CROSS-ENTROPY LOSS

NISER,BHUBANESWAR, ODISHA

Ankur Abhijeet
Summit Bikram Nayak

NISER,Bhubaneswar, Odisha

April 29, 2023

PART I: FAMILIARISATION

1 Introduction 4

2 Formulae 5

3 Algorithm 6

4 Example 7

PART II: EXPLANATION

1 Cross Entropy in Binary Classification 12

2 Cross Entropy in Multiclass Classification 13

3 Cross Entropy in Neural Networks 14

4 Applications of Cross Entropy 15

Part I

FAMILIARISATION

Introduction

Meaning of CROSS ENTROPY LOSS:

- Cross entropy loss is a measurement of the dissimilarity between two probability distributions.
- In machine learning, it is commonly used to compare the predicted probability distribution of a model with the true probability distribution of the training data.
- By minimizing the cross entropy loss during training, the model can learn to make more accurate predictions.

What is it?

- Cross entropy loss is a commonly used loss function in machine learning, particularly for classification problems.
- It measures the difference between two probability distributions: the predicted probability distribution and the true probability distribution.
- The goal is to minimize the cross entropy loss during the training process, which helps the model make accurate predictions.

Formulae

The formula for cross entropy loss is:
$H(y, \hat{y}) = -\sum_{i=1}^{n} y_i log(\hat{y}_i)$
where:

- y is the true probability distribution (i.e., the one-hot encoded true labels)
- $\hat{y}$ is the predicted probability distribution (i.e., the model's predicted probabilities for each class)
- n is the number of classes in the classification problem
- log is the natural logarithm function

This formula calculates the sum of the products of the true probability distribution and the log of the predicted probability distribution for each class, multiplied by -1. The resulting value represents the dissimilarity or distance between the two probability distributions. The goal of the training process is to minimize this value to improve the accuracy of the model's predictions.

Algorithm

Inputs:

- True labels y (one-hot encoded)
- Predicted probabilities $\hat{y}$ for each class

Algorithm:

- Initialize the cross entropy loss value to zero: $H(y, \hat{y}) = 0$
- For each class i from 1 to n (where n is the number of classes):
  - Calculate the product of the true label for class i and the log of the predicted probability for class i: $y_i * log(\hat{y}_i)$ the result to the cross entropy loss value: $H(y, \hat{y}) = H(y, \hat{y}) + (-y_i * log(\hat{y}_i))$
- Return the final cross entropy loss value: $H(y, \hat{y})$

This algorithm calculates the cross entropy loss between the true labels and predicted probabilities for a multi-class classification problem. The goal of the training process is to minimize this value using optimization techniques such as gradient descent.

Calculating Cross Entropy Loss for a Simple Classification Model

Considering a simple example:

Suppose you have a binary classification problem where you are trying to predict whether an image contains a cat or not. You have a dataset of 100 images, where 50 images contain cats and 50 images do not. You train a simple logistic regression model to make predictions on this dataset.
After training, the model predicts the probability of an image containing a cat, which can be either 0 or 1. To calculate cross entropy loss for this model, you need to compare the predicted probability with the true label for each image.

Calculating Cross Entropy Loss for a Simple Classification Model

Calculation for One Image

Suppose you have an image that contains a cat, so the true label is y = 1. The model predicts a probability of $\hat{y}$ = 0.8 for this image. You can represent the true label y as a probability distribution [0, 1], where the first element represents the probability of the image not containing a cat and the second element represents the probability of the image containing a cat.
The cross entropy loss for this image is calculated using the following formula:
$H(y, \hat{y}) = -\sum_{i=1}^{2} y_i log(\hat{y}_i) = -(0 * log(0.2) + 1 * log(0.8)) = 0.2231$

Calculating Cross Entropy Loss for a Simple Classification Model

Calculation for the Entire Dataset

To calculate the overall cross entropy loss for the entire dataset, you would repeat this process for each image and then take the average of the cross entropy loss values. The goal of the training process is to minimize this value, which helps the model make more accurate predictions on the classification problem.

Calculating Cross Entropy Loss for a Simple Classification Model

Appendix:

- The curve is steeper near the boundaries: The curve becomes steeper as the predicted probability moves closer to 0 or 1. This means that the model is penalized more harshly for making incorrect predictions when it is more certain about its prediction.
- The minimum value is at the true label value: The curve reaches its minimum value when the predicted probability is equal to the true label value (i.e., $y_pred_cat = y_true$). This means that the model is rewarded for making accurate predictions, and that the cross entropy loss is minimized when the model is confident in its prediction.
- The curve is bounded between 0 and infinity: The curve is bounded between 0 and infinity, and it approaches 0 as the predicted probability approaches the true label value. This means that the cross entropy loss can never be negative, and that it is minimized when the predicted probability is equal to the true label value.

