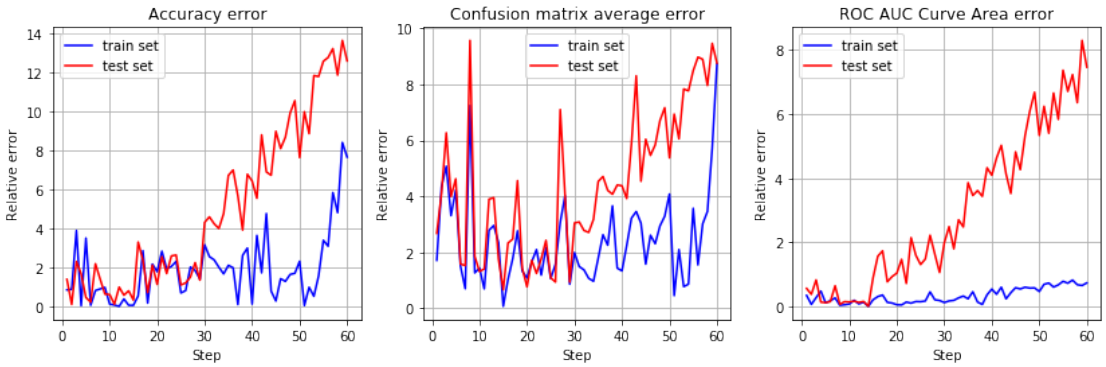

Figure 5.8: Classical metrics errors in each epoch

Finally, two graphs are shown that show the trajectory of the neural network in the two-dimensional space of the Beta distributions and in the "triangular" space of the hypotheses. The heatmap represents the accuracy, given by the formula 4.42, while the straight line represents the restriction in the case of perfect calibration, discussed in section 4.4. The 4 groups of points represent the Beta distributions relating to the 2 classes and 2 sets. The size of the points is inversely proportional to the fitting error. So the smaller points are less significant. The "motion" of the neural network begins with the largest points.

The first thing that is observed is the overfitting, which manifests itself with a clear separation between the two sets, simultaneously with a reduction of the points, corresponding to an increase in the fitting error.

Finally, it is observed that the points are around the line corresponding to the perfect calibration.

