mathematics

MDPI

Article

ROC Curves, Loss Functions, and Distorted Probabilities in Binary Classification

Phuong Bich Le [1,*,†] and Zung Tien Nguyen [2,3,†]

[1] Department of Mathematics, Hanoi University of Mining and Geology, No. 18 Vien Street, Duc Thang Ward, Bac Tu Liem District, Hanoi City 100000, Vietnam
[2] Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France; ntzung@torus-actions.fr or tienzung@math.univ-toulouse.fr
[3] Torus Actions SAS, 31062 Toulouse, France
* Correspondence: lbphuong@sputnik.vn or lebichphuong@humg.edu.vn; Tel.: +84-988782112
† These authors contributed equally to this work.

Abstract: The main purpose of this work is to study how loss functions in machine learning influence the "binary machines", i.e., probabilistic AI models for predicting binary classification problems. In particular, we show the following results: (i) Different measures of accuracy such as area under the curve (AUC) of the ROC curve, the maximal balanced accuracy, and the maximally weighted accuracy are topologically equivalent, with natural inequalities relating them; (ii) the so-called real probability machines with respect to given information spaces are the optimal machines, i.e., they have the highest precision among all possible machines, and moreover, their ROC curves are automatically convex; (iii) the cross-entropy and the square loss are the most natural loss functions in the sense that the real probability machine is their minimizer; (iv) an arbitrary strictly convex loss function will also have as its minimizer an optimal machine, which is related to the real probability machine by just a reparametrization of sigmoid values; however, if the loss function is not convex, then its minimizer is not an optimal machine, and strange phenomena may happen.

Keywords: optimization; binary classification; machine learning; ROC curve; accuracy metrics; loss function; quadratic loss; quartic loss; cross-entropy; convexity; information space; optimal machine; real probability machine; distorted probabilities

MSC: 68T20; 68V99

Citation: Le, P.B.; Nguyen, Z.T. ROC Curves, Loss Functions, and Distorted Probabilities in Binary Classification. *Mathematics* **2022**, *10*, 1410. https://doi.org/10.3390/math10091410

Academic Editors: María Purificación Galindo Villardón and Xuan Zhao

Received: 31 January 2022
Accepted: 18 April 2022
Published: 22 April 2022

1. Introduction

The aim of this paper is to study loss functions that are used in machine learning for training binary classification machines. We refer to [1–5] for an introduction to machine learning. Loss functions play an extremely important role in differential machine learning, and many different loss functions have been created for each problem, based on some general principles (see, e.g., [4,6–9]). However, most people use some available loss functions without paying much attention to the question of "how the choice of a loss function will affect the outcome of a machine learning problem". Recently, researchers started paying more attention to the properties of the loss functions, which would help the stochastic gradient flows in differential machine learning converge to the desired values of the parameters; see, e.g., [10–19]. Our present work is also a contribution in this direction.

The question that we want to address in this paper is how to design the loss functions so that the machines that are loss minimizers are also optimal in terms of prediction accuracy. In order to study this question, we first need to study the ways to measure and compare the accuracy of different binary machines. This led us to the notions of information spaces, real probability machines, optimal machines, the convexity of the ROC curve, and natural inequalities relating different metrics of accuracy. We then discovered

that the convexity condition is the main condition on the loss function in order to ensure that its minimizer will be an optimal machine in terms of precision. In general, those loss-minimizing machines will give *distorted probabilities* instead of real probabilities. (When people interpret sigmoid values as probabilities in the artificial intelligence literature, this is not accurate in general, as those numbers are *not* real probabilities.) However, this distortion may be viewed positively, as a feature instead of a bug, and one can go back from distorted probabilities to real probabilities by performing large-scale testing if one wishes.

The main results presented in this paper are the following:

1. (Section 2, Proposition 1) Different measures of accuracy such as the area under the curve (AUC) of the ROC curve, the maximal balanced accuracy, and the maximally weighted accuracy are "topologically equivalent" in the sense that if one of them is high (i.e., close to 1), then the other ones are also automatically high, with natural inequalities relating them.
2. (Section 3, Proposition 2) The so-called real probability machines with respect to given information spaces are the optimal machines, i.e., they have the highest precision among all possible machines, and moreover, their ROC curves are automatically convex.
3. (Section 4, Proposition 3) The cross-entropy and the square loss are the most natural loss functions in the sense that the real probability machine is their minimizer.
4. (Sections 5 and 6, Proposition 4) An arbitrary smooth strictly convex loss function will also have as its minimizer an optimal machine, which is related to the real probability machine by just a reparametrization of the sigmoid values. However, if the loss function is not convex, then its minimizer is not an optimal machine, and strange phenomena may happen.

Propositions 1 and 2 have been announced by us in a recent talk at AICI 2022 [20].

2. Binary Machines, ROC Curves, and Accuracy Metrics

Let us fix some notations for this paper. Denote by Ω an input space together with some probability measure P_Ω, and

$$Y : \Omega \to \{0, 1\} \tag{1}$$

a binary classification problem on Ω. For example, Ω is the population, and Y is COVID-positive (1) or COVID-negative (0). Y is often called the **ground truth**.

We want to build a *binary machine*:

$$M : \Omega \to [0, 1] \tag{2}$$

(a test, whose values are in the interval [0,1]) that predicts the value of Y. Given a threshold $\sigma \in]0, 1[$, for each element $x \in \Omega$, we put

$$Y_\sigma(x) = 1 \text{ if } M(x) \geq \sigma \text{ and } Y_\sigma(x) = 0 \text{ if } M(x) < \sigma \tag{3}$$

The performance (i.e., precision) of the predictor Y_σ with respect to the ground truth Y can be measured by two basic performance indicators, called the **sensitivity** (=*true positive* rate) $TP(\sigma)$ and **specificity** (=*true negative* rate) $TN(\sigma)$, defined by the following formulas:

$$TP(\sigma) = P(Y_\sigma = 1 | Y = 1) = \frac{P_\Omega(M(x) \geq \sigma \,,\, Y(x) = 1)}{P_\Omega(Y(x) = 1)}, \tag{4}$$

$$TN(\sigma) = P(Y_\sigma = 0 | Y = 0) = \frac{P_\Omega(M(x) < \sigma \,,\, Y(x) = 0)}{P_\Omega(Y(x) = 0)}. \tag{5}$$

The curve $ROC : [0, 1] \to [0, 1] \times [0, 1]$ given by the formula

$$ROC(\sigma) = (1 - TN(\sigma), TP(\sigma)) \tag{6}$$

is called the receiver operating characteristic (ROC) curve of the machine M in the literature and is very widely used in many fields; see, e.g., [21–26] and the references therein. The number $FP(\sigma) = 1 - TN(\sigma)$ is called the *false positive* rate at the threshold σ.