Part II

EXPLANATION

Cross Entropy in Binary Classification

- Binary classification problem: predicting one of two classes
- Cross entropy loss measures the difference between predicted probabilities and true labels
- Formula for binary cross entropy loss:

$$L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

  where $y \in \{0, 1\}$ is the true label and $\hat{y}$ is the predicted probability of the positive class
- Intuition behind the formula: penalizes incorrect predictions more harshly than correct ones

Cross Entropy in Multiclass Classification

- Multiclass classification problem: predicting one of multiple classes
- Cross entropy loss measures the difference between predicted class probabilities and true labels
- Formula for multiclass cross entropy loss:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log \hat{y}_i$$

  where $\mathbf{y}$ is a one-hot encoded vector of true labels, $\hat{\mathbf{y}}$ is a vector of predicted class probabilities, and C is the number of classes
- Intuition behind the formula: penalizes incorrect predictions more harshly than correct ones, and encourages the model to assign high probabilities to the correct classes

Cross Entropy in Neural Networks

- Cross entropy loss is commonly used in training neural networks
- In backpropagation, the cross entropy loss gradient is used to update the model weights
- Variants of cross entropy loss include sparse categorical cross entropy and binary cross entropy with logits
- Choosing the appropriate cross entropy loss depends on the problem at hand and the output format of the model

Applications of Cross Entropy

- Cross entropy loss is used in a wide range of machine learning applications, including image classification, natural language processing, and recommender systems
- In image classification, cross entropy loss is commonly used in convolutional neural networks to predict the class of an image
- In natural language processing, cross entropy loss is used to predict the next word in a sequence or to classify text into different categories
- In recommender systems, cross entropy loss is used to predict the likelihood that a user will interact with a particular item or recommendation

Part III

PSEUDO-CODE

Binary Cross Entropy Loss Pseudocode

INPUT: cost function J(theta), number of iterations N, learning rate alpha, training examples X and labels y
INITIALIZE: random theta
FOR i = 1 to N DO
FOR each training example Xj and corresponding label yj DO Compute the predicted probability distribution over classes:
z = Xj * theta
$y_pred = softmax(z)$
Compute the cross entropy loss for the predicted and true label:
loss = - sum(yj * log(y_pred))
Compute the gradient of J with respect to theta:
gradient = Xj.T * ($y_pred - yj$)
Update the parameters theta:
theta = theta - alpha * gradient
OUTPUT: theta

- y is the true binary label (0 or 1)
- $\hat{y}$ is the predicted probability (between 0 and 1)
- The loss is calculated as the negative log likelihood of the predicted probability given the true label

Categorical Cross Entropy Loss Pseudocode

INPUT: predicted probabilities P and ground truth probabilities Q for a categorical variable with k classes
FOR i = 1 to k DO
Compute the cross entropy for each class:
entropy$_i = -Q_i * log(P_i)$
Compute the total cross entropy loss:
cross$_e ntropy_l oss = (entropy_i)$
OUTPUT: cross$_e ntropy_l oss$

Sparse Categorical Cross Entropy Loss Pseudo-code

INPUT: predicted probabilities matrix
$Y_predofshape(batch_size, num_classes), truelabelsvectory_trueofshape(batch_size,)$
INITIALIZE: loss = 0.0
FOR i = 1 to $batch_sizeDO$
SELECT the ith row from $Y_pred, givingavectorofpredictedclassprobabilities$
SELECT the ith element from
$y_true, givingthetrueclasslabelforthisexampleCOMPUTEthecrossentropylossforthisexample$:
$example_loss = -log(Y_pred[i, y_true[i]])$
ADD the example loss to the total loss:
loss = loss + $example_loss$
COMPUTE the mean loss across all examples:
loss = loss / $batch_size$
OUTPUT: loss

Advantages of Cross Entropy Loss

- Cross entropy loss is a widely used loss function in machine learning and deep learning for classification problems.
- It is well-suited for models that output a probability distribution over multiple classes.
- Cross entropy loss penalizes the model more heavily for predictions that are far from the true class label, compared to predictions that are close but not exactly correct.
- This property makes it particularly effective in problems where the classes are imbalanced or where misclassification of certain classes is more important than others.
- Cross entropy loss is also differentiable, which allows for gradient-based optimization methods such as stochastic gradient descent to be used to train the model.

Disadvantages of Cross Entropy Loss

- The Cross Entropy Loss assumes independence between classes, which may not always hold in practice.
- It can be sensitive to class imbalance, where the minority class(es) may be assigned more weight than the majority class(es).
- It can be computationally expensive for large numbers of classes, as it requires computing the softmax function for all classes.

References I

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
Bishop, Christopher M. Pattern recognition and machine learning. Springer, 2006.
Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).