A CRPS Loss for Deep Probabilistic Regression

Franco Marchesoni-Acland
AI Lab
SLB
Paris, France
marchesoniacland@gmail.com

Rodrigo Alonso-Suárez
Laboratorio de Energía Solar
Universidad de la República
Salto, Uruguay
r.alonso.suarez@gmail.com

Andrés Herrera
Laboratorio de Energía Solar
Universidad de la República
Montevideo, Uruguay
97.andres.herrera@gmail.com

Josselin Kherroubi
AI Lab
SLB
Paris, France
jkherroubi@slb.com

Jean-Michel Morel
Department of Mathematics
City University of Hong Kong
Hong Kong, Hong Kong
MorelJeanMichel@gmail.com

Gabriele Facciolo
Centre Borelli
ENS Paris-Saclay
Paris, France
gfacciol@gmail.com

***Abstract*—Probabilistic regression is relevant in high-stakes areas such as energy forecasting, financial risk assessment, or healthcare. Deep models that directly output a probability distribution usually use ensembles or frame regression as classification into bins. In contrast, we propose to optimize directly for the Continuous Ranked Probability Score (CRPS), a proper scoring rule for probabilistic predictions. For the flexible histogram-like distributions, the CRPS is differentiable and can be used as the loss function of any deep model. We derive and implement the CRPS loss and showcase its performance against cross-entropy in a solar forecasting application. This new loss enables anyone to easily make any deep regressor probabilistic by simply using the new loss with the same computational cost. Surprisingly, using the CRPS loss provides superior results even when training a deterministic regressor. Code and data available at github.com/franchesoni/differentiable-crps .**

***Index Terms*—CRPS, Probabilistic Regression, Deep Learning, Solar Forecasting**

I. Introduction

Predicting probability distributions is crucial in fields where the consequences of predictions are significant, such as energy forecasting, financial risk management, and healthcare [1], [2]. Unlike traditional regression methods that output single-point estimates, probabilistic regression provides a full distribution, offering a richer understanding of the uncertainty in predictions. Already in 1950 Brier promoted probabilistic weather forecasting and introduced a score for its evaluation [3]. Today, proper scoring rules, which are minimized when the predicted distribution matches the true distribution, are widely adopted in various domains [2].

Despite the advancements, there is still room for improving probabilistic regression, particularly through the integration with deep learning. While deep learning models have gained popularity in healthcare, finance, and meteorology [4], the common approaches to probabilistic regression often involve complex ensemble methods or Bayesian neural networks. Ensembles aggregate predictions from multiple models, which can be computationally intensive and require intricate post-processing [5]. Bayesian neural networks, though theoretically robust, demand significant modifications to standard architectures and specialized knowledge [6].

This paper introduces a straightforward alternative: optimizing deep learning models directly using the Continuous Ranked Probability Score (CRPS), a proper scoring rule commonly used to evaluate probabilistic forecasts. By reformulating the regression task to optimize the CRPS, any deep learning model can be transformed into a probabilistic regressor without the need for ensembles or Bayesian techniques. The CRPS-based approach is both intuitive and efficient, allowing for seamless integration with existing deep learning frameworks and minimal additional computational overhead.

The remainder of this paper is structured as follows: Section II reviews related work, while Section III presents our method, providing the necessary background and notation, the current baseline, and the loss derivation for deep models. Section IV presents experimental results demonstrating its effectiveness. The main contributions of this work are the introduction of the CRPS loss for deep learning-based probabilistic regression and the demonstration of its practical benefits through its effective application in solar irradiance forecasting. Finally, Section V summarizes the main conclusions of this work.

II. Related Work

The study of probabilistic forecasting spans multiple disciplines, including meteorology, finance, healthcare, and statistics [2], [5]. Evaluation of probabilistic forecasts often relies on scoring rules such as the Brier score [3], valued for its strict propriety [7], although alternative scores like the log score may also be used [8].

Various methods have been developed for probabilistic regression, including Heteroscedastic Regression Models, Gaussian Processes, Bayesian Linear Regression, Quantile Regression, and Generalized Additive Models. These approaches often integrate Bayesian techniques or ensemble methods to enhance prediction accuracy.

