Optimization of deep learning precipitation models using categorical binary metrics.

Pablo R. Larraondo[1], Luigi J. Renzullo[1], Albert I. J. M. Van Dijk[1], Iñaki Inza[2], Jose A. Lozano[2]

[1]Fenner School of Environment and Society, Australian National University, Canberra, Australia
[2]Intelligent Systems Group, University of the Basque Country, Donostia, Spain

Key Points:

- A methodology for optimizing neural network models based on categorical binary indices is introduced
- A novel multi-objective loss function combining continuous and categorical binary indices is presented.
- The experimental section tests this generic methodology training a neural network for estimating precipitation.
- Results of the experiments are tested using well-known metrics in weather analysis, such as ROC curves.

Corresponding author: Pablo R. Larraondo, pablo.larraondo@anu.edu.au

Abstract

This work introduces a methodology for optimizing neural network models using a combination of continuous and categorical binary indices in the context of precipitation forecasting. Probability of detection or false alarm rate are popular metrics used in the verification of precipitation models. However, machine learning models trained using gradient descent cannot be optimized based on these metrics, as they are not differentiable. We propose an alternative formulation for these categorical indices that are differentiable and we demonstrate how they can be used to optimize the skill of precipitation neural network models defined as a multi-objective optimization problem. To our knowledge, this is the first proposal of a methodology for optimizing weather neural network models based on categorical indices.

Plain Language Summary

Deep neural networks have recently demonstrated great versatility and an unprecedented capacity to model complex problems. In weather modeling, these algorithms have been applied to solve different problems. This is a promising area of research, given the availability of large volumes of weather data and increasingly powerful computers.

Neural network models can learn to solve problems based on a metric, which the model tries to optimize. However, the quality of weather models is measured using a large variety of metrics, which can be a challenge when choosing which metric the model should optimize.

In the case of precipitation, categorical binary metrics are a popular choice to asses the quality of a model. These metrics reduce precipitation to a 'yes' or 'no' event and the results of the predicting model can be compared with the actual observations. This method is simple, yet powerful and a large number of indices and statistics have been developed to assess different aspects of the quality of precipitation models.

As precipitation models are commonly assessed using these categorical binary metrics, it would be very convenient to optimize models based on them. Unfortunately, the mathematical nature of these metrics makes them unsuitable for optimizing deep learning models.

In this work we present an alternative formulation for these categorical binary indices which can be used to train models. We demonstrate how a deep learning model can be trained to generate better quality precipitation data.

1 Introduction

It is increasingly common in meteorology to use machine learning approaches for identifying patterns in the atmosphere using large amounts of historical data. (Weyn et al., 2019; Scher & Messori, 2019; Dueben & Bauer, 2018; Ukkonen & Mäkelä, 2019). This approach, of extracting the underlying physical relationships in the atmosphere from data, opens an opportunity to explore new algorithms that optimise the output based on different verification metrics. In this work, we propose a methodology to train neural network precipitation models using a loss function which combines continuous and binary, or dichotomous [*yes, no*], metrics.

Verification dichotomous events has been extensively explored in the context of weather forecasting (Stephenson, 2000; Casati et al., 2008; Jolliffe & Stephenson, 2012; Ebert et al., 2013). Detection of rain, frost, flood and fog are examples of dichotomous meteorological events.

Verification of categorical binary events usually starts with the construction of a contingency table, which represents the frequency of "yes" and "no" model predictions and observed occurrences. In weather forecasting, thresholds are usually defined to categorically determine the occurrence of weather events from continuous variables by being above or below these thresholds. Several popular indices can be derived from contingency tables, such as Probability Of Detection (POD) and False Alarm Rate (FAR). These indices allow measuring different aspects of the quality of dichotomous predictive models and are popular evaluation metrics in meteorology studies.

Gradient descent is a versatile and popular technique in machine learning and currently constitutes the de-facto methodology to train Artificial Neural Network (ANN) models. Gradient descent prescribes an iterative process that computes the derivative of the loss function (model error) and updates the model parameters following the direction that minimises this loss until a local or global minimum is reached.

Weather models trained using gradient descent can be evaluated using binary indices, but these indices cannot be naturally integrated in the optimization process, as they are not differentiable. Gradient descent requires smooth, differentiable loss functions for determining its minima points. Categorical binary indices are built using logical comparison operators ($<, >$), which define a function containing a discontinuity at the threshold point, and therefore are non-differentiable.

The problem of optimizing non-differentiable categorical classifiers has been explored before in the context of machine learning (Yan et al., 2003; Herschtal & Raskutti, 2004). However, in weather forecasting, precipitation generated by Numerical Weather Prediction (NWP) is usually a quantitative variable and is verified using a variety of quantitative and categorical verification metrics. We propose a methodology for combining both types of metrics and optimizing models that perform well using these metrics. This problem can be formulated as a Pareto or multi-objective optimization problem in which no single solution exists that simultaneously optimizes each objective individually.

In this work we present an alternative formulation of binary indices, which present the desired characteristics of being both continuous and differentiable. We show how these indices can be integrated in the loss function of weather models trained with gradient descent, learning to optimize them. In the experimental section we apply this methodology to train a deep learning network used to predict gridded total precipitation using NWP geopotential heights as input. We demonstrate how the proposed indices are used to optimize the skill of neural network models based on different categorical binary metrics. To our knowledge, this is the first proposal of a methodology for optimizing weather precipitation models using a combination of categorical and quantitative indices.