The ROC curves goes "backward" from the point $ROC(0) = (1,1)$ to the point $ROC(1) = (0,0)$ in the unit square, and the higher the curve, the more accurate the machine is. The so-called AUC is the area of the region under the ROC curve in the unit square and is a popular measure for the accuracy of the machine. See Figure 1 for an illustration.

Another popular measure of accuracy is the **maximally weighted accuracy**, denoted here by MWA (see, e.g., [23]): given a weight $w \in [0,1]$ (determined by the ratio between the cost of a false negative and the cost of a false positive), we put

$$WA(\sigma) = w \cdot TP(\sigma) + (1-w) \cdot TN(\sigma) = w \cdot TP(\sigma) - (1-w) \cdot FP(\sigma) + (1-w), \quad (7)$$

$$MWA = \max_{\sigma \in [0,1]} WA(\sigma). \quad (8)$$

In particular, when $w = 0.5$, then one obtains the so-called **maximal balanced accuracy**:

$$MBA = \max_{\sigma \in [0,1]} BA(\sigma), \text{ where } BA(\sigma) = \frac{TP(\sigma) + TN(\sigma)}{2} \quad (9)$$

We remark that $0 \leq AUC, MWA, MBA \leq 1$ for any machine, and if any of these numbers is equal to 1, then it means that the machine is perfect, 100% accurate. To borrow a notion from topology and functional analysis, we can say that the AUC, MWA, and MBA are different metrics of accuracy, but they are *topologically equivalent*, in the sense that if one of these numbers is close to 1, then the other two numbers must also be automatically close to 1, i.e., if the machine is highly precise with respect to one of these metrics, then it is also highly precise with respect to the other metrics. More precisely, we have the following simple inequalities relating these metrics of accuracy:

Proposition 1. *With the above notations:*
(i) For any binary machine M, we have

$$1 - 2(1 - MBA)^2 \geq AUC \geq 2MBA - 1. \quad (10)$$

If, moreover, the ROC curve of the machine M is convex, then we have

$$AUC \geq MBA. \quad (11)$$

(ii) For any given weight $w \in]0,1[$ and any given binary machine M, we have

$$1 - \frac{(1 - MWA)^2}{2w(1-w)} \geq AUC. \quad (12)$$

If, moreover, the ROC curve of the machine M is convex, then we have

$$AUC \geq 1 - \frac{(1 - MWA)}{2\min(w, 1-w)}. \quad (13)$$

Proof. (See Figure 1). W remark that a number $\sigma \in [0,1]$ is a threshold where the machine M attains the highest weighted accuracy if and only if the straight line through the point $ROC(\sigma)$, consisting of the points $(FP(\sigma) + wt, TP(\sigma) + (1-w)t)$, $t \in \mathbb{R}$, lies above the ROC curve. Indeed, the lines $\{(FP(\sigma) + wt, TP(\sigma) + (1-w)t),\ t \in \mathbb{R}\}$ of slope $(w, 1-w)$ are simply "lines of constant weighted accuracy". If the point $B = (FP(\sigma), TP(\sigma))$ of the ROC curve gives the maximally weighted accuracy, then no point of the ROC curve can lie above its corresponding line of slope $(w, 1-w)$, because lying above means higher weighted accuracy.

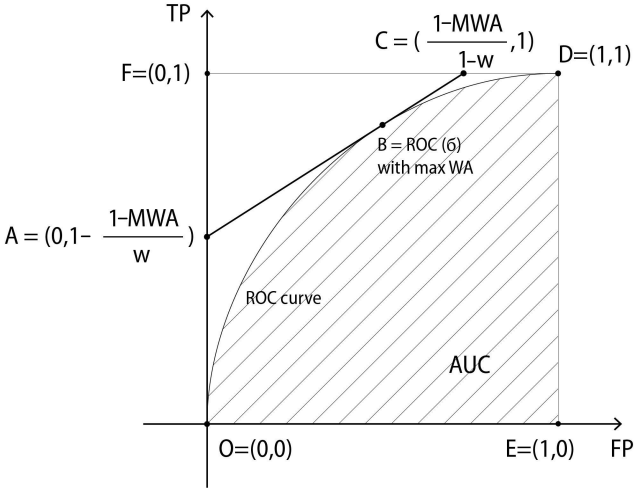

Figure 1. The ROC curve and the tangent line at a maximally weighted average point.

The line $\ell = \{(FP(\sigma) + wt, TP(\sigma) + (1-w)t),\ t \in \mathbb{R}\}$, where σ gives the maximally weighted accuracy for the machine M, cuts the boundary of the unit square at two points $A = (0, 1 - \frac{1-MWA}{w})$ and $C = (\frac{1-MWA}{1-w}, 1)$. The triangle $\triangle ACF$, where $F = (0,1)$, is disjoint from the region under the ROC curve, which implies that $AUC + \text{area}(\triangle ACF) \leq 1$. Since $\text{area}(\triangle ACF) = \frac{FA \cdot FC}{2} = \frac{(1-MWA)^2}{2w(1-w)}$, we obtain the inequality

$$AUC \leq 1 - \frac{(1-MWA)^2}{2w(1-w)}.$$

On the other hand, the region under the ROC curve contains the rectangle whose vertices are $(FP(\sigma), 0), (FP(\sigma), TP(\sigma), (1, TP(\sigma)), (1,0)$. The surface area of this rectangle is $TP(\sigma) \cdot (1 - FP(\sigma)) = TP(\sigma) \cdot TN(\sigma) = TP(\sigma) + TN(\sigma) - 1 + (1 - TP(\sigma))(1 - TP(\sigma)) \geq TP(\sigma) + TN(\sigma) - 1 = 2BA(\sigma) - 1$ (for every σ). Hence, we obtain the inequality

$$AUC \geq 2MBA - 1.$$

If the ROC curve is convex, then the region below it contains the quadrilateral $OBDE$, where $O = (0,0)$, $B = (FP(\sigma), TP(\sigma)), D = (1,1), E = (1,0)$ (for any σ). The surface area of this quadrilateral is exactly equal to $BA(\sigma)$, i.e., to MBA; hence, we obtain the inequality $AUC \geq BA(\sigma)$ for any σ, i.e., we have $AUC \geq MBA$.