Deep learning, known for its superior performance with large datasets, has become increasingly relevant in probabilistic forecasting. However, deep learning models typically require significant adaptation to handle probabilistic outputs. Bayesian neural networks and ensemble methods are common solutions, but they introduce complexity and computational demands [9]. This work builds on the foundation of parameterized continuous distributions, as demonstrated in models like MetNet [10] and Mixture Density Networks [11]. These models simplify the transition from deterministic to probabilistic predictions by modifying the loss function rather than the model architecture.

III. Method

A. Background

Probabilistic regression implies predicting a probability distribution. For a given input x, we observe an outcome y. More specifically, one predicts a cumulative distribution function $F(y', x) = P(y \leq y'|x)$ that is conditioned on the input x, which we will drop in what follows for simplicity. Let us also define our neural network g_θ which is a parameterized function whose parameters θ are adjusted using some gradient-based optimization technique. Note that $g_\theta(x) = \hat{y}$ where $\hat{y} \in \mathrm{R}^B$ is a real vector. The real vector $\hat{y}$ parameterizes a probability distribution. In our case we will focus on $\hat{y}$ being the probabilities assigned to each one of B bins. We call the $B+1$ bin borders $b_0 < \cdots < b_B$ and the bin intervals themselves will be $B_1, \ldots, B_B$. Note that the bins can be of arbitrary size. To ensure that $\hat{y}$ correctly determines a probability distribution it is custom to use a softmax normalization inside the network.

B. Cross Entropy Loss

Our main contender will be the cross-entropy loss used originally for classification but also used in impressive probabilistic regression applications [10]. The cross entropy loss is defined as:

$$\mathrm{CE}(\hat{y}, y) = -\sum_{i=1}^{B} p(y \in B_i) \log \hat{y}_i, \qquad (1)$$

where $p(y \in B_i) = 1_{y \in B_i}$.

C. CRPS as a metric

The Continuous Ranked Probability Score (CRPS) [12] evaluates the entire predicted distribution, measuring the integrated squared difference between the predicted CDF (F) and a step function centered at the observed value (i.e. the perfect probabilistic forecast). It captures both the accuracy and sharpness of probabilistic predictions, rewarding models that provide both precise and calibrated probability distributions. It is defined by:

$$\mathrm{CRPS}(x, y) = \int_{-\infty}^{\infty} (F(y'|x) - 1_{y \leq y'})^2 dy', \qquad (2)$$

where $1_{y \leq y'}$ is the step function with the step at $y = y'$. CRPS is the most expressive metric for probabilistic regression as it captures both accuracy and sharpness and it is also a strictly proper scoring rule [12]. A strictly proper scoring rule is one that is only minimized when the predicted distribution is identical to the true distribution of the data.

In general, the CRPS should be computed numerically. One way is to sample regular ys from a reasonable interval, get the $F(y)$ where F is the predicted CDF and compute the mean squared difference against the step function of y. For specific distributions such as the ones given by step-wise linear CDFs (e.g. originating from histogram PDFs) closed form formulas can be derived, which are exact and fast, albeit not general. We derive one such formula in the next section.

D. CRPS Loss Derivation

As mentioned above, the CRPS can be computed in closed-form for some distributions. This is true for PDFs that are step-wise constant (histogram-like). Not only that, but this computation is differentiable with respect to the predicted parameters.

Let us work with a CDF defined by bin borders b_i and point values $F(b_i) = P(y \leq b_i)$ with $i \in [0, B]$, such that $F(b_0) = 0$ and $F(b_B) = 1$. The network produces outputs $\hat{y}_i = F(b_i) - F(b_{i-1})$. We assume we do not know what happens between bin borders, thus we assign uniform probability inside each bin. This means the CDF will be composed by linear segments joining the points:

$$F(y) = F(b_i) + \frac{F(b_{i+1}) - F(b_i)}{b_{i+1} - b_i}(y - b_i), \quad y \in [b_i, b_{i+1}]. \qquad (3)$$

This can be used to compute the integral of $F(y)^2$:

$$\int_{b_i}^{b_{i+1}} \left(F(b_i) + \frac{F(b_{i+1}) - F(b_i)}{b_{i+1} - b_i}(z - b_i) \right)^2 dz$$
$$= -\frac{1}{3}\left(F(b_i)^2 + F(b_i)F(b_{i+1}) + F(b_{i+1})^2 \right)(b_i - b_{i+1}). \qquad (4)$$

