University of Tehran

Faculty of New Sciences and Technologies Department

of Mechatronics

Practice of Artificial Neural Networks

By:

Omid Moradi

Lecturer:

Dr. Masoudnia

Fall 2024

1) Compare Negative Log-Likelihood (or Cross-Entropy) Loss with Mean Squared Error (MSE) Loss, describing their advantages and limitations in comparison to each other.

When we train machine learning models, especially in classification or regression tasks, one of the critical decisions is which loss function to use. Two of the most widely used are Negative Log-Likelihood (NLL) Loss, often referred to as Cross-Entropy Loss, and Mean Squared Error (MSE) Loss. These functions serve distinct purposes, and each has its strengths and weaknesses depending on the problem we're tackling. Let's break down what makes them different and when we might choose one over the other.

- **Cross-Entropy Loss**:
  - This loss is a popular choice for **classification problems**—situations where we're trying to assign an input to one of several discrete categories (e.g., identifying an image as a cat, dog, or bird).
  - It measures the distance between the true class label (which we represent as a 1 or 0 for binary classification) and the predicted probability distribution that the model outputs.
  - Formula:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) \,.$$

- **Mean Squared Error (MSE) Loss**:
  - This is commonly used in **regression problems**, where the goal is to predict a continuous value, like house prices or temperature.
  - MSE tells us how far our predicted value is from the actual value, by squaring the difference and averaging it over all the data points.
  - Formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Cross-Entropy (CE) vs. Mean Squared Error (MSE)

ASPECT	CROSS-ENTROPY (CE) LOSS	MSE
TYPE OF PROBLEM	Best for classification tasks	Best for regression tasks
OUTPUT TYPE	Compares predicted probabilities (e.g., softmax output) with true class labels	Compares continuous predictions with actual values
BEHAVIOR ON ERRORS	Penalizes confident wrong predictions more heavily	Penalizes larger errors more heavily, but treats errors linearly
RANGE OF VALUES	Non-negative, bounded below by 0, unbounded from above	Non-negative, bounded below by 0, unbounded from above
INTERPRETABILITY	Intuitive in terms of measuring how far the predicted probabilities are from the actual distribution	Measures the magnitude of error but doesn't relate to probability distributions
GRADIENT BEHAVIOR	The gradient of CE is well-behaved for classification (particularly for logistic and softmax).	The gradient of MSE is proportional to the error, which can lead to smaller updates for small errors and larger updates for big errors.

Advantages of Cross-Entropy (CE) Loss

1. Penalizes Confident Wrong Predictions Heavily:

    - If a model assigns a high probability to the wrong class, cross-entropy loss penalizes it heavily. This is useful for training classifiers to avoid making confident wrong decisions.
    - Example: If y=1y = 1y=1 but the predicted probability p=0.01p = 0.01p=0.01, cross-entropy would penalize this severely.

- In contrast, MSE would not necessarily distinguish between confident wrong predictions and less confident wrong predictions as much.

2. Probabilistic Interpretation:

   - CE Loss provides a probabilistic interpretation of how “wrong” the model is. This makes it a natural fit for classification tasks where the model outputs probabilities.

3. Better for Class Probabilities:

   - For models outputting probabilities (like softmax in neural networks), CE Loss is more appropriate since it compares the entire predicted probability distribution against the true distribution.

4. Well-Behaved Gradients for Classification:

   - The gradient of CE with respect to model parameters is well-behaved, especially when training deep neural networks with softmax. It ensures smooth optimization, as the loss function is convex for logistic regression.

Advantages of Mean Squared Error (MSE) Loss

1. Simplicity and Interpretability:

   - MSE is simple and intuitive: it directly measures how far off predictions are from the actual values. This is suitable for regression tasks where the objective is to minimize the prediction error.

2. Smooth and Convex:

   - MSE loss is convex, which ensures that gradient-based optimization methods (like gradient descent) can converge to a global minimum for simple linear regression models.

3. Less Impact from Single Prediction:
    - Unlike CE, MSE doesn't disproportionately penalize individual wrong predictions, as it treats all errors equally in terms of their squared magnitude.

Limitations of Cross-Entropy Loss

1. Can be Sensitive to Outliers:
    - If a single sample is very hard to classify or misclassified with high confidence, it can lead to a large loss value. This might dominate the overall loss, potentially impacting learning.
2. Assumes Correct Probability Interpretation:
    - The model is expected to output meaningful probabilities (i.e., that sum to 1). If the model doesn’t produce proper probabilities, the loss may not behave as expected.

