Loss Functions in ANN

The loss function is a critical part of model training: it quantifies how well a model is performing a task by calculating a single number, the **loss**, from the model output and the desired target.

Choosing a loss function

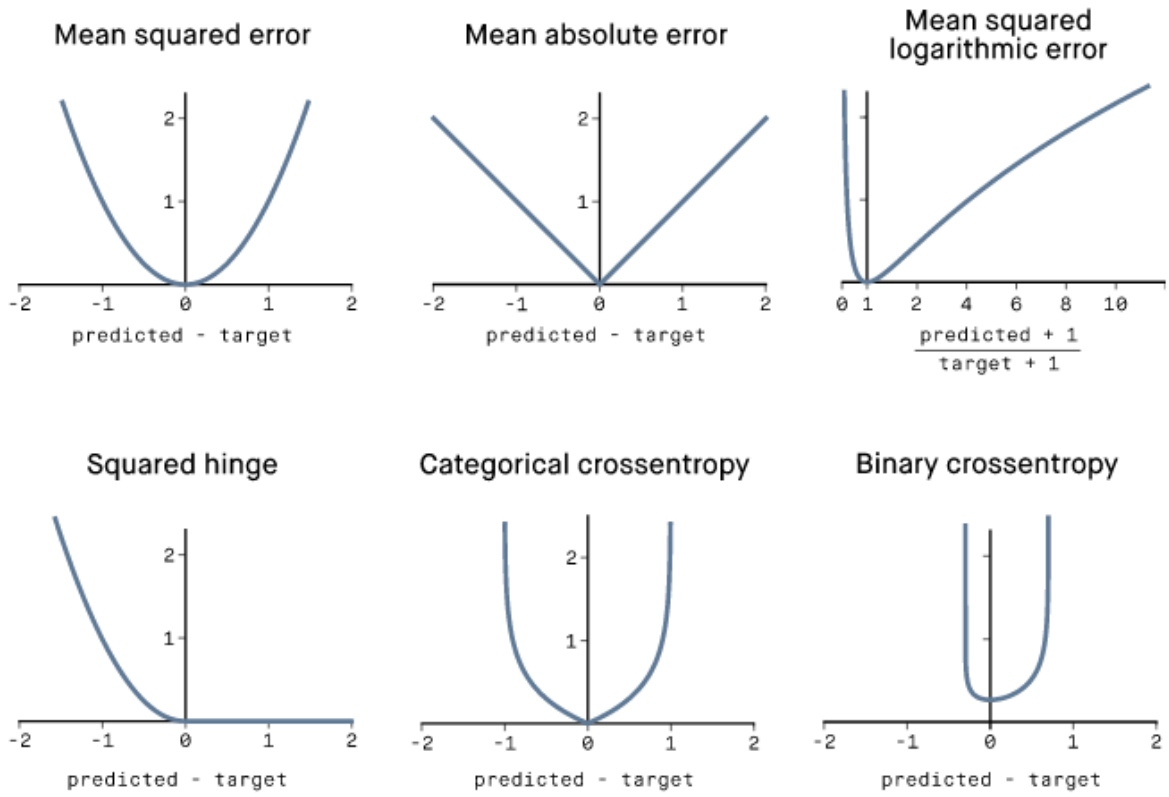


Figure 1. Plots of various loss functions as a function of the relation between predicted and target values. The loss is closest to 0 when the predicted value equals the target.