This manuscript is structured as follows: Section 2 briefly covers the derivation of classical categorical binary indices and presents the theoretical basis of the equivalent differentiable indices. Section 3 presents the data, model and experiments for testing the behaviour of the proposed indices. Section 4 presents the results of the experiments demonstrating how ANN models can be optimised using categorical binary metrics. We finish with Section 5, which provides conclusions and ideas on how the proposed methodology can be further developed in future works.

2 Methodology

2.1 Categorical binary verification metrics

Performance of binary forecasts can be measured as a function of *hits*, *misses*, *false alarms* and *true negatives*, which relate observed and forecasted events. The

four combinations of forecasts [*yes, no*] and observations [*yes, no*], called the joint distribution, can be represented using a contingency table (see Table 1).

Table 1: Contingency table for evaluating models which forecast dichotomous categorical events.

		Observed	
		Yes	No
Forecast	Yes	Hits	False Alarms
	No	Misses	True Negatives

Contingency tables are a useful way to represent the skill and errors made by deterministic models – a perfect forecast contingency table contains only hits and correct negatives, and no misses or false alarms. A variety of popular categorical statistics can be computed based on the indices in this contingency table to describe different aspects of the skill of a model. Probability Of Detection $POD = hits\ /\ (hits\ +\ misses)$, also known as hit rate, and the Probability Of False Detection $POFD = false\ alarms\ /\ (false\ alarms\ +\ true\ negatives)$ are examples of these statistics. We refer readers to *"Forecast verification: a practitioner's guide in atmospheric science"* (Jolliffe & Stephenson, 2012) for a detailed and rigorous coverage of categorical binary indices and weather forecast verification in a broader context.

Categorical binary metrics constitute also a popular choice to determine the skill of quantitative models, such as NWP (McBride & Ebert, 2000; Accadia et al., 2003). Quantitative NWP models generate a continuous range of output values, such as precipitation, wind and temperature. Contingency tables can be computed by setting a threshold value for an event and using the $[<, >]$ relational operators to transform forecast continuous values into its binary [*yes, no*] representations or [*rain, dry*] in the case of precipitation.

Logical relational operators define a step function which is normally represented using 0 to denote the boolean 'no' or 'false', and 1 for 'yes' or 'true' values. The transition between 0 and 1 happens at the threshold value creating a discontinuity or singularity at this point. These functions are non-continuous and therefore non-differentiable.

2.2 Differentiable categorical binary metrics

We propose an alternative formulation for these categorical verification indices using smooth and differentiable functions. Specifically, we use the sigmoid function to represent a smooth transition between the boolean values at the threshold point. The following formula defines the sigmoid function:

$$sigmoid(x) = \frac{1}{1 + e^{-\beta x}}$$

Figure 1 represents the sigmoid function as a differentiable alternative to the '$<$' and '$>$' step functions. Parameter β defines the slope of the sigmoid function, where larger values of β correspond to a steeper transition in the output. In the case of considering a threshold value α the sigmoid variable X gets translated by this amount, resulting in the case of the '$>$' operator:

$$sigmoid(x) = \frac{1}{1 + e^{-\beta(x-\alpha)}}$$

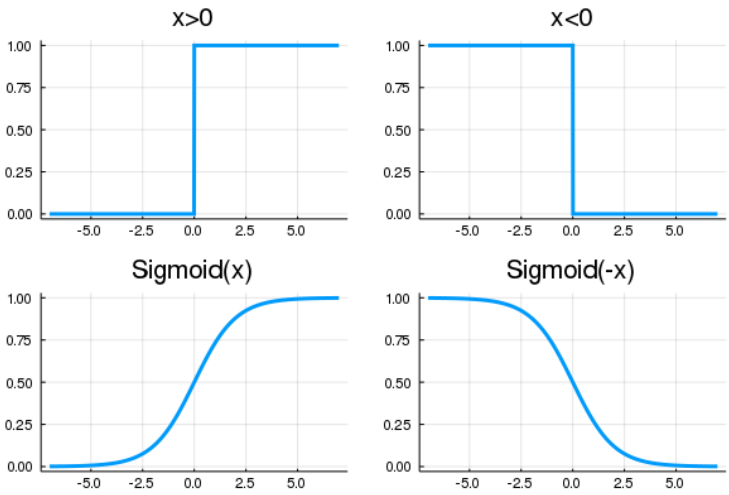


Figure 1: Comparison between the step functions representing the '<' and '>' operators with the equivalent sigmoid functions for $\alpha = 0$ and $\beta = 1$.

These sigmoid functions can be used to approximate the step function and compute a differentiable version of the contingency table previously presented. Each entry in the contingency table is calculated using an element-wise product of the vectors containing the observations and predictions compared with the threshold value α. For example, the following expression calculates $Hits$:.