Limitations of MSE Loss

1. Not Ideal for Classification:
    - For classification tasks, using MSE loss with probability outputs (e.g., from logistic regression) can result in slower convergence and poorer performance.
    - MSE loss treats differences linearly, meaning a prediction of 0.1 for a true label of 0 (binary classification) is penalized similarly to a prediction of 0.9 for a true label of 1. Cross-entropy, on the other hand, would penalize the latter more heavily.
2. Sensitivity to Large Errors:
    - MSE penalizes large errors quadratically. This means that outliers (large errors) can disproportionately affect the loss, leading the model to focus excessively on reducing these large errors, potentially at the cost of overall performance.

3. Does Not Handle Probabilities Well:

    - MSE does not compare probability distributions effectively, meaning it’s less suited for classification tasks where model outputs are probabilities (e.g., softmax outputs).

2) What are the F1 Score, ROC Curve, and AUC measures? How and in which classification scenarios can they be employed?

These are important evaluation metrics in classification problems, particularly when dealing with imbalanced datasets or when understanding the trade-off between different types of errors (false positives and false negatives).

F1 Score

The F1 Score is a performance metric for classification models, specifically the harmonic mean of precision and recall. It is particularly useful in scenarios where the class distribution is imbalanced.

Precision: The ratio of true positive predictions to all positive predictions (i.e., how many of the predicted positives were correct).

$$Precision = True\ Positives\ /\ (True\ Positives + False\ Positives)$$

Recall (Sensitivity): The ratio of true positive predictions to all actual positives (i.e., how many of the actual positives were correctly predicted).

$$Recall = True\ Positives\ /\ (True\ Positives + False\ Negatives)$$

F1 Score Formula:

$$F1\ =\ 2\ *(Precision * Recall)\ /\ (Precision + Recall)$$

When to Use F1 Score:

1. **Imbalanced datasets**: F1 Score is particularly useful when the classes are imbalanced (e.g., in fraud detection, medical diagnostics), because it balances the trade-off between precision and recall.
2. **Binary Classification Problems**: It is often used in binary classification scenarios but can be extended to multi-class classification using techniques like micro-averaging or macro-averaging.

ROC Curve

The Receiver Operating Characteristic (ROC) Curve is a graphical representation that illustrates the performance of a classification model across all classification thresholds. It plots:

- the True Positive Rate (TPR)
- the False Positive Rate (FPR).

$$TPR = True\ Positives\ /\ (True\ Positives + False\ Negatives)$$

$$FPR = False\ Positives\ /\ (False\ Positives + True\ Negatives)$$

When to Use ROC Curve:

- **Evaluating Model Performance**: The ROC curve shows how well the model can distinguish between the positive and negative classes across all possible classification thresholds.

- **Comparing Models**: It’s useful to compare different models. The closer the curve is to the top-left corner, the better the model is performing.

Key Insights from ROC Curve:

- A perfect classifier would have a point in the top-left corner (FPR = 0, TPR = 1), corresponding to no false positives and maximum true positives.

- A **random classifier** would plot a diagonal line from (0,0) to (1,1), indicating random guessing.

AUC (Area Under the ROC Curve)

AUC stands for **Area Under the Curve**, and in this case, it refers to the area under the ROC curve. The AUC summarizes the performance of the classifier across all thresholds.

- **AUC Values** range between 0 and 1:
- **AUC = 1**: Perfect classifier.
- **AUC = 0.5**: Random classifier (i.e., no predictive value).
- **AUC < 0.5**: Worse than random (i.e., making predictions opposite to the true class).

When to Use AUC:

- **Evaluating Model Robustness**: AUC provides a single number that represents the performance of the classifier across all classification thresholds, making it a useful summary metric.
- **Imbalanced Datasets**: AUC is relatively insensitive to class imbalance since it measures the trade-off between true positive and false positive rates.

Example Scenarios

1. **Medical Diagnosis** (Binary Classification, Imbalanced Classes):

    - **F1 Score**: Important because both false positives and false negatives have a cost. For instance, missing a positive cancer case (false negative) is harmful, but overpredicting cancer cases (false positive) can lead to unnecessary stress and tests.

- **ROC Curve & AUC**: Useful to choose an optimal threshold to balance the tradeoff between false positives and false negatives.

2. **Spam Detection** (Binary Classification, Slightly Imbalanced):

   - **F1 Score**: Helps balance the number of correctly identified spam (true positives) against mistakenly flagged non-spam emails (false positives).

   - **AUC**: Helps evaluate how robust the classifier is across different thresholds, giving a good sense of overall performance.

3. **Multi-Class Image Classification**:

   - **ROC Curve & AUC (per class)**: Useful to evaluate how well the model distinguishes between different image classes (e.g., cats, dogs, birds). A per-class AUC can be computed, followed by averaging for an overall AUC score.

   - **F1 Score**: Can be used in a **macro-average** form (to treat each class equally) or **micro-average** form (to account for class imbalances).