Finally, the inequality $AUC \geq 1 - \frac{(1-MWA)}{2\min(w, 1-w)}$ in the case when the ROC curve is convex and the weight w is arbitrary is a direct consequence of the inequalities

$$AUC \geq area(OBDE) \geq \min(area(OADE), area(OCDE))$$

and the equalities $area(OCDE) = 1 - \frac{(1-MWA)}{2(1-w)}$, $area(OADE) = 1 - \frac{(1-MWA)}{2w}$. □

Remark 1. *For some inequalities in Proposition 1, we assume the ROC curve to be convex. The (near-)convexity of ROC curves has been observed empirically in many monographs and articles for long time; see e.g., [21–26] and the references therein; it has been shown in [22] (Theorem 3) that the convexity of the ROC curve is equivalent to some other natural reasonable "rationality" conditions on the machine (namely, the higher the "sigmoid value" is, the higher the probability of the event being true is).*

In Section 3, we show that the so-called real probability machine, which is in a sense the most natural machine, is also the most precise machine, and its ROC is automatically convex. Our result

helps explain why most ROC curves met in practice are nearly convex (because the machines must be nearly optimal in some sense if the machine learning processes for creating them are efficient).

Remark 2. *If we reparametrize a sigmoid function Σ by composing it with an arbitrary increasing bijection $f : [0,1] \to [0,1]$, then we obtain a new sigmoid function $\Sigma' = f \circ \Sigma$ whose ROC curve is the same as the ROC curve of Σ, up to a reparametrization by f. Namely, $ROC_\Sigma(\sigma) = ROC_{\Sigma'}(f(\sigma))$ for all $\sigma \in [0,1]$. In particular, a reparametrization allows us to change the sigmoid values without changing the performance metrics AUC, MBA, and MWA of a system.*

Remark 3. *Some authors (see., e.g., [27]) also use the geometric mean $GM = \sqrt{TP.TN}$ of sensitivity (TP) and specificity (TN) as a measure of accuracy for binary prediction problems. The obvious arithmetical inequalities $(a+b)^2/4 \geq ab \geq (a+b)^2/2 - 1$ (for any positive numbers $a, b \leq 1$) relate (in two ways) the geometric mean accuracy with the balanced accuracy $BA = (TP + TN)/2$. In particular, it means that the (maximal) geometric mean accuracy is also as good a measure of accuracy as MBA and the AUC, in the sense that they are topologically equivalent.*

Remark 4. *Given an original probability distribution $P = P_\Omega$ on the data space Ω, which is imbalanced in the sense that $P(Y = 0) \neq P(Y = 1)$ (imbalance between negative and positive cases), we may change it to a new, balanced, probability distribution $\hat{P}$, defined by the following formula:*

$$\hat{P}(A) = \frac{1}{2}\left[\frac{P(A \cap \{Y = 0\})}{P(Y = 0)} + \frac{P(A \cap \{Y = 1\})}{P(Y = 1)}\right].$$

It is easy to verify that the parametrized ROC curve (for a given machine $M = \Sigma \circ \phi : \Omega \to [0,1]$) with respect to $\hat{P}$ coincides exactly with the ROC curve with respect to P. Indeed, for any $\sigma \in [0,1]$, the true positive level at σ is $TP(\sigma) = \dfrac{P(\Sigma > \sigma, Y = 1)}{P(Y = 1)} = 2\hat{P}(\Sigma > \sigma, Y = 1) = \dfrac{\hat{P}(\Sigma > \sigma, Y = 1)}{\hat{P}(Y = 1)} = \widehat{TP}(\sigma)$, and similarly for $TN(\sigma)$. Thus, in the study of accuracy, without loss of generality, one may suppose that the probability distribution is balanced with respect to Y in the sense that $P(Y = 0) = P(Y = 1) = 0.5$.

3. Information Projection, Sigmoid Functions, and Optimal Machines

Conceptually, we can describe a binary machine M as a composition of two steps:

$$M = \Sigma \circ \phi, \tag{14}$$

where

$$\phi : \Omega \to \Phi \tag{15}$$

may be called the **information projection map** from the original data space Ω to a certain "distilled features space" or **information space** Φ, and

$$\Sigma : \Phi \to [0,1] \tag{16}$$

is a function from the information space Φ to the interval $[0,1]$, which we will call a (generalized) **sigmoid function**, in analogy with the classical sigmoid function $sigmoid(z) = \exp(z)/(\exp(z) + \exp(-z))$ often used in the last layer of a neural network in deep learning, even though our Σ is a function of many variables in general.

The idea is that, in most cases, we cannot know *everything* about an element $x \in \Omega$; we can know only *some information* about x, and that information is given by the projection map ϕ. Based on the available information $\phi(x)$ about x, we have to decide, via the value $M(x) = \Sigma(\phi(x))$, whether x is "negative" or "positive". Even when we know everything about x (e.g., when x is a digital image), the information contained in x may be too big

(millions of bits for an image), so we have first to "distill" that information into something smaller, which we call $\phi(x)$ and which we can control more easily.

In this paper, we assume that the information space Φ and the projection map $\phi : \Omega \to \Phi$ are fixed, and what we want to choose is just the sigmoid function $\Sigma : \Phi \to [0,1]$. The probability measure P_Φ on Φ is the push-forward of the probability measure P_Ω on Ω via the projection map ϕ.

In the artificial intelligence (AI) literature, the number $M(x) = \Sigma(\phi(x))$ is often called the *probability* (of x being positive given the information $\phi(x)$), even though it is *not true* in general, because Σ can be chosen rather arbitrarily, and as we will see in the following sections of this paper, even optimal machines obtained by machine learning methods usually give what we call *distorted probabilities* instead of *real probabilities*.