$$\text{Hits} = (\text{observed} > \alpha) \odot (\text{predicted} > \alpha)$$

The previous expression can be made differentiable, by substituting the comparison in the "predicted" term with a sigmoid function. This new expression provides a gradient allowing model outputs to be optimised around the threshold. Differentiable categorical statistics such as POD or POFD can be formulated using sigmoid functionsto replace the comparison operators. For example, the differentiable versions of POD and POFD can be defined as follows:

$$\text{POD}_{\text{diff}} = \frac{\text{Hits}_{\text{diff}}}{\text{Hits}_{\text{diff}} + \text{Misses}_{\text{diff}}}$$
$$\text{POFD}_{\text{diff}} = \frac{\text{False Alarms}_{\text{diff}}}{\text{False Alarms}_{\text{diff}} + \text{True Negatives}_{\text{diff}}}$$

where :

$$\text{Hits}_{\text{diff}} = (\text{observed} > \alpha) \odot sigmoid\,(\text{predicted} - \alpha)$$
$$\text{Misses}_{\text{diff}} = (\text{observed} > \alpha) \odot sigmoid\,(-\text{predicted} - \alpha)$$
$$\text{False Alarms}_{\text{diff}} = (\text{observed} < \alpha) \odot sigmoid\,(\text{predicted} - \alpha)$$
$$\text{True Negatives}_{\text{diff}} = (\text{observed} < \alpha) \odot sigmoid\,(-\text{predicted} - \alpha)$$

Other differentiable categorical indices can be derived using the new indices in the differentiable contingency table.We refer readers interested in the derivation of these indices to the interactive notebook included in the accompanying code repository (see end of section 4).

3 Data and Experiments

This section presents the dataset, model and experiments used to assess the proposed differentiable categorical binary indices. We choose a neural network model that is trained to derive the total precipitation field using geopotential height as input.

The neural network model is trained to learn the relationship between geopotential values and total precipitation grids. Precipitation is represented using continuous values for each grid cell and during training the models learns to minimize the error between the predicted grid and ERA-Interim's total precipitation. The error of continuous precipitation fields is typically quantified using the Root Mean Squared Error (RMSE) metric (Stanski et al., 1989; Wardah et al., 2011). In machine learning, the use of Mean Squared Error (MSE) is preferred to RMSE as it is computationally simpler but equivalent in terms of their local and global minima.

3.1 Dataset

The ERA-Interim (Dee et al., 2011) global climate reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) is used to run the experiments. ERA-Interim contains reanalysis data from 1979 to present with a 6-hour temporal resolution. The spatial resolution of the dataset is approximately 80 km (reduced Gaussian grid N128) on 60 vertical levels. ERA-Interim data is publicly accessible at the ECMWF's Public Datasets web interface (Berrisford et al., 2011).

For our experiments, we choose geopotential height *(**z**)* and total precipitation *(**tp**)* variables. We consider a subset of the original data centered on the mid-latitudes rectangular region delimited by the coordinates comprising (latitude: [75, 15], longitude = [-50, 40]) degrees, which corresponds to the eastern part of the Atlantic Ocean and Europe. The temporal domain data spans from the year 1979 up to the end of 2018, with a resolution of 6 hours.

Geopotential height at the following pressure levels [1000, 900, 800, 700, 600, 500, 400, 300, 200, 100] hPa is used as input to the model and total precipitation constitutes the output or predicted field. Resulting geopotential height data are represented as a 4-dimensional numerical array with shape [58440, 80, 120, 10] corresponding to dimensions [time, latitude, longitude, height]. Similarly, the total precipitation is represented by a 3-dimensional numerical array with shape [58440, 80, 120] representing the [time, latitude, longitude] dimensions. For clarification, the ERA-Interim total precipitation parameter is originally represented using 3-hour period accumulations, which we further aggregate into 6-hour periods to match the 6-hour frequency of the geopotential height field. Figure 2 represents the geographic area *(bottom-left)* as well as the correspondence between the geopotential height and the total precipitation field time series *(right)*.

3.2 Neural Network Model

Convolutional encoder-decoder networks are a type of neural network that are able to map between multidimensional inputs and outputs by learning a compressed representation of the data. These networks have been used in many different domains to perform classification, segmentation and regression tasks (Krizhevsky et al., 2012; Long et al., 2015). In the field of meteorology similar networks have been used to model

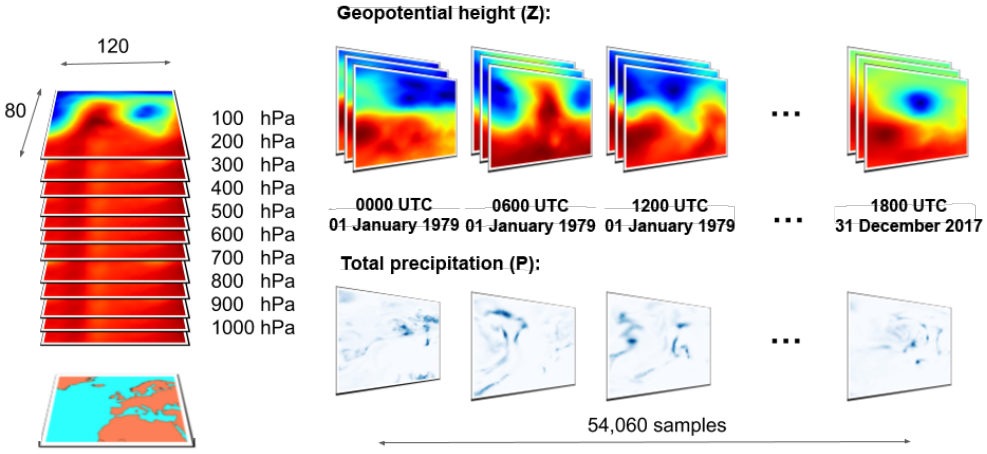


Figure 2: Representation of the geographic study area (latitude: [75, 15], longitude = [-50, 40]) degrees, and temporal extent subset from the ERA-Interim geopotential height *(top-right)* and total precipitation fields *(bottom-right)*.

extreme weather events (Liu et al., 2016) and general circulation of the atmosphere (Scher, 2018).