The following loss functions are available on the platform:

Classification

Single label:
Categorical crossentropy

Regression

Continuous values:
Mean squared error
Mean

Classification

Regression

absolute error
Mean squared logarithmic error

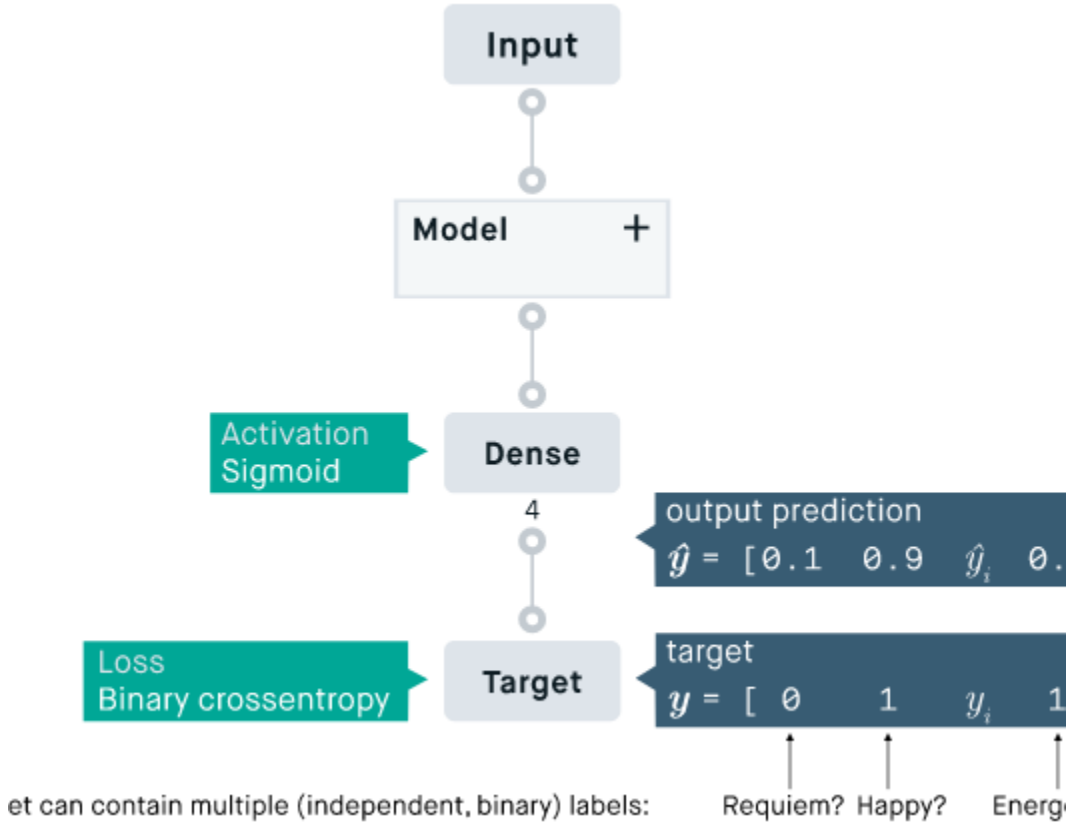
Multi-label:
Binary crossentropy
Squared hinge

Example

- The TanH activation outputs values between -1 and 1, which makes it incompatible with the categorical crossentropy.
- The sigmoid activation outputs values between 0 and 1, which makes it a perfect match for the categorical crossentropy!

1. **Binary Cross entropy**
   Binary crossentropy is a loss function that is used in binary classification tasks. These are tasks that answer a question with only two choices (yes or no, A or B, 0 or 1, left or right). Several independent such questions can be answered at the same time, as in multi-label classification or in binary image segmentation. Formally, this loss is equal to the average of the categorical crossentropy loss on many two-category tasks.

Classification **Regression**



Binary crossentropy math

The binary crossentropy loss function calculates the loss of an example by computing the following average:

$$\text{Loss} = -\frac{1}{\underset{\text{size}}{\text{output}}} \sum_{i=1}^{\underset{\text{size}}{\text{output}}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

