Computer Vision - 2025

A Tiny Lecture. Where Do Loss Functions Come From?

Lectures by Alexei Kornaev [1,2,3]
Practical sessions by Kirill Yakovlev [2]

[1]AI Institute, Innopolis University (IU), Innopolis
[2]Robotics & CV Master's Program, IU, Innopolis
[3]RC for AI, National RC for Oncology, Moscow

February 4, 2025

Outline

Section 1. Core of an AI Model

Formalization

Given a dataset $\{x_i, y_i\}$, $i = 1, 2, ..., N$. Consider a model $\mathbf{f}$ which maps each i^{th} sample $x_i \in \mathbb{R}$ into the hypothesis (prediction) $h_i \in (0, 1)$ which in turn should be close to the label $y_i \in \{0, 1\}$.

1. The model inputs a sample x_i
2. And outputs a prediction h_i which should be close to y_i

To Train a Model means minimizing a loss function

Goals

The goals of this lecture are:

1. To demonstrate the grounds of loss functions in AI
2. To generalize the loss functions intuition

Section 2. Recap

Probability Mass Distribution

A **probability mass distribution** is a function $p(x_i)$ that satisfies the following two properties:

1. **Non-Negativity**: The probability mass function is non-negative for all possible values of x_i:

$$p(x_i) \geq 0 \quad \text{for all } x_i.$$

2. **Normalization**: The sum of the probabilities over all possible values of x is equal to one:

$$\sum_i p(x_i) = 1.$$

Product Rule for the joint probability

The **product rule** (or chain rule) for the joint probability of N variables $x_1, x_2, \dots, x_N$ is given by:

$$p(x_1, x_2, \dots, x_N) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \dots p(x_N \mid x_1, x_2, \dots, x_{N-1}).$$

If the variables $x_1, x_2, \dots, x_N$ are **independent and identically distributed (i.i.d.)**, the joint probability takes the form:

$$p(x_1, x_2, \dots, x_N) = p(x_1) \cdot p(x_2) \cdot p(x_3) \dots p(x_N).$$

Section 3. Loss Functions

Binary Cross-Entropy (BCE) Loss Intuition

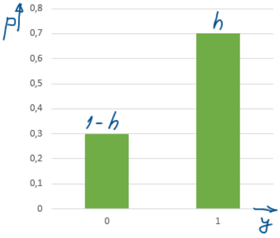

Figure: The Bernoulli distribution Bishop and Nasrabadi (2006); Prince (2023).

Derivation of BCE Loss from MLE

For a pair (x_i, y_i), the Bernoulli distribution takes the form:

$$p(y_i \mid h_i) = h_i^{y_i}(1-h_i)^{1-y_i}.$$

This represents the probability of observing y_i given the predicted probability h_i.
For a dataset of N i.i.d. pairs, the joint probability is:

$$P(y_1, y_2, \ldots, y_N \mid h_1, h_2, \ldots, h_N) = \prod_{i=1}^{N} h_i^{y_i}(1-h_i)^{1-y_i} \to \max.$$

Then we take the negative logarithm of the joint probability:

$$-\log P = -\sum_{i=1}^{N}\left[y_i \log h_i + (1-y_i)\log(1-h_i)\right] \to \min.$$

BCE

Binary Cross-Entropy (BCE) Loss Intuition

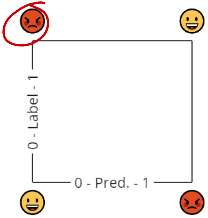

Figure: The BCE loss values intuition concerning the predictions and labels Prince (2023); Goodfellow et al. (2016).

Derivation of BCE Loss from MLE

The Binary Cross-Entropy (BCE) loss can be derived from Maximum Likelihood Estimation (MLE) for binary classification problems. Given a set of predictions $\hat{y}_i$ and true labels y_i, the BCE loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log h_i + \underbrace{(1-y_i)\log(1-h_i)}_{0}\right],$$

$-\infty$

let $h \to 0$, $y = 1$

$L_{BCE} \to \infty$

where:

- N is the number of samples,
- $y_i \in \{0, 1\}$ is the true label,
- $h_i \in (0, 1)$ is the predicted probability.

An Uncertainty Aware Binary Cross-Entropy (UBCE) Loss Intuition

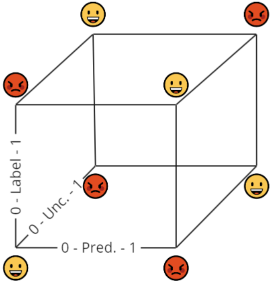

Figure: The UBCE loss values intuition concerning the predictions, the labels, and the uncertainties .

Derivation of BCE Loss from MLE

The Binary Cross-Entropy (BCE) loss can be derived from Maximum Likelihood Estimation (MLE) for binary classification problems. Given a set of predictions $\hat{y}_i$ and true labels y_i, the BCE loss is defined as:

$$\mathcal{L}_{\text{UBCE}} = -?,$$

where:

- N is the number of samples,
- $y_i \in \{0, 1\}$ is the true label,
- $h_i \in (0, 1)$ is the predicted probability,
- $u_i \in (0, 1)$ is the uncertainty of the prediction.

Section 4. Conclusion

Conclusion

Given a dataset $\{x_i, y_i\}$, $i = 1, 2, ..., N$. Consider a model $\mathbf{f}$ which maps each i^{th} sample $x_i \in \mathbb{R}$ into the hypothesis (prediction) $h_i \in (0, 1)$ which in turn should be close to the label $y_i \in \{0, 1\}$.

Recipe for constructing loss functions by Prince (2023)

1. Choose a suitable probability distribution defined over the domain of the predictions
2. Set the machine learning model to predict
3. To train the model, find the model parameters that minimize the negative log-likelihood loss function over the training dataset pairs
4. to perform inference for a new test sample, return either the full distribution or the value where this distribution is maximized.

Bibliography

Bishop, C. M. and Nasrabadi, N. M. (2006). Pattern recognition and machine learning, volume 4. Springer.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press. `http://www.deeplearningbook.org`.

Prince, S. J. (2023). Understanding Deep Learning. The MIT Press.