The proposed model is a specific type of convolutional encoder-decoder network called U-net (Ronneberger et al., 2015). We refer readers to our previous work (Larraondo et al., 2019) for a detailed comparison between different encoder-decoder architectures for the case of deriving precipitation from geopotential fields.

Figure 3 shows the architecture of the U-net network representing the changes it performs to the dimensionality of the data. This network is composed by two symmetric parts which perform a compression of the input data (encoder) and a subsequent decompression that recreates the output space (decoder). The chained convolution operations are able to capture the spatial relationships in the data at different scales and extract the relevant features that relate the input and output spaces. In our case, these are the geopotential heights (at 10 atmospheric levels) and total precipitation. Numbers at the top of this figure represent the dimensions of the images at each stage of the CNN model. Similarly, numbers at the bottom represent the channels or features at each layer of the network. The input to the network are the 10 geopotential levels and its output is one image representing the total precipitation field.

3.3 Experiment design

The objective of the experiments presented in this section is to demonstrate how the proposed differentiable categorical indices can be used to optimize the performance of neural network models. We define an objective function using a combination of these indices and use it to train a U-net model that predicts ERA-Interim total precipitation with geopotential levels as input. In particular, we choose Probability Of Detection (POD) and Probability Of False Detection (POFD) as the indices to optimise. These indices measure different aspects of the performance of a model and a combination of them is used to generate Relative Operating Characteristics (ROC) (Fawcett, 2006), which is a popular graphical method often used to represent the overall skill of categorical weather models.

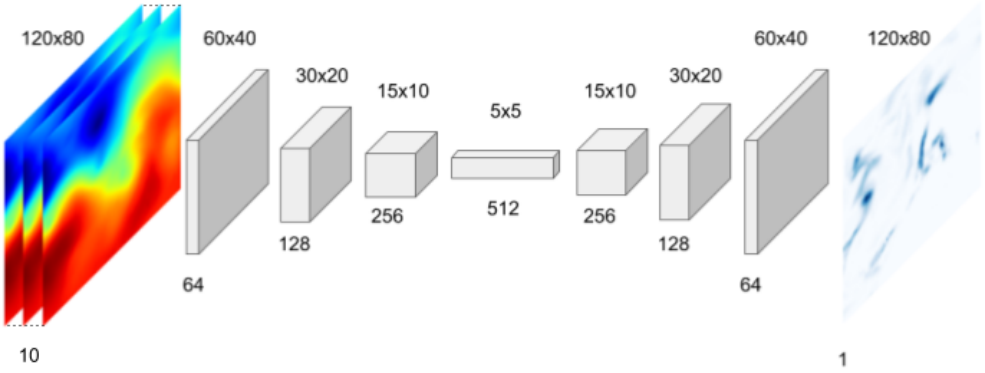


Figure 3: Transformations in the dimensionality of the data performed by a U-net convolutional encoder-decoder mapping geopotential heights to total precipitation.

POFD measures the fraction of the observed "no" events incorrectly forecast as "yes". This index ranges from 0 to 1, being 0 the score of a perfect model. POD measures the fraction of the observed "yes" events correctly forecast. POD also ranges from 0 to 1, but differently than POFD, its optimal value is 1, the maximum. Improving model performance, using these two indices, requires maximizing POD and minimizing POFD scores. POD cannot be directly used in gradient descent optimization as minimizing POD would minimize this index resulting in a model with no skill at all. For performing optimization in the right direction POD needs to be inverted, so minimization corresponds to an increase in the skill of the model. For this purpose, we use the False Negative Rate (FNR), which is the complementary index to POD, formalised through the following equation:

$$FNR = 1 - POD = 1 - \frac{hits}{hits + misses} = \frac{misses}{hits + misses}$$

The loss function used in the experiments uses the differentiable versions of FNR and POFD, defined using the equations introduced in section 2.2. The sigmoid functions used to compute the differentiable indices in the experiments set a fixed value of $\beta = 1$, which works well when using the original unscaled values of precipitation. Section 5 discusses the effect of this parameter in the results of the experiments.

Optimizing a model which combines these two metrics, FNR and POFD, results in finding a balance between two opposing forces. Reducing FNR generates overconfident models which predict precipitation everywhere whereas minimizing POFD generates under-predicting models with a complete absence of precipitation. Our objective is to use these indices to enhance the output of quantitative models trained to minimize the MSE error. In the case of precipitation prediction, MSE is commonly used for verification in the literature (Murphy, 1988; Jolliffe & Stephenson, 2012). We propose the following loss function combining MSE with the differentiable versions of FNR and POFD, following the method defined in section 3:

$$\min \{MSE + \lambda\, FNR_{diff} + \mu\, POFD_{diff}\}$$

In this equation λ and μ are constant parameters that control the relative weight of each categorical index in the overall loss function. MSE acts as a regularization term allowing the model to output continuous precipitation values in the range of the

original ERA-Interim total precipitation variable. Without the MSE term, the network would learn to differentiate categorical precipitation around the defined threshold α, but would not account for quantitative differences in the range of precipitation values.