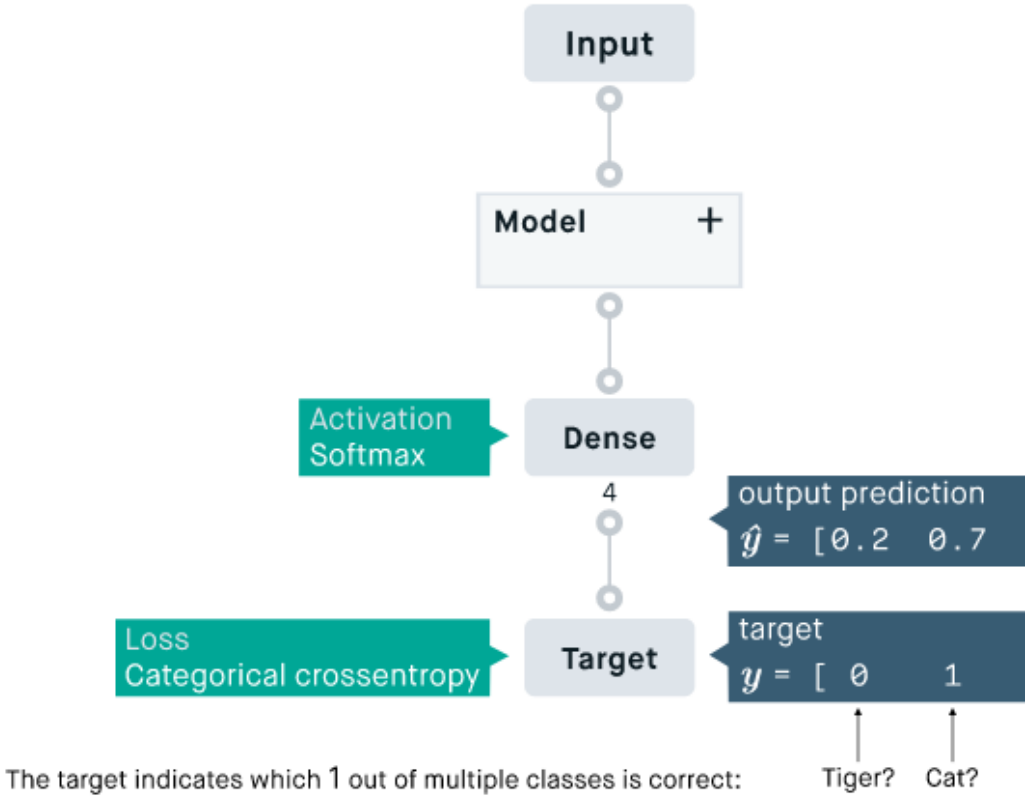
where y^i\hat{y}_iy^i is the iii-th scalar value in the model output, yiy_iyi is the corresponding target value, and `output size` is the number of scalar values in the model output.

Sigmoid Activation function is used.

Classification **Regression**

2. **Categorical cross entropy function**

   Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.



Categorical crossentropy math

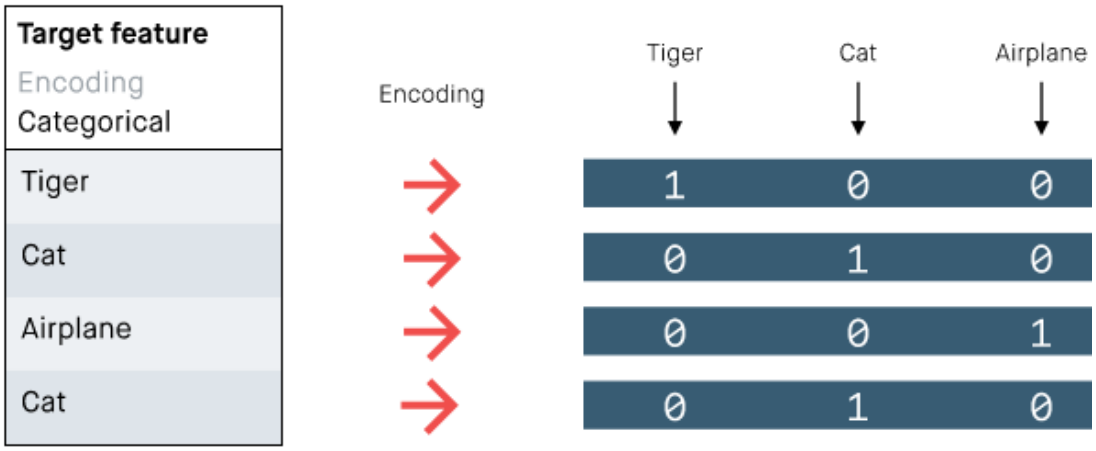
The categorical crossentropy loss function calculates the loss of an example by computing the following sum:

