Outline

Introduction

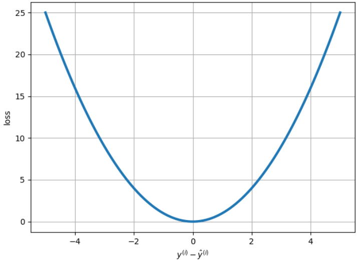
So far, we have used *mean square error* (mse) only.

There are different loss functions that better suit different tasks.

Mean square error (mse)

$$l_{mse} = \frac{1}{M}\sum_{m=i}^{M}\left(y^{(i)} - \hat{y}^{(i)}\right)^2,$$

where, M indicates the number of training samples in a batch.

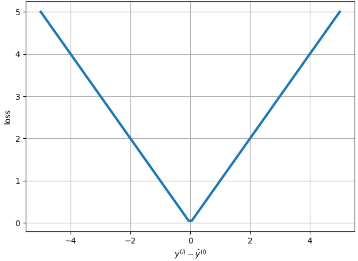


- a.k.a., $L2$ loss.
- Good for **regression** tasks.
- Trivial derivative for gradient descent.

Mean absolute error (mae)

$$l_{mae} = \frac{1}{M}\sum_{m=i}^{M} |y^{(i)} - \hat{y}^{(i)}|,$$

where, M indicates the number of training samples in a batch.

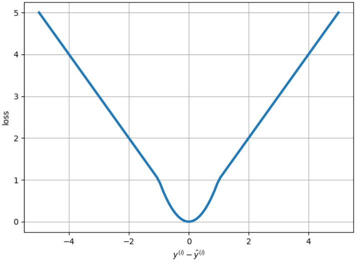


- a.k.a., $L1$ loss.
- More robust to outliers than mse.
- Good for **regression** tasks.
- Discontinuity in its derivative.

Pseudo-Huber loss

$$l_{PH} = \begin{cases} \frac{1}{2}\left(y - \hat{y}\right)^2, & |y - \hat{y}| < \delta, \\ \delta|y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases}$$

for a single training sample.

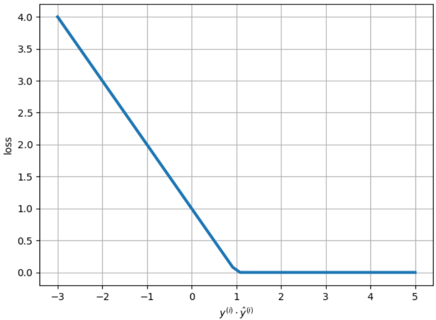


- Quadratic for small errors, and linear for large errors.
- Less sensitive to outliers than mse.
- Good for **regression** tasks.

Hinge loss

$$l_H^{(i)} = \max(0, 1 - y^{(i)} \cdot \hat{y}^{(i)}),$$

for a single training sample.



- Also used in SVM's.
- Consider y and $\hat{y}$ to be probabilities.
- Penalizes errors, but also correct predictions of low confidence.
- Good for binary **classification** tasks.

(Information theory I, Information)

C. Shannon: 1948 “A Mathematical Theory of Communication”.

For a random variable, taking N possible values with equal probability, we need $\log_2(N)$ bits to transmit its information.

For a random variable, taking N possible values with varying probabilities p_i, we obtain $-\sum_i p_i \log_2(p_i)$ bits of information, on average.

(Information theory II, Entropy)

"How uncertain events are".

$$H(p) = -\sum_i p_i \log_2(p_i).$$

- Average amount of information obtained from one sample drawn from a given probability distribution $\mathbf{p}$.
- How unpredictable that probability distribution is.

The more variation, the higher the entropy.

(Information theory III, Cross entropy)

Cross entropy $H(p, q)$ is a function of two probability distributions $\mathbf{p}$ and $\mathbf{q}$,

$$H(p, q) = -\sum_i p_i \log_2(q_i).$$

Provides the average message length when we encode $\mathbf{p}$ into $\mathbf{q}$.

If prediction is correct, then $H(p) = H(p, q)$.

Categorical cross entropy

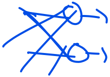
Clasificación multi-class,-
multi-instance
[0, 0, 1, 0, 1, 1]
activación: sigmoide
pérdida: binary cross-entropy

$$l_{CCE} = -\sum_i y_i \log_2(\hat{y}_i),$$

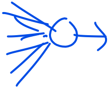
donde $\sum y_i = 1$, y $\sum \hat{y}_i = 1$.

- Notice subindices represent elements of a vector.
- Values between 0 and 1.
- Good for **multi-class classification** problems.
- Consider y to be a one-hot encoding vector, e.g., $[0, 0, 0, 1, 0]$ represents a label for the 4-th class.
- Prediction $\hat{y}$ might look like $[0.01, 0.01, 0.03, 0.93, 0.02]$.

Binary cross entropy



vs



2 neuronas
con softmax y —
categorical cross entrop.

1 neurona
sigmoide —
binary cross entr.

Special case of cross entropy for only two classes.

$$l_{BCE} = -\left(y \log_2(\hat{y}) + (1-y)\log_2(1-\hat{y})\right).$$

- Values between 0 and 1.
- Good for **binary classification** problems.

Kullback-Leibler divergence (D_{KL})

$$l_{D_{KL}} = \sum_i y_i \log_2 \frac{y_i}{\hat{y}_i}.$$

- $D_{KL}(p||q) = H(p,q) - H(p)$.
- Equivalent to categorical cross entropy up to a scale factor.
- Gives a notion of "the difference between the expected and predicted length of a message".
- Good for **classification** problems.

Adaptive

Few attempts have been made on getting adaptive loss functions.

Barron, 2019. “A General and Adaptive Robust Loss Function”.

Common practices

- For regression problems, try mse and then mae.
- For binary classification, try *binary cross entropy*.
- For multi-class classification, try *categorical cross entropy*.