AIML 2024-2025 Trial Coursework

September 12, 2024

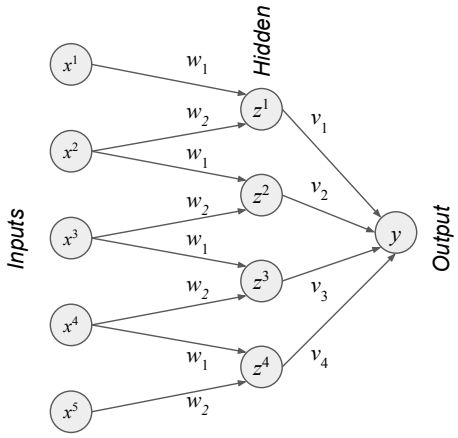

Figure 1: Convolutional neural network for coursework assignment.

Problem The goal of this take-home assignment is to implement, in Python, a simple two-layer convolutional neural network (CNN) with five inputs $x_1, \ldots, x_5$, four hidden nodes $z_1, \ldots, z_4$ and one output y with ReLU activations, according to the diagram shown in Figure 1. The hidden layer and output of the CNN is to be computed along with the gradient of the hidden layer and output with respect to parameter w_1. The values of the parameters will be $w_1 = 1.2$, $w_2 = -0.2$, $v_1 = -0.3$, $v_2 = 0.6$, $v_3 = 1.3$ and $v_4 = -1.5$.

Instructions The CNN implementation is to be computed using a single Python function in single Python file. The interface to the function should be in the **precise format**,

$$\mathtt{y}, \mathtt{z} = \mathtt{convnet}(\mathtt{x}) \tag{1}$$

where $\mathtt{x} = [\mathtt{x}_1, \mathtt{x}_2, \mathtt{x}_3, \mathtt{x}_4, \mathtt{x}_5]$ is a list of five numerical inputs (for example, a set of real numbers $\mathtt{x} = [0.3, -1.5, 0.7, 2.1, 0.1]$), and it should return the value of `y` as a **number of the type** `dual` and, $\mathtt{z} = [\mathtt{z}_1, \mathtt{z}_2, \mathtt{z}_3, \mathtt{z}_4]$ as a **list of four numbers of type** `dual` defined in the course code module `ad.py`. Therefore, when testing, you should expect to import this module. The implementation should use the specific values of the weight parameters given above.

Submission To prepare the Python code file for submission, it must be named in the format `initials_studentid.py`, for instance if your initials are 'AJD' and your ID is 5716631 then your file should be named `ajd_5716631.py`. Submit the file through the Trial Assignments page on Canvas. The deadline for submissions is **12pm UK time, 7th November 2024**.

Marking The function will be marked automatically by calling it inside Python, and checking the results against a model solution. A fully correct solution will receive 20 marks. A solution which has a partially correct

interface, or produces only partially correct output values for y and/or z, will lose marks accordingly. A solution which does not have the correct interface may score 0 marks, therefore it is critical that the interface (1) is correctly implemented.