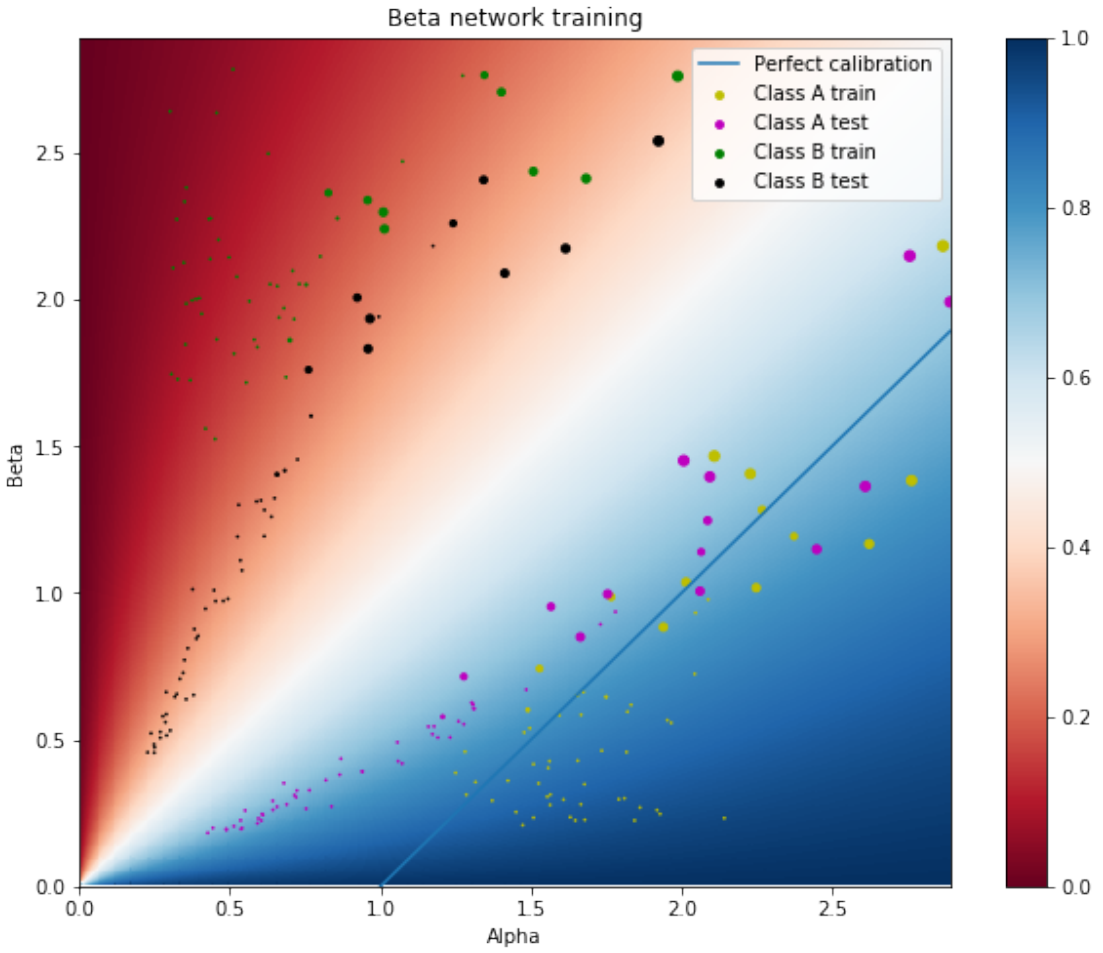

Figure 5.9: Training represented as trajectory in the Beta distributions space

In the "triangular" space of opinions, it is observed that, after overfitting, the train set tends towards absolute opinion, corresponding to the total certainty and accuracy of 100%, while the test set approaches the vacuous opinion , or the total uncertainty. This is consistent with the fact that, with a fairly large over fitted model, you can get to 100% accuracy for the training set and total inaccuracy for the test set.