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

Classification **Regression**

Soft max activation function is used.

One hot encoding for the target attribute values is used.



3. **Mean absolute error**
   Used in regression. The loss is the mean overseen data of the absolute differences between true and predicted values, or writing it a formula:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} |y - \hat{y}_i|$$

Use Mean absolute error when you are doing regression and don't want outliers to play a big role. **Example**: When doing image reconstruction, MAE encourages less blurry images compared to MSE.

4. **Mean squared logarithmic error**

   Mean squared logarithmic error (MSLE) can be interpreted as a measure of the ratio between the true and predicted values.

   Mean squared logarithmic error is, as the name suggests, a variation of the Mean Squared Error.

   MSLE only care about the percentual difference.

MSLE penalizes underestimates more than overestimates.

MSLE math

The loss is the mean over the seen data of the squared differences between the log-transformed true and predicted values, or writing it as a formula:

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2$$

This loss can be interpreted as a measure of the ratio between the true and predicted values, since:

$$\log(y_i + 1) - \log(\hat{y}_i + 1) = log\left(\frac{y_i + 1}{\hat{y}_i + 1}\right)$$

Use MSLE when doing regression, believing that your target, conditioned on the input, is normally distributed, and you don't want large errors to be significantly more penalized than small ones, in those cases where the range of the target value is large.

Example: You want to predict future house prices, and your dataset includes homes that are orders of magnitude different in price. The price is a continuous value, and therefore, we want to do regression. MSLE can here be used as the loss function.

5. Mean Squared Error

Mean squared error (MSE) is the most commonly used loss function for regression. The loss is the mean overseen data of the squared differences between true and predicted values, or writing it as a formula.

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(y - \hat{y}_i)^2$$

When to use mean squared error

Use MSE when doing regression, believing that your target, conditioned on the input, is normally distributed, and want large errors to be significantly (quadratically) more penalized than small ones.

Example: You want to predict future house prices. The price is a continuous value, and therefore we want to do regression. MSE can here be used as the loss function.

6. Poison Loss Function

The poisson loss function is used for regression when modeling count data. Use for data follows the poisson distribution. Ex: churn of customers next week. The loss takes the form of:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} (\hat{y}_i - y_i log \hat{y}_i)$$

Use the Poisson loss when you believe that the target value comes from a Poisson distribution and want to model the rate parameter conditioned on some input. Examples of this are the number of customers that will enter a store on a given day, the number of emails that will arrive within the next hour, or how many customers that will churn next week.

7. Squared Hinge

The squared hinge loss is a loss function used for "maximum margin" binary classification problems. Mathematically it is defined as:

$$L(y, \hat{y}) = \sum_{i=0}^{N} \left(max(0, 1 - y_i \cdot \hat{y}_i)^2 \right)$$

Y is either -1 or 1. Thus, the squared hinge loss is:

0	when the true and predicted labels are the same and * when $\hat{y} \geq 1$ (which is an indication that the classifier is sure that it's the correct label)
quadratically increasing with the error	* when the true and predicted labels are not the same or * when $\hat{y} < 1$, even when the true and predicted labels are the same (which is an indication that the classifier is not sure that it's the correct label)

Use the Squared Hinge loss function on problems involving yes/no (binary) decisions and when you're not interested in knowing how certain the classifier is about the classification (i.e., when you don't care about the classification probabilities). Use in combination with the tanh() activation function in the last layer.