Nevertheless, among all the possible machines M (all the possible sigmoid functions $\Sigma : \Phi \to [0,1]$), there is one that is more natural than the others, which we call the **real probability machine** (the **probability sigmoid function**). The probability sigmoid function is simply the following conditional probability function:

$$\Sigma_{proba}(\varphi) = P(Y(x) = 1 | \phi(x) = \varphi) \tag{17}$$

for each $\varphi \in \Phi$.

We remark that, if we change the sigmoid function Σ by composing it with another function, $\Sigma' = \theta \circ \Sigma$, where $\theta : [0,1] \to [0,1]$ is a strictly increasing bijective function, then Σ and Σ' give the same ROC curve up to a reparametrization by θ. In other words, we can change a sigmoid value to any other value by composing it with a function, without changing the accuracy of the system, and this is one more reason why the sigmoid values should not be called "probabilities" in general.

Proposition 2. *With the above notations:*

(i) If $\Sigma : \Phi \to [0,1]$ is an arbitrary sigmoid function, then the real probability machine $M_{proba} = \Sigma_{proba} \circ \phi$ is more precise than (or at least as precise as) the machine $M = \Sigma \circ \phi$, in the sense that the ROC curve of M_{proba} lies above the ROC curve of M. In other words, for any false positive level $\alpha \in]0,1[$, if σ and σ_{proba} are the corresponding thresholds such that

$$FP_{M_{proba}}(\sigma_{proba}) = FP_M(\sigma) = \alpha, \tag{18}$$

then

$$TP_{M_{proba}}(\sigma_{proba}) \geq TP_M(\sigma). \tag{19}$$

(ii) The ROC curve of the real probability machine M_{proba} is convex.

Proof. Fix an arbitrary false positive level $\alpha \in]0,1[$ and $\sigma_{proba}, \sigma \in]0,1[$ the two corresponding threshold values as in the statement of the proposition. Then, we have the following formula for α:

$$\alpha = \frac{\int_{\{\varphi \in \Phi | \Sigma_{proba}(\varphi) \geq \sigma_{proba}\}} (1 - \Sigma_{proba}(\varphi)) d\varphi}{\int_\Phi (1 - \Sigma_{proba}(\varphi)) d\varphi} = \frac{\int_{\{\varphi \in \Phi | \Sigma(\varphi) \geq \sigma\}} (1 - \Sigma_{proba}(\varphi)) d\varphi}{\int_\Phi (1 - \Sigma_{proba}(\varphi)) d\varphi}, \tag{20}$$

(where $\int_\Phi (1 - \Sigma_{proba}(\varphi)) d\varphi = P_\Omega(Y = 0)$ is the probability measure of the negative set $\{x \in \Omega \mid Y(x) = 0\}$), which implies that

$$\int_{\{\varphi \in \Phi | \Sigma_{proba}(\varphi) \geq \sigma_{proba}\}} (1 - \Sigma_{proba}(\varphi)) d\varphi = \int_{\{\varphi \in \Phi | \Sigma(\varphi) \geq \sigma\}} (1 - \Sigma_{proba}(\varphi)) d\varphi. \tag{21}$$

To simplify the notations, put

$$A = \{\varphi \in \Phi | \Sigma_{proba}(\varphi) \geq \sigma_{proba}\} \text{ and } B = \{\varphi \in \Phi | \Sigma(\varphi) \geq \sigma\}. \tag{22}$$

Then, we have $\int_A (1 - \Sigma_{proba}(\varphi))d\varphi = \int_B (1 - \Sigma_{proba}(\varphi))d\varphi$, which implies that

$$\int_{A \setminus B} (1 - \Sigma_{proba}(\varphi))d\varphi = \int_{B \setminus A} (1 - \Sigma_{proba}(\varphi))d\varphi. \tag{23}$$

Since $(1 - \Sigma_{proba}(\varphi)) \leq 1 - \sigma_{proba}$ on $A \setminus B$ while $(1 - \Sigma_{proba}(\varphi)) > 1 - \sigma_{proba}$ on $B \setminus A$, we must have that $P(A \setminus B) \geq P(B \setminus A)$, which implies that

$$\int_{A \setminus B} \Sigma_{proba}(\varphi)d\varphi \geq \sigma_{proba} P(A \setminus B) \geq \sigma_{proba} P(B \setminus A) \geq \int_{B \setminus A} \Sigma_{proba}(\varphi)d\varphi, \tag{24}$$

which implies that

$$\int_A \Sigma_{proba}(\varphi)d\varphi \geq \int_B \Sigma_{proba}(\varphi)d\varphi. \tag{25}$$

This last inequality means exactly that the true positive level of Σ_{proba} at the false positive level α is greater than or equal to the true positive level of Σ at the same false positive level. In other words, the ROC curve of the probability sigmoid function Σ_{proba} lies above the ROC curve of Σ everywhere, i.e., Σ_{proba} is the optimal sigmoid function.

The two ROC curves coincide if and only if, in the above formulas, B coincides with A (up to a set of measure zero) for every false positive level α, and it basically means that Σ is obtained from Σ_{proba} by composing it with a monotonous function. In other words, up to a reparametrization of the sigmoid values, the probability sigmoid function is the only optimal sigmoid function.

The convexity of the ROC curve of the probability sigmoid function Σ_{proba} follows directly from its construction, which ensures that the conditional event probability is nondecreasing (the higher the sigmoid value σ, the higher the conditional probability value is, which is obvious because this value is equal to σ in our construction). See Theorem 3 of [22]. Indeed, denote by

$$\alpha(\sigma) = \frac{\int_{\{\varphi \in \Phi | p(\varphi) \geq \sigma\}} (1 - \Sigma_{proba}(\varphi))d\varphi}{\int_\Phi (1 - \Sigma_{proba}(\varphi))d\varphi} \quad \text{and} \quad \beta(\sigma) = \frac{\int_{\{\varphi \in \Phi | \Sigma_{proba}(\varphi) \geq \sigma\}} p(\varphi)d\varphi}{\int_\Phi \Sigma_{proba}(\varphi)d\varphi} \tag{26}$$

the false negative and false positive levels at threshold σ for the probability sigmoid function Σ_{proba}, then we have

$$\frac{d\beta}{d\alpha} = \frac{\int_\Phi (1 - \Sigma_{proba}(\varphi))d\varphi}{\int_\Phi \Sigma_{proba}(\varphi)d\varphi} \cdot \frac{\sigma}{1 - \sigma}, \tag{27}$$

which is an increasing function in σ, but a decreasing function in α, because α itself is a decreasing function in σ. Hence, β is a concave function in α, which means that the ROC curve is convex. □