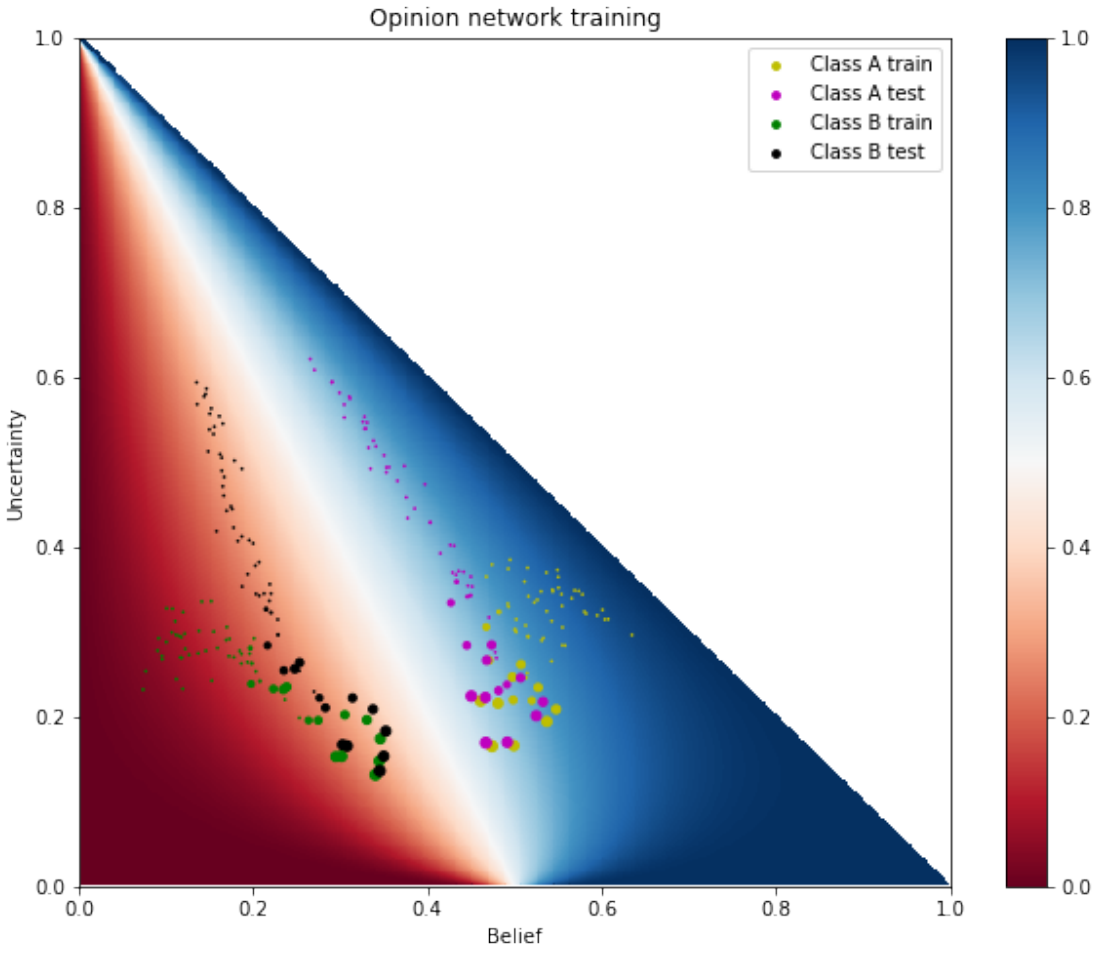

Figure 5.10: Training represented as trajectory in the Opinion space

Chapter 6

Application

In this chapter the conclusions and applicative aspects of the properties discussed during the thesis are discussed.

The previous chapters are very focused on the mathematical considerations underlying the new metric because it is necessary to have more or less rigid indications of the validity of the ideas discussed.

Once the theoretical bases have been discussed, it is possible to exploit the results obtained on a practical level. Taking medical diagnosis as the main purpose, it is of fundamental importance to extrapolate as much information as possible from the automated diagnosis model, because neural networks are "black boxes" and, on a social level, it is difficult to put people's health in hand. an algorithm whose output cannot be interpreted. Further information on the characteristics of the algorithm can be very useful both to the programmer, who may have further tools to correct the model, and to the doctor, who will be able to follow a more reliable decision-making process based on extra information extracted from the neural network.

6.1 Summary of the new metric

All the metrics of a model, including the one proposed) are made up of parameters obtainable through calculations applied to the empirical distributions of the outputs of the two classes.

The proposed metric, in particular, consists of 7 parameters: the calibration error, the 2 Beta fitting errors and the 4 parameters of the opinions of the subjective logic (obtained from the parameters of the Beta distributions).

It is possible to divide the metric in two according to the type of applications for which it can be used.

The 3 parameters relating to calibration and fitting errors are very useful for

the programmer to improve the performance of the model, while the 4 parameters relating to opinions are very useful for the doctor to implement a rational decision-making process on the outcome of the diagnoses.

6.1.1 For the developer

The programmer's purpose is to optimize the model's performance. The type of performance required strongly depends on the type of task, but is usually measured by classic metrics.

Two classic examples are the war drone algorithm, for which it is very important to minimize false positives, which would involve friendly fire, and medical diagnosis, in which it is important to minimize false negatives, which correspond to not detecting a disease. potentially serious of a patient.

The new metric allows programmers to also consider 2 other neural network performances that are related to the interpretability of the output.

This interpretability is represented by calibration and mapping with subjective logic: the calibration allows to interpret the output as "true" probability, while the mapping allows to interpret the neural network as an opinion, and to exploit all the decision making techniques already developed in this field of mathematics and statistics.

Neural networks, from the programmer's point of view, can be classified on the basis of the errors of these two characteristics. In particular, by setting a threshold for these errors, we can identify 4 types of models, represented in the 4 quadrants in fig 6.1.

