Binary cross entropy is a cost function used in machine learning, particularly in binary classification tasks. It measures the difference between the predicted probability distribution of the output and the true probability distribution of the output.

The formula for binary cross entropy is as follows:

$J(y, \hat{y}) = - 1/m * \sum [y * \log(\hat{y}) + (1-y) * \log(1-\hat{y})]$

where:

J is the binary cross entropy cost function
y is the true label (either 0 or 1)
$\hat{y}$ is the predicted probability of the output being 1
m is the number of training examples
The first term in the sum ($y * \log(\hat{y})$) penalizes the model heavily if it predicts 0 and the true label is 1. The second term in the sum ($(1-y) * \log(1-\hat{y})$) penalizes the model heavily if it predicts 1 and the true label is 0. The overall cost is then the average of the sum of these penalties across all training examples.

The binary cross entropy cost function is commonly used in logistic regression and other models where the output is a probability value between 0 and 1. The goal is to minimize the cost function by adjusting the model parameters (e.g., weights and biases) to improve the predictions.