Modelling of Cognitive Processes

Test 2

You have 1 hour for this test. You can look up info online during the test. You are also allowed to use code from the practical sessions, or code you wrote earlier.

Comment your code, and make sure it runs. A code that does some of the parts of the test but runs, is better than a code that "can potentially" do everything but does not run. Also, even if does not run properly, send us the last version of your script with comments so we can evaluate what you were trying to do.

Mind that the problem statement consists of 2 pages.

Problem statement

Human children learn throughout their childhood what makes a sentence 'grammatical', and which sentences are considered 'unlikely' taking into account the language knowledge they have so far. This process is quite interesting, but we are not quite sure whether 'learning grammar' is a linearly separable problem or not. Therefore, we ask you to help us out.

In the world this test was made in, a word is considered grammatical when the first and the last letter are the same. For simplicity, we will assume that we only have **4 letters**: A, B, C and D to work with. To make it even easier, we will assume that communication always works with **three letter words**. So, likely words are: 'ABC', 'ABA', 'CDC' and 'BAC'. Mind that only the second and the third word (the underlined ones) are considered to be grammatical (because the first and the last letters are equal), while the others are labeled as nonsensical.

The steps you should take:

1. Find a way to generate all possible words that can be formed. You should have 4^3 (64) different words.
2. Make sure that your words are translated into appropriate patterns that can be presented to your model. Do *not* code your words as strings ([‘A’, ‘B’, ‘A’] to represent the word ‘ABA’ will not work), or integers ([1,2,1] to represent the word ‘ABA’ will not work). Try to work with arrays of 0’s and 1’s to represent letters, and their position in the word.
3. Make an input array by repeating each possible input 50 times. So, your final input array should have a length of 3200 (50 x 64). Make an associated output array (1 if the word is considered grammatical, 0 if otherwise).
4. Train a two-layered Perceptron 50 times on the data. For each model fit use a *different* training- and test set. Keep track of the accuracy for each model fit, and print the average accuracy for your Perceptron.
5. Fit a three-layered Perceptron. Find the minimal model: the model with the lowest amount of hidden units with is able to achieve a 100% accuracy 50 times in a row. Mind that the training- and testing set should be different for the 50 runs.
6. Based on your acquired results, describe why the two-layered Perceptron has a worse / equal / better performance than the three-layered one.