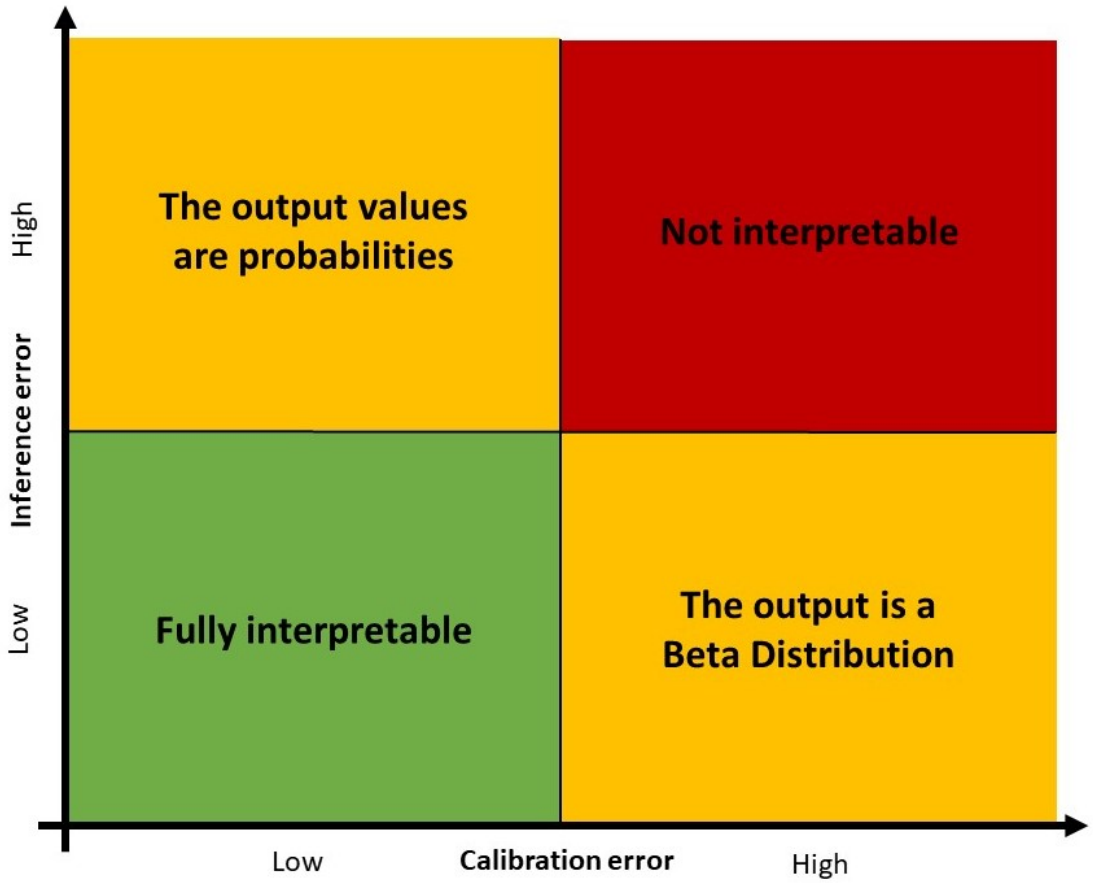

Figure 6.1: Qualitative division of interpretability of a trained model.

The programmer's goal, of course, is to direct the network towards the lower left quadrant, which minimizes errors and allows to state that the output is interpretable as probability and to respect the criteria underlying certain protocols of decision making.

The results reported in the 5 chapter show that the networks converge spontaneously to that quadrant but, even in more complex and difficult cases where this does not happen, there are techniques to correct the calibration of the model [31].

If the programmer fails to fit his model into the "interpretable" quadrant, then he can declare that the model cannot be interpreted using the new metric. Without the new metric, in fact, the programmer has no information about which quadrant his model fits. So high levels of error are not a problem per se, they simply result in less interpretability of the output.

This metric for the programmer can only be obtained with the 3 parameters of fitting and calibration errors and does not need the 4 parameters referring to Beta distributions and opinions.

6.1.2 For the doctor

Once an algorithm is classified as interpretable, and is used in the medical field, the 4 parameters of opinions come into play.

The parameters contain information on how likely a patient is to be classified as sick. This information has two degrees of freedom, which can be interpreted as belief and uncertainty, discussed in the 3 chapter, or as confidence and likelihood [39].

Likelihood refers to how high the output of the algorithm will be on average in the presence of a patient. Confidence, on the other hand, refers to how likely it is that two clients with the same disease will be classified with different outputs.

In this case, however, it doesn't make much sense to divide the space into just 4 quadrants, but it becomes much more useful to segment the classification more. An example of practical levels is described is Sherman Kent's *Words os Extimated Probability* [40]; based on the *Admirality Scale* as used within the UK National Intelligence Model [41].