In the next section we compare the output of the U-net model trained using different values of λ and μ in the loss function. To carry out the experiments we apply a 70/30 split over the temporal dimension of the ERA-Interim dataset for training and testing the results (training split contains years from 1979 to mid-2005 and validation split from mid-2005 to 2018). The same splits are consistently applied to train each model, so results can be fairly compared using MSE, POD and POFD as measures of performance.

The baseline performance is set by training the U-net network with MSE exclusively, which corresponds to setting both λ and μ constants to zero. Iterating through different combinations in the values of λ and μ, we compare the resulting models performance relative to the baseline to understand the influence of these categorical indices.

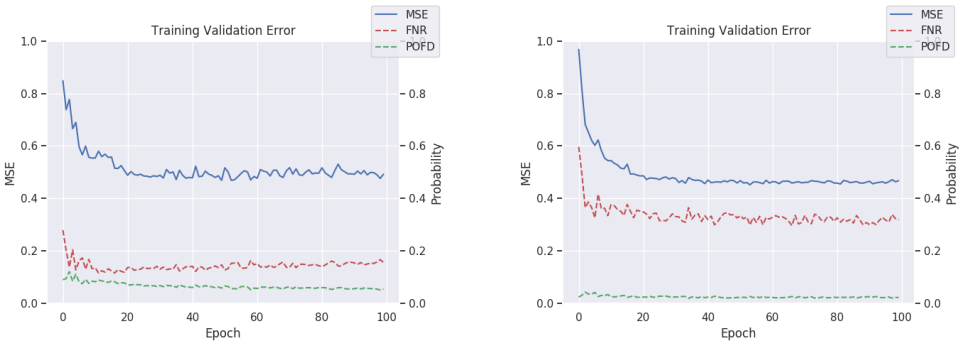
4 Results

We start this section by setting up a baseline for the model comparison using just MSE in the loss function, which corresponds to setting both $\lambda = 0$ and $\mu = 0$ in the loss function presented in the previous section. We train the U-Net model using the predefined ERA-Interim splits during 100 epochs – iterations over the whole training dataset.

During training, we assess the model performance on the validation split at the end of each epoch. The skill of the model is measured using MSE, FNR and POFD considering $\alpha = 1.0$ [mm/h] as the threshold value that discriminates precipitation. As a clarification for the reader, the standard POD and POFD indices, and not their differentiable versions, are used for verification. The differentiable indices are used in the loss function, which is derived at the backpropagation phase during the NN model training.

Figure 4: Evolution of the U-net MSE, FNR and POFD scores during training of the baseline model $\lambda = 0, \mu = 0$ using validation data.

Figure 4 shows the evolution of the validation MSE, FNR and POFD scores across the 100 epochs of training of the U-net model. The first iterations result in a



(a) $\lambda = 2$ $\mu = 0$ (b) $\lambda = 0$ $\mu = 2$

Figure 5: Evolution of the POD, POFD and MSE values for different μ and λ parameter combinations.

rapid reduction in these indices, which slow down and stabilize towards the end of the training period.

Using this model as the reference, we now train similar models using different combinations of the λ and μ constants in the loss function. We consider three scenarios, in the first two we fix one of the constants to 0 and traverse the {2, 4, 8} values with the other one. The third scenario traverses the same set of values for both variables. In total, we end up with 9 models trained with the resulting combinations of λ and μ.

Figure 5 shows the evolution during training of two of these models, in the same way as the previous figure. On the right, for the model trained with $\lambda = 2$ and $\mu = 0$, we see how the POD score is significantly lower than the baseline model at the expense of an increase in MSE and POFD. Similarly, on the left, the model trained with $\lambda = 0$ and $\mu = 2$ shows a significant decrease in the POFD score, which penalises MSE and POD.

Table 2 contains the score values of each model, with the baseline model in the first row. FNR has been converted to its complementary POD, which is a more familiar score in weather forecasting. This table shows the relationships between the values of both constants in the loss function and the resulting variation of the scores.

Figure 6 provides a visual understanding of the effect of these constants in the precipitation field learned by the models. Taking one sample from the validation split, which corresponds to the 16^{th} September 2016, we sequentially represent the original ERA-Interim total precipitation field (not used during training), the output of the baseline U-net model and, in the second row, the three models corresponding to the extremes for each of the scenarios considered.

Fig. 6c shows an extremely conservative model which only predicts rain in the regions where there is a strong signal. On the other extreme, in Fig. 6d, POD has a large weight in the loss function and the model becomes overconfident, representing precipitation values greater than the threshold $\alpha = 1$. The third model corresponding to $\lambda = 8$ and $\mu = 8$, achieves a significantly better POD score with a slight sacrifice of POFD and a significant degradation in MSE, by looking at the values in Table 2. Visually, this model tends to generate precipitation with crisper edges than the reference but it also seems to over-estimate precipitation values on average, which probably explains the increase in MSE.

Table 2: Results for the MSE, POD and POFD values over the validation dataset for different combinations of λ and μ.