4. Differential Machine Learning and Loss Functions

The main idea of machine learning is that we have not just one, but a large family of machines $M_\theta : \Omega \to [0, 1]$ that depend on some vector parameter $\theta \in \Theta$, where Θ is a multi-dimensional space, and the learning process consists of changing θ step by step, e.g.,

$$\theta = \theta_0 \mapsto \theta_1 \mapsto \theta_2 \mapsto \ldots \mapsto \theta_n \mapsto \ldots \tag{28}$$

in order to improve the performance or the precision of M_θ. In *differential learning*, one constructs a **loss function**:

$$L : \Theta \to \mathbb{R} \tag{29}$$

which acts as a proxy for the precision of the machines (the lower the loss $L(\theta)$, the higher the precision of the machine M_θ in some sense) and uses the stochastic gradient descent method to find a minimal point θ (or a near-minimal point) for the loss function L. That

(near-)minimal point θ would logically correspond to a (near-)optimal machine M_θ. Theoretically, L is equal to the integral over the whole data space Ω of a **pointwise loss function** ℓ:

$$L(\theta) = \int_{x\in\Omega} \ell(M_\theta(x), Y(x))dP_\Omega \tag{30}$$

A priori, one can choose the loss function ℓ as one pleases, the only natural restriction being that the further $m = M_\theta(x)$ is away from the ground truth $y = Y(x)$, the higher the loss $\ell(m, y)$ should be, and if $m = y$, then there is no loss, i.e., $\ell(y, y) = 0$. The two most popular loss functions are the **quadratic loss**:

$$\ell_{quadratic}(m, y) = (m - y)^2 \tag{31}$$

and the so-called binary **cross-entropy**, which corresponds to the function

$$\ell_{crossentropy}(m, y) = -\ln(1 - |m - y|) \tag{32}$$

However, one can choose many other loss functions. For example, the following **quartic loss** function will work very well in many problems:

$$\ell_{quartic}(m, y) = (m - y)^2 + (m - y)^4 \tag{33}$$

One can even try to use non-smooth, non-convex loss functions, for example

$$\ell_{broken}(m, y) = \min(|m - y|, 0.5)^2 + \max(|m - y| - 0.25, 0) \tag{34}$$

Below, we give a theoretical explanation of the following facts:

(i) The quadratic loss and the cross-entropy are the two most natural loss functions;
(ii) Convex loss functions such as $\ell_{quartic}$ are good loss functions in the sense that their minimizers are optimal machines in terms of accuracy);
(iii) Nonconvex loss functions such as ℓ_{broken} may lead to very erratic results (stochastic traps) in machine learning.

One can rewrite the loss $L(M) = \int_{x\in\Omega} \ell(\Sigma(\phi(x)), Y(x))dP_\Omega$ of a binary machine $M = \Sigma \circ \phi$ as an integral on the information space Φ and then call it the loss of the sigmoid function Σ, as follows:

$$L(\Sigma) = \int_\Phi \Big[(1 - \Sigma_{proba}(\varphi)) \cdot \ell(\Sigma(\varphi), 0) + \Sigma_{proba}(\varphi) \cdot \ell(\Sigma(\varphi), 1)\Big] d\varphi. \tag{35}$$

(For each given φ, the value of $\ell(\Sigma(\phi(x)), Y(x))$ under the condition $\phi(x) = \varphi$ will be equal to $\ell(\Sigma(\varphi), 0)$ with probability $(1 - \Sigma_{proba}(\varphi))$ and equal to $\ell(\Sigma(\varphi), 1)$ with probability $\Sigma_{proba}(\varphi)$. The integrand $(1 - \Sigma_{proba}(\varphi)) \cdot \ell(\Sigma(\varphi), 0) + \Sigma_{proba}(\varphi) \cdot \ell(\Sigma(\varphi), 1)$ in the above formula is nothing but the integral of $\ell(\Sigma(\phi(x)), Y(x))$ over the space $\{x \in \Omega,\ \phi(x) = \varphi\}$) with respect to the conditional probability measure on that space; that is why we have the above formula).

For example, in the case of the cross-entropy loss, we have the integral formula, whose integrand is really a cross-entropy:

$$L_{crossentropy}(\Sigma) = \int_\Phi -\Big[(1 - \Sigma_{proba}(\varphi)) \cdot \ln(1 - \Sigma(\varphi)) + \Sigma_{proba}(\varphi) \cdot \ln(\Sigma(\varphi))\Big] d\varphi. \tag{36}$$

The above examples of loss functions are *symmetric*, in the sense that they treat the losses in negative cases ($Y = 0$) and the losses in positive cases ($Y = 1$) on an equal footing. However, due to huge data imbalance in some problems (for example, when the number of positive cases is just $1/1000$ the number of negative cases), in practice, it is sometimes better to use *asymmetric loss functions* instead of symmetric loss functions. Given a function:

$$f : [0, 1] \to \mathbb{R} \tag{37}$$

which is increasing and such that $f(0) = 0$, we can create a family of **asymmetric loss functions** ℓ_c depending on an **asymmetry coefficient** $c > 0$ by the following formula:

$$\ell_c(m, y) = (1 - y)f(m) + cyf(1 - m). \tag{38}$$

Since our ground truth admits only two values $y = 0$ and $y = 1$, the above formula simply means that the loss is equal to $f(m)$ if $y = 0$ and is equal to $cf(1 - m)$ if $y = 1$, so the negative cases and the positive cases are treated differently in the total loss. For example, when $c = 100$, then it is like every positive case is counted one hundred times while every negative case is counted only once. As such, the asymmetry coefficient can be used to offset data imbalances.