To fully develop the CRPS from Eq. (2) we assume $b_0 < y < b_B$, name k the index such that $b_{k-1} < y < b_k$ and get $F(y)$ from linear interpolation between $F(b_{k-1}), F(b_k)$ using Eq. (3). The decomposition yields:

$$\begin{aligned}
\mathrm{CRPS}(F, y) &= \int_{-\infty}^{\infty} (F(y') - 1_{y \leq y'})^2 dy' \\
&= \int_{b_0}^{y} F(y')^2 dy' + \int_{y}^{b_B} (F(y') - 1)^2 dy' \\
&= \int_{b_0}^{b_B} F(y')^2 dy' - 2\int_{y}^{b_B} F(y') dy' + b_B - y \\
&= b_B - y + \sum_{i=1}^{i=B} \int_{b_{i-1}}^{b_i} F(y')^2 dy' \\
&\quad - 2\left(\int_{y}^{b_k} F(y') dy' + \sum_{i=k+1}^{i=B} \int_{b_{i-1}}^{b_i} F(y') dy' \right), \qquad (5)
\end{aligned}$$

and find that the sum of individual applications of Eq. (4) added to simple terms yields the score. For $b_B < y$ the formula is $\mathrm{CRPS}(F, y) = \int_{b_0}^{y} F(y')^2 dy' = (y - b_B) + \int_{b_0}^{b_B} F(y')^2 dy'$ and for $y < b_0$ it is $\mathrm{CRPS}(F, y) = (b_0 - y) + \int_{b_0}^{b_B} (F(y') - 1)^2 dy'$.

Two approaches can be applied for the prediction of a piece-wise linear CDF, one is histogram prediction for given bins, which means fixing $b_0 < \cdots < b_B$ and predicting $F(b_0) < \cdots < F(b_B)$ (we do this by predicting a PDF which is used to compute the different $F(b_i)$). The other approach is to fix the $F(b_i) = i/B$ and predict the bin borders $b_0 < \cdots < b_B$, akin to quantile regression. For this last case, the monotonicity $b_0 < \cdots < b_B$ must be enforced, which can be done via predicting residuals. We will focus on the former approach for the experiments.

IV. Experiments

To validate our approach we conduct a fair comparison between CE minimization, CRPS minimization, and Mean Absolute Error (MAE) minimization. Note that CE and CRPS are probabilistic and both predict the probability assigned to each one of a set of predefined bins. In contrast, the MAE is a deterministic loss, but it helps to draw a connection with the deterministic prediction literature.

A. Deterministic and Probabilistic Modes

It is well known that the MAE is minimized when the model predicts the median of the true distribution. Inspired by this fact, to use probabilistic algorithms as if they were deterministic ones, we obtain the median of the predicted distribution and use it as the point prediction. Note that it is not the aim of probabilistic methods to provide point predictions, but they can do it well if they learned to correctly predict the probabilities. We evaluate all methods as if they were deterministic in terms of the average test MAE.

Analogously, even though a model trained with the MAE is a deterministic one, we can turn it into a probabilistic one by setting a step function at the point prediction. This is similar to how the ground truth value is treated and allows us to draw a parallelism between the MAE and the CRPS: when the point prediction is used to define a step-like CDF, the CRPS is exactly the same as the MAE!

B. Data

To validate our approach we take a time-series of two years (2016 and 2017) of Global Horizontal Irradiance (GHI) measurements collected at the station of the Laboratorio de Energía Solar, located in Salto, Uruguay. These are high quality measurements that are accompanied by a clear sky model [13] that estimates the GHI under cloudless condition at any time. The clear sky model takes into account the position of the sun with respect to the station and the atmosphere state, leaving the clouds out. Dividing the GHI by the clear sky estimates yields the Clear Sky Index (CSI). We predict CSI, which can obviously be converted back to GHI. CSI is dimensionless, and has been processed so that nighttime is removed from the time series following previous work [1]. The time basis of the data is 10 minutes.

