The goal of this task is to determine the number of boolean functions that are linearly separable for dimensions $n = 2, 3, 4, 5$. Initially, the `generate_boolean_functions` function generates all possible Boolean combinations (size 2^n) for a given dimension, it creates binary combinations of $\{0, 1\}$ and replaces 0 with -1. To ensure that each function has been used only once, the `has_been_processed` function compares two boolean functions. Then the perceptron is being trained. Weights are initialized with a normal distribution with $\mu = 0$ and variance $1/n$. The perceptron is trained using a learning rate of 0.05, following the learning rules given in the task description, according to Eq. 5.18 in [1], over 20 epochs. In function `count_linearly_separable_functions`, a random Boolean function is generated and checked for uniqueness then it trains the perceptron to separate it, if it successfully separates it then it's counted as linearly separable function. After 10,000 iterations, the count of separable functions is displayed.

Total amount of boolean functions in dimension n is 2^{2^n} while the amount (theoretical) of linearly separable boolean functions presented in Table 1 is according to [2]. Observe the share of linearly separable functions (theoretical amount) out of the total amount of boolean functions decreases exponentially when the dimension increases.

Dimension	My results	Theoretical amount	Total amount of boolean functions
2	14	14	16
3	104	104	256
4	254	1,882	65,536
5	0	94,572	4,294,967,296

Table 1: Number of linearly separable boolean functions (according to my results and theory) and total amount of boolean functions for dimension $n = 2, 3, 4, 5$.

As one can see from the table 1 my results are correct for dimensions 2 and 3, but deviate for 4 and 5. The results for dimensions 2 and 3 are correct because the total number of Boolean functions (16 for dimension 2 and 256 for dimension 3) is way less than the trials 10,000, (N). But for dimension 4 the total amount of boolean functions are 65,536, this is more than our trials, only 10,000. This means we didn't cover all the Boolean functions. For dimension 5, the difference is even bigger. I got a result of 0, but the theoretical value is 94,572. This is because there are way more Boolean functions (4,294,967,296) than the amount I trained my perceptron, which was just 10,000. So in order to find more linearly separable boolean functions, especially for higher dimensions, one need to increase N.

However, even setting N = total amount of Boolean function isn't foolproof due to the random sampling. Out of curiosity, I experimented with, $N = 256$ (total for dimension 3) and it yielded only 68 separable functions, which is less than the theoretical 104. Hence, N must be substantially bigger than the total amount of boolean functions in a certain dimension in order to capture most linearly separable functions.

References

[1] Bernhard Mehlig. *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*. English. Cambridge University Press, 2021. ISBN: 9781108494939.

[2] Wikipedia. *Linear separability*. Accessed: 2023-09-09. URL: `https://en.wikipedia.org/wiki/Linear_separability`.