Task_7.4

Machine Learning (WiSe 2025/2026)

Author: Suvansh Shukla
Matriculation No. 256245

Assignment 7 Task 7.4

Three other activation functions besides the sign (binary step) function and the sigmoid function are the following:

Softmax function

A **Softmax** activation function that converts the raw output scores into a vector of probabilities
that sum to 1. The class with the highest probability is the network's prediction.

Softmax function formula:

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{j=1}^{C} e^{z_j}}$$

Where C is the number of classes.

The advantages of Softmax function are[1]:

- highlights the largest values and suppresses values which are significantly below the maximum value
- converts a vector of real numbers into a probability distribution (range between 0 and 1) which allows for direct interpretation
- suitable for multi-class problems
- works well with gradient descent
- enhances model generalizations, makes model confident about most probable predictions and reduces confidence in in correct predictions

The disadvantages of Softmax function are[2]:

- Overconfidence: Tends to produce extremely confident predictions even for uncertain inputs.
- Sensitivity to Outliers: Small variations in logits can cause large shifts in probability outputs.
- Softmax Bottleneck: Limited ability to model complex relationships between output classes.
- Poor Calibration: Predicted probabilities often do not align with true likelihoods.
- Gradient Saturation: Can cause vanishing gradients when one class probability dominates others.

ReLU (Rectified Linear Unit) Function

The ReLU is one of the most popular and widely used activation functions. This function provides non-linearity to the model for better computation performance.[3]

The ReLU activation function has the form:

$$f(x) = max(0, x)$$

The ReLU function outputs the maximum between its input and zero. For positive inputs, the output is equal to the input. For strictly negative outputs, the output of the function is equal to zero.

The advantages of ReLU are:

- The helps mitigate the vanishing gradient problem
- The Since ReLU is zero for all negative inputs, it leads to sparse activations leading to more efficient computation
- allows networks to scale to many layers without a significant increase in computational burden, compared to more complex functions like tanh or sigmoid

The disadvantages of ReLU are:

- dying ReLU problem: Since ReLU always outputs NULL values for negative inputs, this can cause the weights to update in such a way that the neuron will never activate on any data point again
- the opposite is also possible: the exploding gradients problem occurs when the gradients get increasingly large, leading to huge parameter updates and divergent training

Leaky ReLU

Leaky ReLU has the form:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

This was made specifically to address the "dying ReLU" problem. To solve this problem Leaky ReLU uses a multiplying factor for negative inputs. This results in a function that will not be zero but will instead have a small negative slope.

The advantages of Leaky ReLU are:

- solves the dying ReLU problem
- computationally efficient
- preserves features information

The disadvantages of Leaky ReLU:

- (minor) added computation, owing to multiplication
- have an extra parameter that requires tuning (slope)
- does not guarantee learning: the small gradient α might not be large enough to be effective still leading to very slow learning for those units

1. https://botpenguin.com/glossary/softmax-function ↩
2. https://www.geeksforgeeks.org/deep-learning/the-role-of-softmax-in-neural-networks-detailed-explanation-and-applications/ ↩
3. https://www.datacamp.com/blog/rectified-linear-unit-relu ↩