Likelihood levels:		Absolutely not	Very unlikely	Unlikely	Somewhat unlikely	Chances about even	Somewhat likely	Likely	Very likely	Absolutely
Confidence levels:		9	8	7	6	5	4	3	2	1
No confidence	E	9E	8E	7E	6E	5E	4E	3E	2E	1E
Low confidence	D	9D	8D	7D	6D	5D	4D	3D	2D	1D
Some confidence	C	9C	8C	7C	6C	5C	4C	3C	2C	1C
High confidence	B	9B	8B	7B	6B	5B	4B	3B	2B	1B
Total confidence	A	9A	8A	7A	6A	5A	4A	3A	2A	1A

Figure 6.2: Qualitative levels of Likelihood and Confidence.

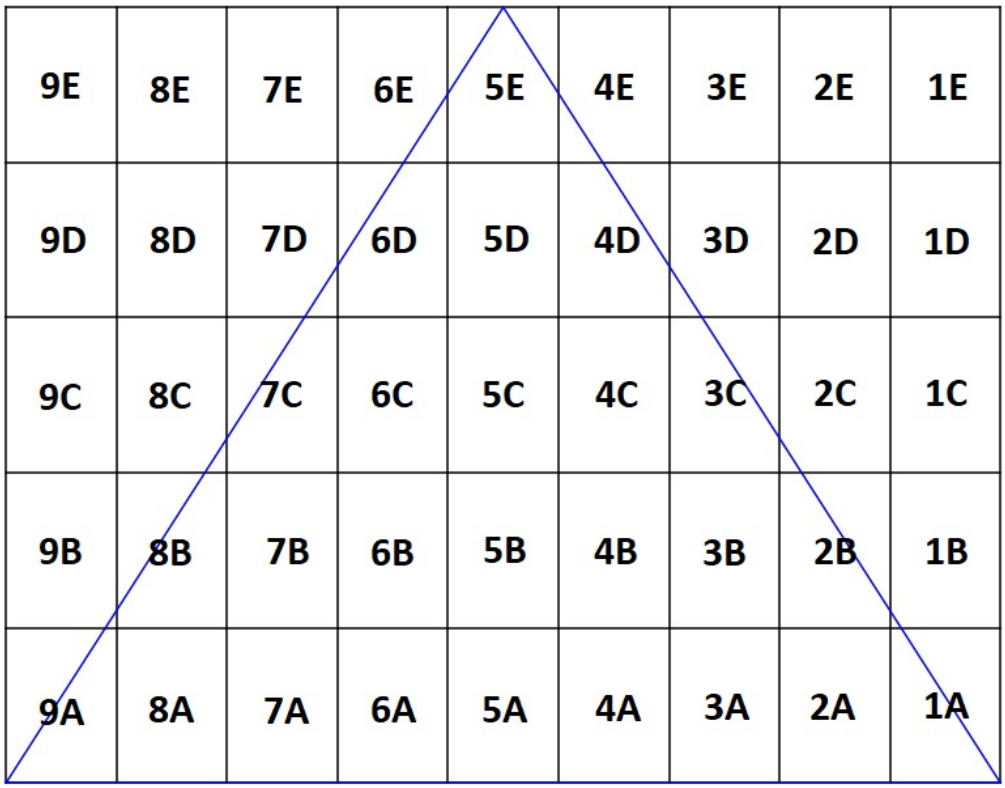

Figure 6.3: Qualitative levels division of opinion space.

From a human point of view it is useful to identify these qualitative levels to evaluate the performance of the algorithm, but the new metric has even more quantitative applications, linked to the decision making protocol to be implemented after having carried out the medical examination and having had the result. from the output of the [3, p. 51] machine.

Of course, this division into qualitative levels is only one of many possible, and is not intended for the specific case of machine learning models for binary classification, so it is possible to develop a classification more suited to specific [42] cases.

Through the use of Bayesian statistics and the techniques discussed in the book of Subjective Logic it is possible to efficiently use the additional information extracted from the model to organize medical services to a large number of patients whose probabilistic truths about the presence and severity are known. of a certain pathology [3, p. 133].

These applications are not the object of study of this thesis, if not only the

example discussed in the following section.