Model	MSE	POD	POFD
$\lambda = 0,\ \mu = 0$	0.4148	0.7090	0.0278
$\lambda = 1,\ \mu = 0$	0.4549	0.8355	0.0550
$\lambda = 2,\ \mu = 0$	0.4923	0.8438	0.0553
$\lambda = 4,\ \mu = 0$	0.5710	0.8570	0.0584
$\lambda = 8,\ \mu = 0$	0.7944	0.8833	0.0678
$\lambda = 0,\ \mu = 1$	0.4356	0.7075	0.0277
$\lambda = 0,\ \mu = 2$	0.4675	0.6815	0.0231
$\lambda = 0,\ \mu = 4$	0.5695	0.6563	0.0200
$\lambda = 0,\ \mu = 8$	0.8360	0.6247	0.0161
$\lambda = 1,\ \mu = 1$	0.4516	0.7997	0.0433
$\lambda = 2,\ \mu = 2$	0.5305	0.8197	0.0466
$\lambda = 4,\ \mu = 4$	0.7217	0.8328	0.0490
$\lambda = 8,\ \mu = 8$	1.1311	0.8444	0.0503

The proposed objective function establishes a three way trade-off between the scores. The sensitivity of these scores is however non-symmetric. For example, Figure 7 represents the evolution of the three scores for the two scenarios where one the constants is set to 0. For small values of λ and μ there is a small performance decrease in MSE and the non-weighted variable. Performance is significantly degraded for constant values greater than 4.

Another interesting observation about the results is that POFD in the baseline model is low. The relative improvement achieved by weighting POFD with large μ values might not compensate the penalization in the rest of scores. This fact becomes apparent when assessing performance with compound indices, such as Relative Operating Characteristics (ROC) (Fawcett, 2006).

ROC is a popular metric often used to assess the skill of categorical weather forecasts (Mason, 1982; Kharin & Zwiers, 2003) which combines POD and POFD scores measured at different thresholds in a single plot. Area Under the Curve (AUC) (Marzban, 2004), measures the skill of a ROC plots in the range [0,1], being 1 the score of the perfect model.

Figure 8 represents the ROC plots and corresponding AUC scores for the baseline and $\lambda = 2$ and $\mu = 0$ model using the following threshold values $\alpha = \{0.5, 1, 2, 5, 10\}$. We can see how by increasing λ and μ the shape of the ROC plot changes by bringing the points closer to the top and left sides of the figure respectively. However, the nature of the optimization problem makes it impossible to achieve the perfect AUC score.

Increasing μ penalises POD so strongly that the resulting AUC scores are worse than the baseline model. For our models, we find the best combination at [$\lambda = 2$, $\mu = 0$], which results in an AUC score of 0.982 compared to 0.977 for the baseline model. Increasing the value of λ further penalises POFD index resulting in lower AUC scores. Using the methodology presented in this work it would be possible to design new loss functions that optimize NN models based on the AUC score.

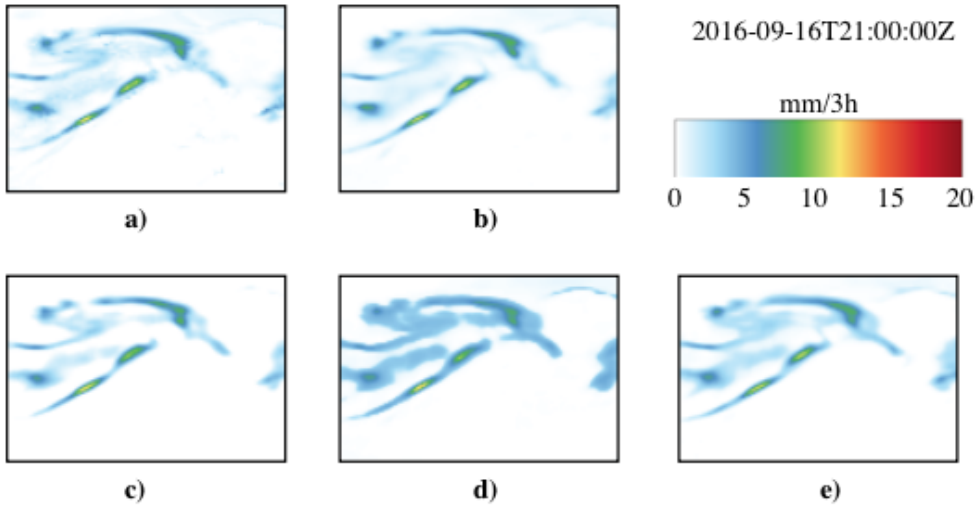
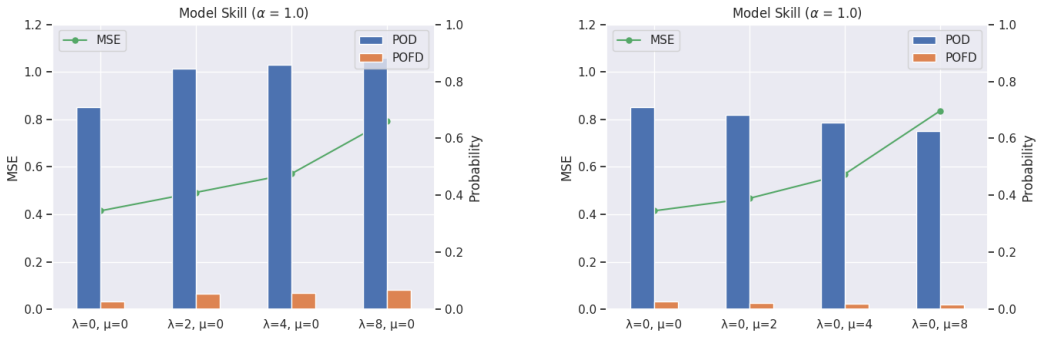