The two Formulas (35) and (38) give us the following formula for the loss of a machine $M = \Sigma \circ \phi$ with respect to a given generating function f and asymmetry coefficient c:

$$L(\Sigma) = \int_{\Phi} \Big[(1 - \Sigma_{proba}(\varphi)) \cdot f(\Sigma(\varphi)) + c\Sigma_{proba}(\varphi) \cdot f(1 - \Sigma(\varphi))\Big] d\varphi. \tag{39}$$

For each given $\varphi \in \Phi$, the integrand $(1 - \Sigma_{proba}(\varphi)) \cdot f(\Sigma(\varphi)) + c\Sigma_{proba}(\varphi) \cdot f(1 - \Sigma(\varphi))$ in the above integral formula can be written as a function of one variable $\sigma = \Sigma(\varphi)$ and one parameter $p = \Sigma_{proba}(\varphi)$ (we cannot change p, but can choose our sigmoid function Σ, i.e., choose σ, in order to minimize the loss):

$$g(\sigma) := (1 - p)f(\sigma) + cpf(1 - \sigma). \tag{40}$$

Minimizing the loss $L(\Sigma)$ means minimizing $g(\sigma)$ for each φ. In other words, a sigmoid function Σ is a minimizer of the loss function $L(\Sigma)$ given by Formula (39) if and only if (up to a set of measure zero) for each $p \in [0, 1]$ and each $\varphi \in \Omega$ such that $\Sigma_{proba}(\varphi) = p$, we have

$$\Sigma(\varphi) = \underset{\sigma}{\operatorname{argmin}}[(1 - p)f(\sigma) + cpf(1 - \sigma)] \tag{41}$$

This last equation leads us to the following very interesting result about the *naturality* of the classical quadratic loss function (the case with $f(\sigma) = \sigma^2$ and $c = 1$) and the binary cross-entropy (the case with $f(\sigma) = -\ln(1 - \sigma)$ and $c = 1$):

Proposition 3. *With the above notations, we have:*
(i) The real probability machine is the only loss minimizer for the quadratic loss function.
(ii) The real probability machine is also the only loss minimizer for the binary cross entropy function.

Proof. (i) The quadratic loss case. As discussed above, a sigmoid function $\Sigma : \Phi \to [0, 1]$ is a minimizer of the quadratic loss function if and only if

$$\Sigma(\varphi) = \underset{\sigma}{\operatorname{argmin}}\Big((1 - p)\sigma^2 + p(1 - \sigma)^2\Big). \tag{42}$$

for each $p \in [0,1]$ and each $\varphi \in \Omega$ such that $\Sigma_{proba}(\varphi) = p$.

The quadratic function $g(\sigma) := (1 - p)\sigma^2 + p(1 - \sigma)^2$ has its derivative equal to $g'(\sigma) = 2(1 - p)\sigma + 2p(\sigma - 1) = 2(\sigma - p)$, and the equation $g'(\sigma) = 0$ has a unique solution $\sigma = p$. This point $\sigma = p = \Sigma_{proba}(\varphi)$ is the unique minimal point for the function $g(\sigma)$. It follows that the loss $L(\Sigma)$ achieves its minimal at (and only at) the function $\Sigma(\varphi) = \Sigma_{proba}(\varphi)$, i.e., when the machine is the real probability machine.

(ii) The cross-entropy case. In this case, the minimizer of the loss function satisfies the equation

$$\Sigma(\varphi) = \underset{\sigma}{\operatorname{argmin}}(-(1 - p)\ln(1 - \sigma) - p\ln(\sigma)). \tag{43}$$

for each $p \in [0, 1]$ and each $\varphi \in \Omega$ such that $\Sigma_{proba}(\varphi) = p$.

The logarithmic function $h(\sigma) = -(1-p)\ln(1-\sigma) - p\ln(\sigma)$ tends to infinity at both ends (when σ tends to 0 and when σ tends to 1) and has its derivative equal to $h'(\sigma) = \frac{1-p}{1-\sigma} - \frac{p}{\sigma} = \frac{\sigma - p}{\sigma(1-\sigma)}$, and the equation $h'(\sigma) = 0$ has a unique solution $\sigma = p = \Sigma_{proba}(\varphi)$. This value of σ is the unique minimal point of $h(\sigma)$, so similar to the previous case, the probability function $\Sigma(\varphi) = \Sigma_{proba}(\varphi)$ is also the unique minimizer of the cross-entropy loss function. □

5. Convex Loss Functions and Distorted Probabilities

The following proposition shows that, not only the cross-entropy and the quadratic loss function can lead to the real probability machine (which is the most natural and most precise machine according to Proposition 2), but all the other convex loss functions can also lead to this optimal machine, up to a reparametrization (distortion of the probabilities).

Proposition 4. *Let $f : [0,1[\to \mathbb{R}_+$ be an arbitrary strictly convex increasing continuously differentiable function such that either $f'(0) = 0$ or $\lim_{\sigma\to 1} f'(\sigma) = +\infty$, $c > 0$ be an arbitrary positive number (the asymmetry efficient), and*

$$L(M) = \int_{x\in\Omega} [(1 - Y(x)) \cdot f(M(x)) + cY(x) \cdot f(1 - M(x))]dx \tag{44}$$

be the loss of a machine M for a given binary classification problem Y, measured by f and c.

Then, the minimizer $M = \Sigma \circ \phi$ for the loss function $L(M)$ is just a reparametrization of the real probability machine. In other words, there is an increasing bijection $g : [0,1] \to [0,1]$ such that the machine with the sigmoid function:

$$\Sigma(\varphi) := g(\Sigma_{proba}(\varphi)) \tag{45}$$

has the minimal loss with respect to L.

Proof. Recall from (35) that the loss $L(M)$ for a machine M can be written as a loss for its sigmoid function Σ as follows:

$$L(M) = L(\Sigma) = \int_\Phi \Big[\Big(1 - \Sigma_{proba}(\varphi)\Big) f(\Sigma(\varphi)) + c\Sigma_{proba}(\varphi) f(1 - \Sigma(\varphi))\Big] d\mu_\phi \tag{46}$$

In order to minimize the loss $L(\Sigma)$ over all functions $\Sigma : \Phi \to [0,1]$, we have to minimize $(1-p)f(\sigma) + cpf(1-\sigma)$ for each given $p = \Sigma_{proba}(\varphi)$ over all σ and put

$$\Sigma(\varphi) = \operatorname*{argmin}_\sigma [(1-p)f(\sigma) + cpf(1-\sigma)] \tag{47}$$

to obtain the optimal sigmoid function with respect to the loss function L.