6.2 Conclusions

Machine learning is a technology that is very promising in the medical field but is still in its infancy.

The use of artificial intelligences will lead to a paradigm shift in the world of medical diagnosis and therefore, as technology advances very quickly, regulations and standardization are slow in coming.

Currently all over the world the responsibility for errors lies with the doctor, which guarantees a disincentive to make mistakes and protection towards the client. Even if it is shown that the performance of artificial intelligences exceeds that of humans, the problem of responsibility for errors that is at the basis of the system of incentives and protections remains.

For this reason, more and more transparency [43] will be required in the future and more and more regulation on these technologies will be implemented.

This thesis is an attempt to extrapolate as much humanly interpretable information as possible from a classical neural network.

The proposed new metric does not intend to be an alternative to classical metrics because it does not aim to evaluate the classical accuracy performances, but tries to describe other properties of the algorithm concerning the interpretability and reliability of the model output.

The way of interpreting the model as probability distribution and mapping with subjective logic is just one of the many approaches that can be taken to extrapolate more [44] [45] information. Another approach, for example, is to investigate the output of neurons located in the internal layers of a CNN to understand the patterns recognized by the network that led to a certain final output.

The method proposed in this thesis shows that, at least in a certain range of cases, it is possible to extrapolate more information on interpretability only from the model output, to give the programmer the tools to make the model more interpretable and the doctor the tools to frame the diagnosis of an algorithm in a larger context of decision making that can be implemented more automatically.

List of Figures

Bibliography

[1] Yogesh Kumar; Apeksha Koul; Ruchi Singla; and Muhammad Fazal Ijaz. «Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda.» In: (2022) (cit. on p. i).

[2] *Artificial Intelligence and Diagnostic Errors*. `https://psnet.ahrq.gov/perspective/artificial-intelligence-and-diagnostic-errors`. 2020 (cit. on p. i).

[3] Audun Josang. *Subjective Logic*. Olso, Norway: Springer, 2016 (cit. on pp. i, 20, 22, 23, 39, 62).

[4] Christopher Bishop. *Pattern Recognition and Machine learning*. Springer, 2006 (cit. on p. 1).

[5] F. Sebastiani. «Machine learning in automated text categorization.» In: (2002) (cit. on p. 1).

[6] Nelson B. Barreno M.; Chi F. J. ;Joseph A. D. ;Rubinstein B. I. P. ;Saini U. ;Sutton C.; Tygar J. D. ; Xia K. «Exploiting machine learning to subvert your spam filter.» In: (2008) (cit. on p. 1).

[7] T. Rosten E. ; Drummond. «Machine learning for high-speed corner detection.» In: (2006) (cit. on p. 1).

[8] A. McCallum ; K. Nigam ; J. Rennie ; K. Seymore. «A machine learning approach to building domain-specific search engines.» In: (1999) (cit. on p. 1).

[9] Maithra Raghu; Ben Poole; Jon Kleinberg; Surya Ganguli; Jascha Sohl Dickstein. «On the Expressive Power of Deep Neural Networks». In: (2017) (cit. on p. 2).

[10] Kurt Hornik; Maxwell Stinchcombe; Halbert White. «Multilayer feedforward networks are universal approximators.» In: (1989) (cit. on p. 2).

[11] Jason Brownlee. «A Gentle Introduction to the Rectified Linear Unit (ReLU)». In: (2019) (cit. on p. 3).

[12] Massimo Buscema. «Back Propagation Neural Networks». In: (1996) (cit. on p. 4).

[13] T. N. Hubel D. H.; Wiesel. «The Problem of Overfitting». In: (2004) (cit. on p. 4).

[14] Kunihiko Fukushima. «Receptive fields and functional architecture of monkey striate cortex». In: (1968) (cit. on p. 6).

[15] Kunihiko Fukushima. «Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position». In: (1980) (cit. on p. 6).

[16] Lacra Gao Bolin; Pavel. «On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning». In: (2017) (cit. on p. 7).

[17] George Cybenko; Dianne P. O'Leary; Jorma Rissanen. *The Mathematics of Information Coding, Extraction and Distribution*. Springer, 1999 (cit. on pp. 8, 17).

[18] Powers David M W. «Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness Correlation». In: (2011) (cit. on p. 9).

