1. Theoretical Tasks

1.1 Loss Functions

The following formulas are useful for doing the exercise, where *n* denotes the length of both the prediction vector *y* and the ground truth vector *g*.

Cross-Entropy Loss (or Logistic Loss)

$$H(y, g) = -\sum_{i}^{n} g_i \log(y_i)$$

Mean Squared Error-Loss

$$MSE(y, g) = \frac{1}{n}\sum_{i}^{n} (y_i - g_i)^2$$

Hinge Loss (or SVM Loss)

$$SVM(y, j) = \sum_{i \neq j}^{n} \max(0, y_i - y_j + 1)$$

where *j* is the index of the correct label for the sample.

Task

Consider the following two vectors:

$$g = [0, 1, 0]$$

$$y = [0.25, 0.6, 0.15]$$

Using the formula defined in the previous section, calculate the values:

- Cross-Entropy Loss
- Mean Squared Error Loss
- Hinge Loss

between the two

vectors.

Cross-Entropy Loss

(H(y, g)):

H(y, g) = -∑(gi * log(yi))

H(y, g) = -((0 * log(0.25)) + (1 * log(0.6)) + (0 * log(0.15)))

H(y, g) = -(0 - log(0.6) + 0) ≈ 0.5108

Mean Squared Error Loss (MSE(y, g)):

MSE(y, g) = (1/n) * ∑((yi - gi)^2)

MSE(y, g) = (1/3) * (((0.25 - 0)^2) + ((0.6 - 1)^2) + ((0.15 - 0)^2))

MSE(y, g) = (1/3) * ((0.0625) + (0.16) + (0.0225)) ≈ 0.0817

Hinge Loss (SVM(y, j)):

SVM(y, j) = ∑(max(0, yi - yj + 1) for i ≠ j)

SVM(y, j) = max(0, y0 - y1 + 1) + max(0, y2 - y1 + 1)

SVM(y, j) = max(0, 0.25 - 0.6 + 1) + max(0, 0.15 - 0.6 + 1)

SVM(y, j) = max(0, 0.65) + max(0, 0.55) ≈ 1.2

So, the calculated values
are:

• **Cross-Entropy
Loss: ≈ 0.5108**

• **Mean Squared
Error Loss: ≈ 0.0817**

• **Hinge Loss: ≈ 1.2**

Resources:

- What’s an intuitive way to think of cross entropy?
- Section 3.13 from the Deep Learning Book
- Notes from CS231n

(Optional) Evaluation Metrics

Measuring the performance of a system is a common yet very important task. Of particular interest is doing it in a way such that the results of two different system are comparable. For this reason there are metrics which are computed in a very specific way, s.t. their values for two systems can be compared directly. However, in practice, things are slightly more complicated.

Task A: Theoretical Foundations

Typically people refer to accuracy as THE evaluation metric, but there are a lot of evaluation metrics which can be better suited than accuracy depending on the task/dataset.

1. In which situation using accuracy is not necessarily a good idea?
2. What part of the formula for computing the accuracy makes it less desirable than the Jaccard Index (Intersection Over Union) in a multi-class setting ?
3. What is the difference between Jaccard Index (Intersection Over Union) and F1-Measure? Which one is more suited to measure performance of NNs ?

Task B: Practice

In this part of the exercise we want to compute some common alternatives which can be used instead of accuracy. We'll take an example from a real case scenario of layout analysis at pixels level of historical documents.

Given the following prediction and ground truth (note: this is a multi-class and multi-label scenario!), where **B** stays for background, **T** for text, **D** for decoration and **C** for comment.

	1	2	3	4	5	6	7	8
GT	B	B	B	B	TD	TD	TD	TD
P	B	T	TD	BD	BC	TC	T	TD

Compute the class frequencies and the following metrics per class:

- Jaccard Index
- Precision
- Recall
- F1-measure

Then compute their mean in two different ways: once with class balance (sum of per class values divided by number of classes) and once with the class frequencies (per class values times the class frequency). Finally, compute the global Exact Match metric.

Show the intermediate steps.

1. Intuitively the Exact Match would be the strictest metric possible. However, it might not be the lowest number: how come?
2. Can you compute the global accuracy on this example? Justify your answer.
3. What is the main issue going from multi-class to multi-label setting?

Resources

***Jaccard Index (or Intersection over Union*:**

https://en.wikipedia.org/wiki/Jaccard_index

***Exact Match (and others metrics)*:**

https://en.wikipedia.org/wiki/Multi-label_classification

***Precision and Recall*:**

https://en.wikipedia.org/wiki/Precision_and_recall