Under our assumptions about the function f, the minimal value of the function $(1-p)f(\sigma) + cpf(1-\sigma)$ (for a given p) is attained at the point σ where its derivative vanishes, i.e., $(1-p)f'(\sigma) - cpf'(1-\sigma) = 0$, or

$$\frac{f'(\sigma)}{f'(1-\sigma)} = \frac{cp}{1-p}. \tag{48}$$

Notice that $h(\sigma) := \frac{f'(\sigma)}{f'(1-\sigma)}$ is a strictly increasing continuous function, with $h(0) = 0$ and $\lim_{\sigma\to 1} h(\sigma) = +\infty$; hence, for each p ($p \in [0,1]$), there is a unique σ such that $h(\sigma) = \frac{cp}{1-p}$, and moreover, this value of σ increases when p increases. In other words, there is an increasing bijection

$$g : [0,1] \to [0,1] \tag{49}$$

such that $\sigma = g(p)$ will satisfy the above equation, i.e., $h(g(p)) = \frac{cp}{1-p}$. This implies that $\Sigma(\varphi) := g(p) = g(\Sigma_{proba}(\varphi))$ is the sigmoid function whose corresponding machine $M = \Sigma \circ \phi$ has the minimal loss with respect to the loss function L.

Of course, this machine is just a reparametrization of the real probability machine by the reparametrization function g. □

Remark 5. *If the loss function is not convex (i.e., the function f in the formula of the loss function is not convex), then Equation (48) may have zero solution or many solutions instead of a unique solution in the interval $[0,1]$. If there is no solution, then it means that the minimal value of $(1-p)f(\sigma) + cpf(1-\sigma)$ falls at $\sigma = 0$ or $\sigma = 1$, and the optimizer for L will be an kind of extreme machine whose values will be mostly just 0 or 1 instead of some kind of probability numbers; such a machine will not be very useful. When there are many solutions, then a minimizer Σ for the loss function will not be a reparametrization of the probability sigmoid function either, i.e., the machine that minimizes the loss function will not be an optimal machine.*

Remark 6. *In practice, hyper-convex loss functions such as the quartic loss are often preferable to the square loss and the cross-entropy, even though they manifestly lead to machines that give distorted probabilities. One of the reasons is their* ***focality*** *(see, e.g., [10,16]), which allows the machine learning process to concentrate more of its learning on the difficult cases instead of "revising" too much the easy cases.*

Remark 7. *Every probability is in fact a conditional probability and can be distorted one way or another. For example, we do not know what the real probability distribution on the dataset Ω is, especially when Ω is very large, and the samples that we have just a small subset of Ω. Therefore, even if we use the cross-entropy or the square loss and a very good machine learning method, there is no guarantee that the obtained machine will give real probabilities, and it is better to assume that all the obtained sigmoid values will be just distorted probabilities. To go back from distorted probabilities to real probabilities, i.e., to find the reparametrization function g such that $\Sigma(\varphi) = g(\Sigma_{proba}(\varphi))$, one may make large-scale post-training tests of the machine (similar to clinical studies for medical products). This problem of going from distorted probabilities to true probabilities is a well-known problem of the calibration of probabilities in machine learning; see, e.g., [28,29].*

Remark 8. *Distorted probabilities may actually be a good thing. For example, if we have a cancer detection machine that gives the sigmoid value 0.1 for a patient, how should this number be interpreted? Since this number is very small, one may want to dismiss it as "very low cancer risk". However, if it is a real probability number, i.e., the chance of having cancer is 10%, then that number is already high enough to be taken very seriously. It would be better if the sigmoid value 0.1 would correspond to the cancer probability of just 0.001 for example (and to the probability of having a late-stage cancer much smaller than that).*

6. Non-Convex Loss Functions and Stochastic Traps

We performed many experiments of deep learning with both "good" (hyper-convex loss functions, such as the quartic loss function in Formula (33)) and "bad" loss functions (non-convex functions, such as the broken loss function in Formula (34) and other functions, which the reader can probably invent easily by herself/himself). We used only well-known neural networks such as VGG16 [30] and standard data augmentation methods, absolutely nothing fancy. The purpose was not to achieve the best-performing AI models, but to study phenomena created by different loss functions.

For example, we performed hundreds of experiments with the binary classification problems such as "cat versus not-cat", "dog versus not-dog" on the well-known public dataset called CIFAR-10 collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton (see [31,32]). This dataset consists of 60 thousand 32×32 color images divided into 10 equal classes: cat, dog, ship, plane, etc. The experiments confirmed our theoretical reasoning that hyper-convex loss functions lead to good results in general, while non-convex loss

functions very often (more than half of the time in our experiments) lead to *stochastic traps* in the parameter space: the stochastic descent of the machine learning process falls into those places of low accuracy, gets trapped there, and cannot get out.

Due to the stochastic nature of machine learning, sometimes, the machine does break out of the stochastic trap after being stuck there for many epochs (learning steps). Sometimes, the trap is so big or so strong that the machine breaks out of it only to fall back into it again after some machine learning epochs. An illustration of a stochastic trap that we observed is shown in Figure 2.

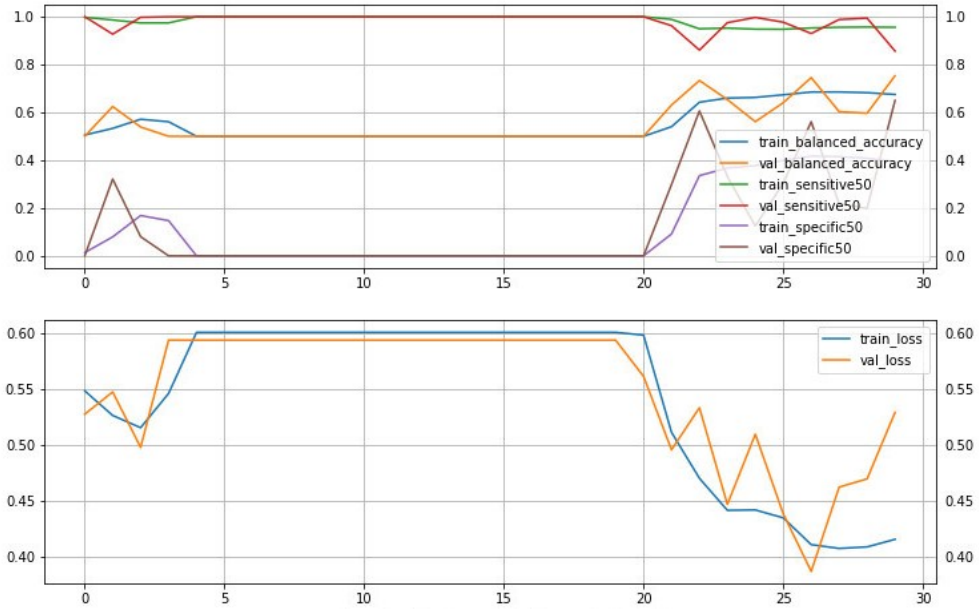

Figure 2. Getting in the trap and then getting out after machine learning epochs. "Cat vs. not-cat" problem on CIFAR-10, trained with VGG16 and the "broken loss" function (34). While in the trap, the machine takes every image for a cat (sensitivity = 1, specificity = 0 at threshold $\sigma = 0.5$).