[19] Jeffrey S. Evans Michael; Rosenthal. *Probability and statistics: the science of uncertainty*. University of Toronto, 2010 (cit. on p. 12).

[20] Ramon de Elia; Renè Laprise. «Diversity in Interpretations of Probability: Implications for Weather Forecasting». In: (2005) (cit. on p. 12).

[21] Carl-Axel S.; Staël Von Holstein. *The Concept of Probability in Psychological Experiments*. Springer, 2012, pp. 15–23 (cit. on p. 13).

[22] Jacques-Paul Dubucs. *Philosophy of Probability*. Springer, 1993 (cit. on p. 13).

[23] Arjun K. Gupta; Saralees Nadarajah. *Handbook of Beta Distribution and Its Applications*. CRC Press, 2020 (cit. on p. 14).

[24] D.B. Karp. «Normalized incomplete beta function: log-concavity in parameters and other properties». In: (2016) (cit. on pp. 15, 33, 37).

[25] C.E. Shannon. «A mathematical theory of communication». In: (1948) (cit. on p. 17).

[26] S. Kullback; R. A. Leibler. *On Information and Sufficiency*. The Annals of Mathematical Statistics, 1951 (cit. on p. 17).

[27] Daniel Ellsberg. «Risk, Ambiguity, and the Savage Axioms». In: (1961) (cit. on p. 21).

[28] George Boole. *The Mathematical Analysis of Logic*. 1847 (cit. on p. 21).

[29] Nils J. Nilsson. *Probabilistic Logic*. 1986 (cit. on p. 21).

[30] August Ferdinand Mobius. *Der barycentrische Calcul*. 1827 (cit. on p. 22).

[31] Chuan Guo; Geoff Pleiss; Yu Sun; Kilian Q. Weinberger. «On Calibration of Modern Neural Networks». In: (2017) (cit. on pp. 28, 31, 60).

[32] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991 (cit. on p. 29).

[33] Leigh J. Halliwell. «The Log-Gamma Distribution and Non-Normal Error». In: (2021) (cit. on p. 35).

[34] Ingram Olkin; Herman Rubin. «Multivariate Beta Distributions and Independence Properties of the Wishart Distribution». In: (1964) (cit. on p. 39).

[35] *MNIST Digits*. `https://web.archive.org/web/20180211153301/http://www.gavo.t.u-tokyo.ac.jp/~qiao/database.html` (cit. on p. 44).

[36] *MNIST Fashion*. `https://www.kaggle.com/datasets/zalando-research/fashionmnist` (cit. on p. 44).

[37] Shima Yoshihiro ; Nakashima Yumi ; Yasuda Michio. «Handwritten Digits Recognition by Using CNN Alex-Net Pre-trained for Large-scale Object Image Dataset». In: (2018) (cit. on p. 45).

[38] Sanghyeon An; Minjun Lee; Sanglee Park; Heerin Yang; Jungmin So. «An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition». In: (2018) (cit. on p. 45).

[39] Andrew D. Brasfield. «Forecasting accuracy and cognitive bias in the analysis of competing hypotheses». MA thesis. Erie, Pennesylvania: Mercyhurst College, 2009 (cit. on p. 61).

[40] Sherman Kent. *Words of Estimative Probability*. CIA, Center for the study of Intelligence, 2012 (cit. on p. 61).

[41] *Guidance on the National Intelligence Model*. `https://whereismydata.files.wordpress.com/2009/01/national-intelligence-model-20051.pdf`. 2005 (cit. on p. 61).

[42] Ries Sebastian; Habib Sheikh; Mühlhäuser Max; Varadharajan Vijay. *CertainLogic: A Logic for Modeling Trust and Uncertainty*. 2011 (cit. on p. 62).

[43] U. Hoffrageand;S. Lindsey; R.Hertwig; G. Gigerenzer. «Communicating Statistical Information». In: (2001) (cit. on p. 63).

[44] Diogo Carvalho; Eduardo Pereira; Jaime Cardoso. «Machine Learning Interpretability: A Survey on Methods and Metrics». In: (2019) (cit. on p. 63).

[45] Jianlong Zhou; Amir H. Gandomi; Fang Chen; Andreas Holzinger. «Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics». In: (2021) (cit. on p. 63).