C. Data splits

We take the first half of the data to develop models, and the second half to test them. This is one year of test set comprising the four seasons. We ignore approximately two days of the beginning of the test data to ensure non-overlap. Correlation between points separated more than one day is negligible. From the first half of the data, we build a validation set by taking the first 5% of the points and an interval in the middle with 5% more points. These are summer and winter times, respectively. We consider the task of predicting an hour ahead with three hours of context. This implies that for a time series x indexed by t, we set $h = 6$ and $c = 18$ and build input-output pairs $([x_{t-c+1}, x_{t-c+2}, \ldots, x_t], x_{t+h})$. For the probabilistic methods, we use the training set to obtain $b_0 = 0$ and $b_B = \max_t (x_t)(1 + 1/|X|)$, where $|X|$ is the size of the train set.

D. Model

We use a multi-layer perceptron (MLP) with GeLU [14] activation function and a softmax layer at the end for the multi-output models. One hyperparameter is the model size, which is either small, medium or large. For each model size, we sweep the learning rate and the batch size in a grid defined at Table I. The optimizer is `AdamWScheduleFree` [15] with 5% of warmup and the number of steps 500. Finally, $B = 100$.

TABLE I: Hyperparameters considered for each model

Name	**Values**
(neurons, layers)	{(32, 1), (128, 3), (512, 5)}
batch size	{32, 64, 128}
learning rate	{1e-5, 4e-5, 1.6e-4, 6.4e-4, 2.56e-3, 1.024e-2}

E. Methodology

For each one of the three losses (MAE, CE, CRPS) we try all model hyperparameter options for the same number of iterations in the training set, while monitoring the validation loss. We save the weights that achieve the lowest validation loss for any point in the training and any hyperparameter set. This way we obtain for each loss function one model to be tested. The model weights are initialized identically for a given model size.

F. Results

What we expect is the MAE loss to be the best for deterministic predictions evaluated with the MAE and the CRPS loss to be at least as good as the CE loss for probabilistic predictions, both surpassing the MAE. Table II shows the results: for the probabilistic prediction the MAE loss is the worst, as the method is not probabilistic. The CE is better but is surpassed by the CRPS loss by a whopping 24%. Optimizing for the right thing can have a big impact in the quality of the predictions. When looking at the deterministic errors, the MAE loss is naturally better than the CE loss, but surprisingly it is surpassed by the CRPS loss. From this experiment, one could conclude that the CRPS ought to be always used for regression, even if it is deterministic. Further experiments are needed to confirm the generality of this statement. However, looking at literature from reinforcement learning, one can find that distributional approaches work better than deterministic ones [16], even when the interest is in a single value. None

TABLE II: Test set evaluation. Lower is better. Best results highlighted.

Loss name	MAE ↓	CRPS ↓
MAE	0.155	0.155
CE	0.171	0.128
CRPS (ours)	**0.130**	**0.0975**

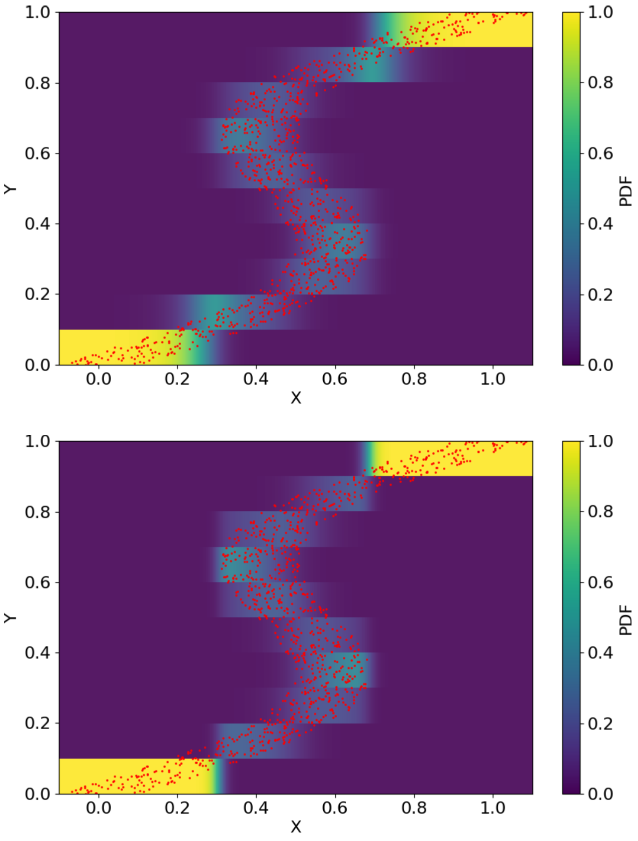

Fig. 1: Predicted PDF using CE Loss (top) and CRPS Loss (bottom) over Bishop synthetic data.

of these works optimize for the CRPS directly though, which opens an avenue for improvement.