The problem of describing precisely the mechanisms for stochastic traps in machine learning is a very large and interesting problem, but it is outside of the scope of this paper. Here, we just wanted to show our observation that ill-designed non-convex loss functions may be responsible for such traps.

Author Contributions: The first author (P.B.L.) is a Ph.D. student working under the guidance of the second author (Z.T.N.). The two authors worked together on the ideas, as well as the details and the writing of this paper and contributed equally to this research work. All authors have read and agreed to the published version of the manuscript.

Funding: This theoretical research was partially financially supported by Torus Actions SAS (https://torus.ai, accessed on 18 August 2020) and BelleTorus Corp. (https://belle.ai, accessed on 16 August 2020).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Le Hong Van (Institute of Mathematics, Czech Academy of Science), as well as the Referees of this paper for many critical remarks, which helped us improve the paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Cucker, F.; Smale, S. On the Mathematical Foundation of Learning. *Bull. Am. Math. Soc.* **2002**, *39*, 1–49. [CrossRef]
2. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA; London, England, 2016.
3. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: New York, NY, USA, 2001.
4. Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
5. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms* , 1st ed.; Chapman Hall/CRC: Boca Raton, FL, USA, 2012.
6. Cristianini, N.; Shawe Taylor, J. *An Introduction to Support Vector Machines*; Cambridge University Press: Cambridge, UK, 2000.
7. Hennig, C.; Kutlukaya, M. Some thoughts on the design of loss functions. *REVSTAT–Stat. J.* **2007**, *5*, 19–39.
8. Lapin, M.; Hein, M.; Schiele, B. Analysis and Optimization of Loss Functions for Multiclass, Top-k, and Multilabel Classification. *arXiv* **2016**, arXiv:1612.03663.
9. Lee, T.-H. *Loss Functions in Time Series Forecasting* ; University of California: Los Angeles, CA, USA, 2007.
10. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002.
11. Rosasco, L.; De Vito, E.; Caponnetto, A.; Piana, M.; Verri, A. Are Loss Functions All the Same? *Neural Comput.* **2004**, *16*, 1063–1076. [CrossRef] [PubMed]
12. Shen, C.; Roth, H.R.; Oda, H.; Oda, M.; Hayashi, Y.; Misawa, K.; Mori, K. On the influence of Dice loss function in multi-class organ segmentation of abdominal CT using 3D fully convolutional networks. *arXiv* **2018**, arXiv:1801.05912v1.
13. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Cardoso, M.J. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, LNCS 10553*; Springer: Berlin/Heidelberg, Germany, 2017.
14. Wu, L.; Tian, F.; Xia, Y.; Fan, Y.; Qin, T.; Lai, J.; Liu, T.-Y. Learning to Teach with Dynamic Loss Functions. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
15. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss Functions for Image Restoration With Neural Networks. *IEEE Trans. Comput. Imaging* **2017**, *3*, 47–57. [CrossRef]
16. Abraham, N.; Khan, N.M. A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019.
17. Gosh, A.; Kumar, H.; Sastry, P.S. Robust loss functions under label noise for deep neural networks. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco CA, USA, 4–9 February 2017.
18. James, G.M. Variance and Bias for General Loss Functions. *Mach. Learn.* **2003**, *51*, 115–135. [CrossRef]
19. Kervadec, H.; Bouchtiba, J.; Desrosiers, C.; Dolz, E.G.J.; Ayed, I.B. Boundary loss for highly unbalanced segmentation. *arXiv* **2019**, arXiv:1812.07032.
20. Le, B.P.; Nguyen, T.Z. Accuracy measures and the convexity of ROC curves for binary classification problems. In Proceedings of the Third International Conference on Artificial Intelligence and Computational Intelligence, Hanoi, Vietnam, 8–12 November 2021; 9p.
21. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
22. Gneiting, T.; Vogel, P. Receiver Operating Characteristic (ROC) Curves. *Mach. Learn.* **2021**, 1–29. [CrossRef]
23. Hernandez-Orallo, J.; Flach, P.A.; Ferri, C. A unified view of performance metrics: Translating threshold choice into expected classification loss. *J. Mach. Learn. Res.* **2012**, *13*, 2813–2869.
24. Pesce, L.L.; Metz, C.E.; Berbaum, K.S. On the convexity of ROC curves estimated from radiological test results. *Acad. Radiol.* **2010**, *17*, 960–968.e4. [CrossRef] [PubMed]
25. Swets, J.A.; Dawes, R.M.; Monahan, J. Psychological science can improve diagnostic decisions. *Psychol. Sci. Public Interest* **2000**, *1*, 1. [CrossRef] [PubMed]
26. Wikipedia Page on ROC. Available online: https://en.wikipedia.org/wiki/Receiver_operating_characteristic (accessed on 16 August 2020).
27. Livieris, I.E.; Kiriakidou, N.; Stavroyiannis, S.; Pintelas, P. An Advanced CNN-LSTM Model for Cryptocurrency Forecasting. *Electronics* **2021**, *10*, 287. [CrossRef]
28. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: Berlin/Heidelberg, Germany, 2013.
29. Niculescu-Mizil, A.; Caruana, R. Predicting good probabilities with supervised learning. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 625–632. [CrossRef]
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Krizhevsky, A. CIFAR Dataset. Available online: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 16 August 2020).
32. Technical Report: Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 16 August 2012).