Feedforward Neural Networks

Assignments: Neural Networks

[Marks = 15]

Instructions to follow: Please perform your assignments in Google colab

Main codes for Grading

1. Download the "nn.py" file and copy the content into your new python Notebook.
2. Next, please name your Notebook as "nn". Don't alter the names, it has to be "nn"
3. Follow the scripts /instruction and complete the functions with your code. You have to just replace ## YOUR CODE HERE ## with your actual codes.
4. Once all the codes are written, next download this notebook as .ipynb file and .html file and .py file (.ipynb file, .html file for grading and .py file for testing your codes). ***These are your solution files.***
   Note: The evaluation is solely dependent on this notebook for all correct code implementations. No marks are allocated for testing. The testing process is included for educational purposes to illustrate how codes are validated in industries to ensure their correctness

Implementation of a feedforward neural network for training and conduct experiments on hyper-parameter choices in feedforward networks

5. Download the provided "test_nn.py" file and copy the content into a new python Notebook. Name this Notebook as "test_nn". (If test_nn.ipynb file is provided skip to the next step)
   a. Upload the "nn.py" file (your solution file downloaded in 4) into this Notebook. File → Upload Notebook → Upload → Browse and select your file
   b. Now just run the codes.
      Don't change any codes
   c. Download this Notebook as .py file (i.e., as "test_nn.py"). ***This will be further used in testing setup.***

Running Tests on the codes using the pytest command

6. Open the provided Notebook " feedforward_networks.ipynb" in Google colab.

7. Also upload “nn.py” file (your solution file downloaded in 4) and “test_nn.py” (downloaded in 5.c) into this Notebook.
8. Mount the Google Drive to access files stored in Google Drive within the Colab environment. (code already provided)
9. Now, run all the codes as given in the notebook.
   Please don’t change any codes.
10. Run these tests using the !pytest command. Check how many of your codes are written correctly.
11. Download this file after all the testing’s and submit.

Files to be submitted for evaluation

- Your Solution file “nn.ipynb” and “nn.html” for grading (downloaded in 4)
- Your Solution file downloaded as “nn.py” (downloaded in 4)
- Testing file “feedforward_networks.ipynb” for pytest results (downloaded in 11)

Gradings:

Note: Grading is solely based on your codes submitted in the "nn.ipynb" file. No marks are assigned for testing your codes.

```
1. def create_auto_mpg_deep_and_wide_networks(
        n_inputs: int, n_outputs: int) ->
Tuple[tensorflow.keras.models.Model,
                                                tensorflow.keras.models
.Model]:
    """Creates one deep neural network and one wide neural network.
    The networks should have the same (or very close to the same)
number of
    parameters and the same activation functions.

    The neural networks will be asked to predict the number of miles
per gallon
    that different cars get. They will be trained and tested on the
Auto MPG
    dataset from:
    https://archive.ics.uci.edu/ml/datasets/auto+mpg

    :param n_inputs: The number of inputs to the models.
    :param n_outputs: The number of outputs from the models.
    :return: A tuple of (deep neural network, wide neural network)
    """
```

[Marks = 5]

2.
```
def create_activity_dropout_and_nodropout_networks(
        n_inputs: int, n_outputs: int) ->
Tuple[tensorflow.keras.models.Model,
                                                    tensorflow.keras.models
.Model]:
    """Creates one neural network with dropout applied after each
layer, and
    one neural network without dropout. The networks should be
identical other
    than the presence or absence of dropout.

    The neural networks will be asked to predict which one of six
activity types
    a smartphone user was performing. They will be trained and tested
on the
    UCI-HAR dataset from:
    https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+
using+smartphones

    :param n_inputs: The number of inputs to the models.
    :param n_outputs: The number of outputs from the models.
    :return: A tuple of (dropout neural network, no-dropout neural
network)
    """
```
[Marks = 5]

3.
```
def create_income_earlystopping_and_noearlystopping_networks(
        n_inputs: int, n_outputs: int) ->
Tuple[tensorflow.keras.models.Model,
                                                    Dict,
                                                    tensorflow.keras.models
.Model,
                                                    Dict]:
    """Creates one neural network that uses early stopping during
training, and
    one that does not. The networks should be identical other than the
presence
    or absence of early stopping.

    The neural networks will be asked to predict whether a person makes
more
    than $50K per year. They will be trained and tested on the "adult"
dataset
    from:
    https://archive.ics.uci.edu/ml/datasets/adult

    :param n_inputs: The number of inputs to the models.
    :param n_outputs: The number of outputs from the models.
```

```
    :return: A tuple of (
        early-stopping neural network,
        early-stopping parameters that should be passed to Model.fit,
        no-early-stopping neural network,
        no-early-stopping parameters that should be passed to Model.fit
    )
    """
```

[Marks = 5]