Example: You want to classify email into 'spam' and 'not spam' and you're only interested in the classification accuracy.

8. Focal Loss

Use the focal loss function in single-label classification tasks as an alternative to the more commonly used **categorical crossentropy**.

You can say that the focal loss is an extension to categorical crossentropy with an added weighting factor $(1-\hat{y})^{\gamma}$

$$\text{Loss} = \sum_{i=1}^{\text{output size}} (1 - \hat{y}_i)^{\gamma} \cdot y_i \cdot \log \hat{y}_i$$

Where:

- $(1 - \hat{y}_i)^{\gamma}$ is the weighting factor
- $\hat{y}$ is the predicted value.

The focusing parameter γ smoothly adjusts the rate at which easy examples are down-weighted.

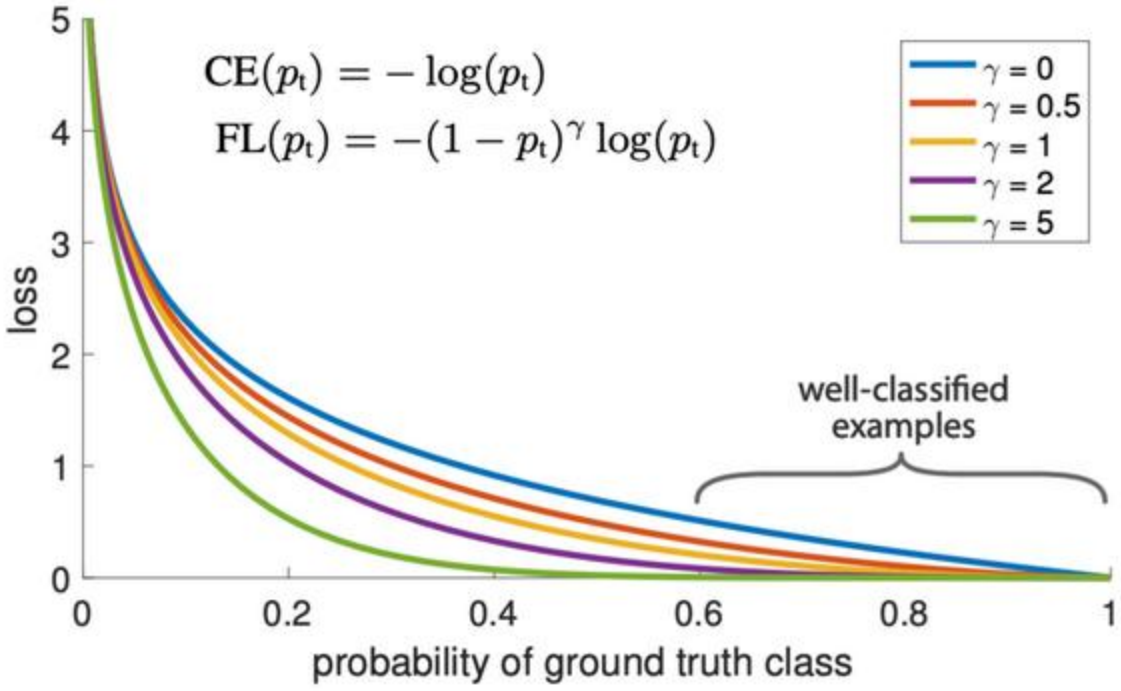


Figure 1. Illustration from the paper *Focal Loss for Dense Object*

The focal loss is a good alternative to categorical crossentropy for single-label classification tasks. Particularly for problems where:

- You have an **unbalanced dataset**
- The **distinction between classes is not clear** in the first place.

Example:
It is hard to define the genre of a song distinctly. Genre is quite subjective, and some songs might include elements from multiple genres. In this use case, it's a good idea to use the focal loss function.

Softmax is the only activation recommended to use with the focal loss function.