Figure 6: Comparison of precipitation outputs for the 16$^{\text{th}}$ September 2016: **a)** ref: ERA-Interim total precipitation **b)** $\lambda = 0$ $\mu = 0$ **c)** $\lambda = 0$ $\mu = 8$ **d)** $\lambda = 8$ $\mu = 0$ **e)** $\lambda = 2$ $\mu = 2$

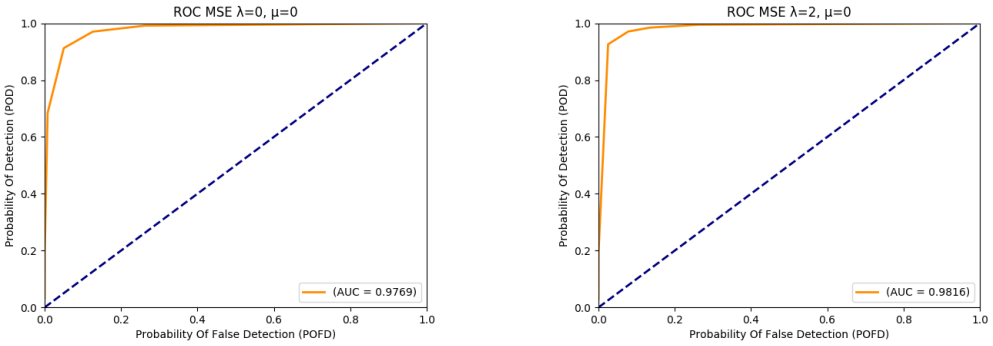


(a) Increasing λ

(b) Increasing μ

Figure 7: Evolution of the POD, POFD and MSE metrics for different combinations of λ and μ.

Figure 9 represents the results in Table 2 as points in a 3-dimensional scatter plot. For interpretability, points have been projected onto each of the three orthogonal planes defined at the origin of coordinates. Points in the vertical planes, containing the MSE axis, define a Pareto front, represented by a blue line in the figure. Pareto efficiency studies the relationship between the variables in a multi-objective optimization problem. Optimality in multi-objective problems happens when an improvement in any one individual criterion makes at least one of the other criterion worse off. Pareto fronts are a graphical representation of the optimal points, as lines or surfaces, using multi-dimensional plots. In our case, these Pareto fronts represent the relationship and trade-offs that POD and PFD present in relation to MSE. The front defines the dependency between both variables showing that it is impossible to optimize both variables simultaneously. The horizontal plane, defined by the POD and POFD axes, shows a nearly linear relationship between both variables and no Pareto relationship is observed.



(a) $\lambda = 0$ $\mu = 0$ (b) $\lambda = 2$ $\mu = 0$

Figure 8: Comparison of two ROC plots and corresponding AUC values for two values of the λ parameter.

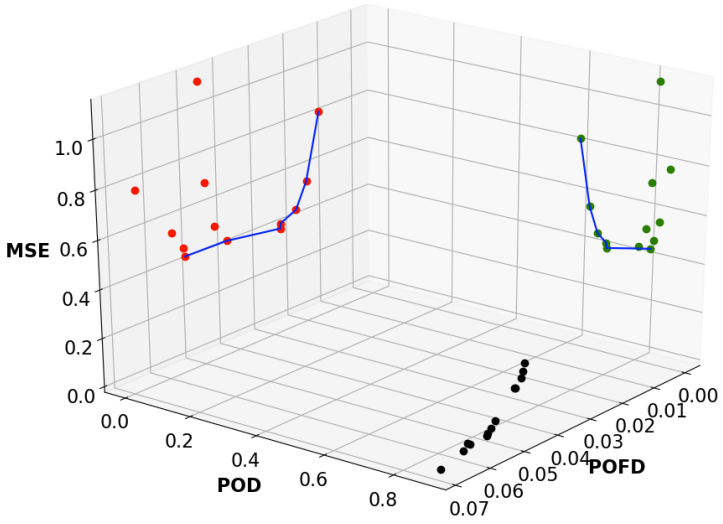


Figure 9: Pareto fronts projected for the three indices in the loss function.

The models used in this section are implemented using Keras (Chollet et al., 2018), a high-level neural networks interface written in Python and TensorFlow (Abadi et al., 2016) as back-end. The code and data to reproduce the experiments presented in this manuscript are available at this repository: `https://github.com/prl900/weather_encoders`. This repository contains a module with the implementation of differentiable version of some of the most popular categorical indices used in meteorology, which can be used independently by external models.

5 Conclusions and Future Work

This manuscript introduces a differentiable version of categorical binary indices that allow training neural networks optimizing categorical indices. These indices use the sigmoid function to approximate the logical comparison operators ($<, >$), making them continuous and therefore differentiable. Building upon these sigmoid functions,

we define differentiable versions of well-known categorical indices, such as POD and POFD. We demonstrate how these differentiable indices can be used to train models using gradient descent methods and optimize their loss function based on them.

In our experiments we use a specific deep learning NN architecture called U-net encoder-decoder to learn the spatial relationships between NWP variables, ERA-Interim geopotential height and total precipitation. The baseline model is optimised to minimize MSE and we propose a new objective function that combines MSE with POD and POFD indices. The experiment results demonstrate how the skill of the model can be optimised towards a specific index by weighting individual scores in this objective function.

Section 2.2 introduces parameter β in the definition of the sigmoid function. This parameter controls the steepness of the sigmoid function, where larger values of β result in steeper sigmoids and therefore better approximations to the step function. Although we expect β to be related to the scale of the precipitation values and have an influence in the optimization results, we did not find significant differences in the results for larger and smaller values of $\beta = 1$. At this point, we do not fully understand why the combined loss function seems to be almost invariant to changes in the shape of the sigmoid (or scale of the precipitation values). We are currently exploring this relationship, experimenting with new scale invariant loss functions and we hope to give an answer to these questions in future works.

