ID2223 - Scalable Machine Learning and Deep Learning

- Review Questions 2 -

PAOLO TETA RALFS ZANGIS

teta | zangis @kth.se

November 20, 2021

1 Question 1

(a) T

(b) F

(c) T

(d) F

2 Question 2

In Spark the *featureSubsetStrategy* parameter in ensemble model estimators, such as Random Forest, stands for the number of features to use as candidates for splitting at each tree node. It's set as a fraction of the total number of features. The possible values are: *auto*, *all*, *onethird*, *sqrt*, *log2* and *n*.

3 Question 3

The entropy of a dataset D is computed as follows:

$$H(D) = -\sum_{i=1}^{m} p_i \cdot \log_2(p_i)$$

where p_i is the probability that an instance in D belongs to class i, with m distinct classes. Generally, the entropy becomes zero when all class partitions are pure, which means that it contains only instances of one class. It happens when a node contains just one class, so $p_i = 1$ at that node, thus the entropy is zero.

4 Question 4

The Gini impurity of a dataset D is computed as follows:

$$Gini(D) = \sum_{i=1}^{m} p_i \cdot (1 - p_i) = 1 - \sum_{i=1}^{m} p_i^2$$

where p_i is the probability that an instance in D belongs to class i, with m distinct classes. It becomes zero when all class partitions are pure, which means that $p_i = 1$. Given this, the explanation is straight forward: if all class partitions are pure and $p_i = 1$, then we have $1 - 1 = 0$, thus the Gini impurity is zero.

5 Question 5

Considering a feedforward neural network with only one hidden layer, which has linear functions for both the activation function in the hidden layer and the output function. Given this, the whole neural network is model as a single layer and can be considered as a simple linear algebraic transformation. So, the linear map takes as input the domain space and linearly transforms it into another one. From the linear algebra theory, functions are linear and remain so if we apply only linear transformations, such as rotation, reflection, scaling and projection on one axis.

6 Question 6

Using a *step* function as an activation function in deep feedforward networks we're not allowed to compute a continuous output. In this case, activation functions must be not linear and differentiable in all the points of their domain, thus the *step* cannot guarantee this requirement. Moreover, it's not differentiable and the first derivative is 0 when it's constant and goes to infinity when the step occurs. So, in the backpropagation algorithm the idea is to replace the *step* function with other activation functions listed below.

- Logistic function (sigmoid) $\Longrightarrow \sigma(z) = \dfrac{1}{1+e^{-z}}$
- Hyperbolic tangent function $\Longrightarrow \tanh z = 2\sigma(2z) - 1$
- Rectified linear units (ReLUs) $\Longrightarrow ReLU(z) = \max(0, z)$

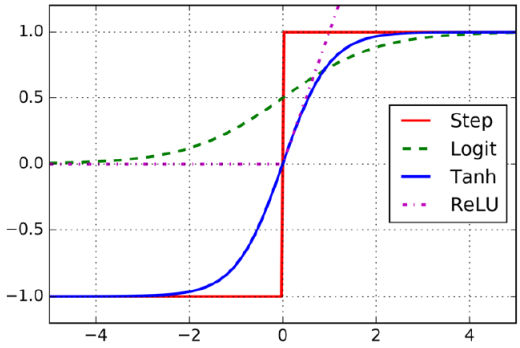

Figure 1: Activation functions

7 Question 7

The new values of the weights are: $w_2^{(next)} = 0.18128$ and $w_8^{(next)} = 0.54713$.
Below there are the calculations to compute those values.