For the solar forecasting task, we also run all methods with the same set of hyperparameters, namely a medium-sized network, learning rate of 2.56×10^{-3}, and a batch size of 128. Test CRPS results are 0.157 for MAE (slightly worsened), 0.097553 for CE (much improved), and 0.097469 for the CRPS loss (same). This shows that the main experiment was too sensitive to hyperparameter selection, and that the CRPS is only slightly better than using CE loss.

To further control and evaluate the performances in probabilistic regression we reproduce the multi-modal dataset of Bishop [17], composed of (x, y) pairs defined by $x = y + 0.3\sin(2\pi y) + \text{Uniform}[-0.1, 0.1]$ (see Figure 1). On this synthetic dataset, we observe that the CE loss generates softer predictions and it slightly surpasses the CRPS loss in test CRPS $(0.0999$ vs. $0.1001)$.

V. Conclusions

We have presented a new way of obtaining a deep probabilistic regressor, namely training with the CRPS loss. For histogram-like distributions, the CRPS loss is mathematically straightforward and directly optimizes a proper scoring rule, which is often the evaluation metric itself. This allows the CRPS to be better than cross-entropy, and even surpass the MAE loss for deterministic forecasts. We showcase the effectiveness of the new loss over a solar forecasting problem, obtaining improvements over the two baselines. We hope our work will encourage other researchers to explore probabilistic methods and to improve those we now have.

References

[1] P. Lauret, M. David, and H. T. C. Pedro, “Probabilistic solar forecasting using quantile regression models,” *Energies*, vol. 10, no. 10, 2017.

[2] T. Gneiting and M. Katzfuss, “Probabilistic forecasting,” *Annual Review of Statistics and Its Application*, vol. 1, no. 1, pp. 125–151, 2014.

[3] G. W. Brier, “Verification of forecasts expressed in terms of probability,” *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[4] F. Marchesoni-Acland, A. Herrera, F. Mozo, I. Camiruaga, A. Castro, and R. Alonso-Suárez, “Deep learning methods for intra-day cloudiness prediction using geostationary satellite images in a solar forecasting framework,” *Solar Energy*, vol. 262, p. 111820, 2023.

[5] D. S. Wilks, *Statistical methods in the atmospheric sciences*. Academic press, 2011.

[6] S. Nie, M. Zheng, and Q. Ji, “The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 101–111, 2018.

[7] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[8] R. Benedetti, “Scoring rules for forecast verification,” *Monthly Weather Review*, vol. 138, no. 1, pp. 203–211, 2010.

[9] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, pp. 1050–1059, PMLR, 2016.

[10] C. K. Sønderby et al., “Metnet: A neural weather model for precipitation forecasting,” *arXiv preprint arXiv:2003.12140*, 2020.

[11] C. M. Bishop, “Mixture density networks,” tech. rep., Aston Univ., 1994.

[12] J. E. Matheson and R. L. Winkler, “Scoring rules for continuous probability distributions,” *Management science*, vol. 22, no. 10, pp. 1087–1096, 1976.

[13] M. Lefèvre et al., “Mcclear: a new model estimating downwelling solar radiation at ground level in clear-sky conditions,” *Atmospheric Measurement Techniques*, vol. 6, pp. 2403–2418, 2013.

[14] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” *arXiv preprint arXiv:1606.08415*, 2016.

[15] A. Defazio, H. Mehta, K. Mishchenko, A. Khaled, A. Cutkosky, *et al.*, “The road less scheduled,” *arXiv preprint arXiv:2405.15682*, 2024.

[16] S. Ivanov and A. D’yakonov, “Modern deep reinforcement learning algorithms,” *arXiv preprint arXiv:1906.10025*, 2019.

[17] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.