Currently, the values of the constants that weight the different scores in the proposed objective function have to be determined relative to the model and data. Although categorical variables represent probabilities bounded between [0,1], the regression term (i.e. MSE, MAE) does not usually have an upper bound. We are planning to carry out further research to come up with new objective functions containing normalised constants which are generic and invariant under scaling of the input data.

Another interesting avenue for research is to explore the definition of high-level objective functions for optimizing models using a combination of scores. Weather forecasting verification is normally performed using a defined suite of tests and scores. Being able to design objective functions according to the verification suites would result in better performing models.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. We are grateful for the support of the Basque Government (IT-IT1244-19, Elkartek-PROMISE), the Spanish Ministry of Economy and Competitiveness (TIN2016-78365-R). Jose A. Lozano is also supported by the Basque Government through the BERC 2018-2021 program and by the Spanish Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718. Finally, the authors acknowledge the financial support given by the Earth Systems Science Organization, Ministry of Earth Science, Government of India (Grant No.IITM/MM-II/ANU/2018/INT-8) to conduct this research under the Monsoon Mission.

The dataset used to carry out the experiments in this manuscript is available through the European Centre for Medium-Range Weather Forecasts (ECMWF) at DOI `https://doi.org/10.1002/qj.828`.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others (2016).

Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265–283).

Accadia, C., Mariani, S., Casaioli, M., Lavagnini, A., & Speranza, A. (2003). Sensitivity of precipitation forecast skill scores to bilinear interpolation and a simple nearest-neighbor average method on high-resolution verification grids. *Weather and forecasting*, *18*(5), 918–932.

Berrisford, P., Dee, D., Poli, P., Brugge, R., Fielding, K., Fuentes, M., ... Simmons, A. (2011, November). *The era-interim archive version 2.0* (No. 1). Shinfield Park, Reading: ECMWF.

Casati, B., Wilson, L., Stephenson, D., Nurmi, P., Ghelli, A., Pocernich, M., ... Mason, S. (2008). Forecast verification: current status and future directions. *Meteorological Applications: A journal of forecasting, practical applications, training techniques and modelling*, *15*(1), 3–18.

Chollet, F., et al. (2018). Keras: The python deep learning library. *Astrophysics Source Code Library*.

Dee, D. P., Uppala, S., Simmons, A., Berrisford, P., Poli, P., Kobayashi, S., ... others (2011). The era-interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, *137*(656), 553–597.

Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, *11*(10), 3999–4009.

Ebert, E., Wilson, L., Weigel, A., Mittermaier, M., Nurmi, P., Gill, P., ... Fowler, T. (2013). Progress and challenges in forecast verification. *Meteorological Applications*, *20*(2), 130–139.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, *27*(8), 861–874.

Herschtal, A., & Raskutti, B. (2004). Optimising area under the roc curve using gradient descent. In *Proceedings of the twenty-first international conference on machine learning* (p. 49).

Jolliffe, I. T., & Stephenson, D. B. (2012). *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.

Kharin, V. V., & Zwiers, F. W. (2003). On the roc score of probability forecasts. *Journal of Climate*, *16*(24), 4145–4150.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Larraondo, P. R., Renzullo, L. J., Inza, I., & Lozano, J. A. (2019). A data-driven approach to precipitation parameterizations using convolutional encoder-decoder neural networks. *arXiv preprint arXiv:1903.10274*.

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., ... others (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440).

Marzban, C. (2004). The roc curve and the area under it as performance measures. *Weather and Forecasting*, *19*(6), 1106–1114.

Mason, I. (1982). A model for assessment of weather forecasts. *Aust. Meteor. Mag*, *30*(4), 291–303.

McBride, J. L., & Ebert, E. E. (2000). Verification of quantitative precipitation forecasts from operational numerical weather prediction models over australia. *Weather and Forecasting*, *15*(1), 103–121.

Murphy, A. H. (1988). Skill scores based on the mean square error and their rela-

tionships to the correlation coefficient. *Monthly weather review*, *116*(12), 2417–2424.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).

Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, *45*(22), 12–616.

Scher, S., & Messori, G. (2019). Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground. *Geoscientific Model Development*, *12*(7), 2797–2809.

Stanski, H. R., Wilson, L. J., & Burrows, W. R. (1989). Survey of common verification methods in meteorology. *WWW Technical report No. 08*.

Stephenson, D. B. (2000). Use of the "odds ratio" for diagnosing forecast skill. *Weather and Forecasting*, *15*(2), 221–232.

Ukkonen, P., & Mäkelä, A. (2019). Evaluation of machine learning classifiers for predicting deep convection. *Journal of Advances in Modeling Earth Systems*, *11*(6), 1784–1802.

Wardah, T., Kamil, A., Hamid, A. S., & Maisarah, W. (2011). Statistical verification of numerical weather prediction models for quantitative precipitation forecast. In *2011 ieee colloquium on humanities, science and engineering* (pp. 88–92).

Weyn, J. A., Durran, D. R., & Caruana, R. (2019). Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2680–2693.

Yan, L., Dodier, R. H., Mozer, M., & Wolniewicz, R. H. (2003). Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 848–855).