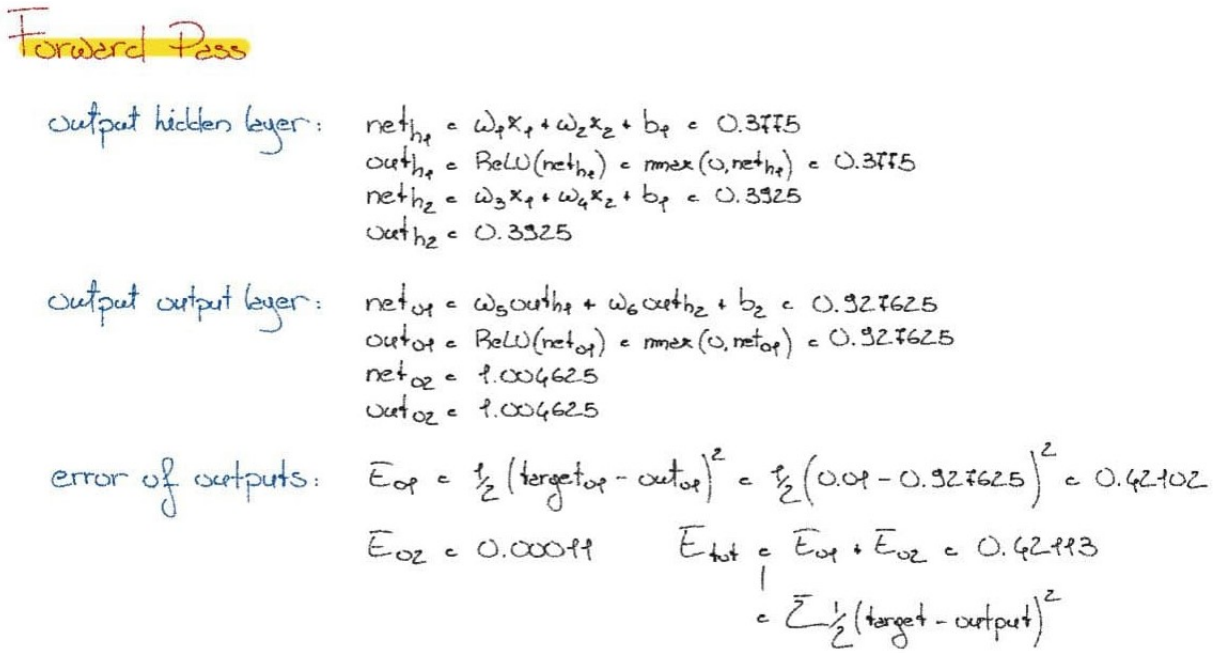

Figure 2: Backpropagation - Forward Pass

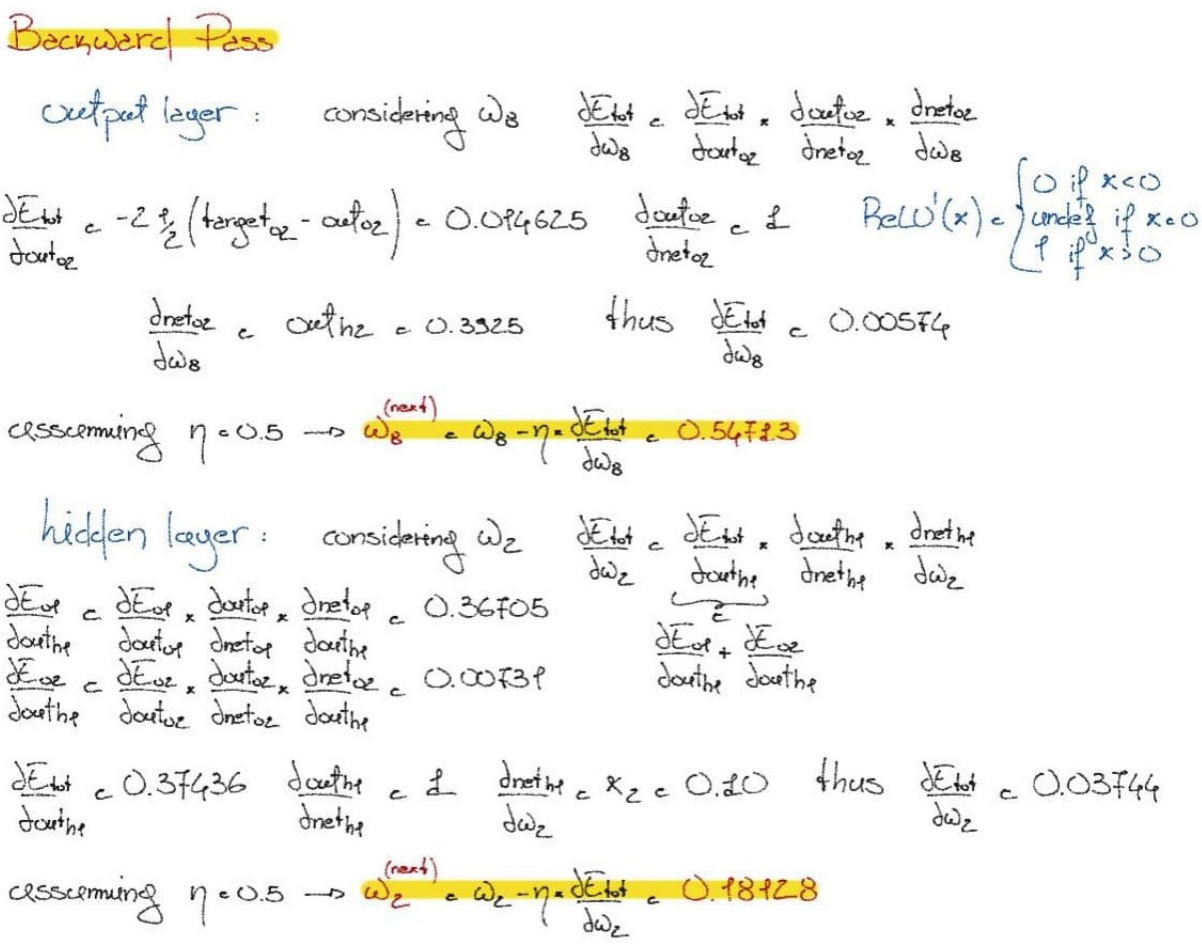

Figure 3: Backpropagation - Backward Pass