2023-August-Mathematics of Network Algorithms

Quiz 4

- Deadline: 15th Nov, 1:10 pm. Please submit your assignment in the specified format here.
- You **must** submit python files named *enrollment-nr-assignment-nr-question-nr-student-name*.py
  For example, for the student XYZ with enrolment number 20251010, a solution for the first question should be in the file `20251010-03-01-XYZ.py`.
- Your code will be evaluated with the command `$ python3 20251010-03-01-XYZ.py`.
- Any deviation from these instructions related to submission will adversely affect the number of test cases your algorithm can solve.
- The points for each question will be determined by the quality of the output.

---

1. (15 pts) [Fashion MNIST] Write a python program that creates a Convolutional Neural Network to train `fashion_mnist` dataset from `keras` that has 3 hidden layers. The first two hidden layers are convolutional Layers with the number of perceptrons being x and y, respectively. The kernel size for both the layers is z. The third layer contains 16 perceptrons. Find the values of $x \in [4, 8, 16]$, $y \in [8, 12]$ and $z \in [3, 5]$ that minimises the test accuracy.

   Your output should be a single line containing your roll number, optimal test accuracy and the corresponding configuration.

   For example, if the best test accuracy is 93.45% when $x = 16$, $y = 8$, and $z = 5$, then your output should **only** be

   $20251010 \ 0.9345 \ [16, 8, 5]$

   and no other information.

2. (15 pts) [IMDb] Write a python program that compares two ANN, say $\mathcal{A}_1$ and $\mathcal{A}_2$ to classify IMDb dataset. Each of these ANN has two hidden layer with 16 perceptrons in each of them. For the second ANN, its first hidden layer is RNN. Evaluate both the networks with the same parameters and print their test accuracy.

   Your output should be a single line containing your roll number, the test accuracy for the first ANN, and the test accuracy for the second ANN.

   For example, if your test accuracies are 80.33% and 85.56%, respectively, then the output should **only** be

   $20251010 \ 0.8033 \ 0.8556$

   and no other information.

   Hints: `reshape`, `astype`, `verbose`, `to_categorical` from `keras.utils`, `pad_sequences` from `preprocessing.sequence`.