Assignment 3: EN4553 (Machine Vision)

University of Moratuwa

January 1, 2023

1. (30 points) We analyze fitting a machine learning model to a small dataset in this question. Download the dataset corresponding to your index number from https://tinyurl.com/mv2022-a3. Your dataset has three partitions: train, validation and test. As usual, the model will be trained on the train partition. Hyperparameters are tuned on the validated partition. Labels for the test partition are not available to you. They are, however, available to the grader. All three partitions come from the same data distribution. The i^{th} row of `y_train.txt` contains the label for the i^{th} row of `x_train.txt`. The same holds for the validation set.

   In this toy problem, the validation set is bigger than the train set. This is not usual in real-world settings, it has been done here only to make the validation metrics more reliable. Note also that the Python type hints in the template codes below work only on Python versions >= 3.5.

   (a) (5 points) Complete the following Python function to read the dataset as a set of arrays. (Hint: you may use the `np.loadtxt()` function.)

```
import numpy as np
from typing import Tuple

def load_dataset(
    src_dir: str
) -> Tuple[np.ndarray, np.ndarray, np.ndarray, np.ndarray, np.ndarray]:
    """Load the dataset as a set of numpy arrays.

    Args:
        src_dir: Directory where dataset files are stored.

    Returns:
        (x_train, y_train, x_val, y_val, x_test) tuple where each array is one dimensional.
    """
    pass  # Implement here.
```

   (b) We will be fitting a linear regression model $y = \sum_{i=1}^{n} w_i x^i$ where each w_i is a learnable parameter and n is a hyperparameter. Variables x and y represent rows in `x_train.txt` and `y_train.txt`.

      i. (5 points) Implement the following function to make input features for the above linear regression model.

```
def get_features(x: np.ndarray, n: int) -> np.ndarray:
    """Creates n-th degree polynomial features for the given vector x.

    Example usage:
    get_features(np.array([1.0, 2.0, 3.0]), 3) outputs
    np.array([[ 1.,  1.,  1.],
              [ 2.,  4.,  8.],
              [ 3.,  9., 27.]])

    Args:
        x: A numpy array of shape (num_examples, ) or (num_examples, 1).
```

```
        n: The degree of the polynomial features.

    Returns:
        A matrix of shape (num_examples, n) where the j-th column is equal to
        the vector x raised, elementwise, to the power j.
    """
    pass  # Implement here.
```

ii. (10 points) Use the above function to complete the following implementation. (Hint: you may use the `sklearn.linear_model.LinearRegression` class.)

```
def fit_and_evaluate(
        x_train: np.ndarray, y_train: np.ndarray,
        x_val: np.ndarray, y_val: np.ndarray,
        n: int
) -> Tuple[float, float]:
    """Fits an n-th degree polynomial and outputs train and validation MSE.

    Fits a linear regression model y = sum_{i=1}^n w_i x^i to the given train
    set and outputs the mean-squared-error (MSE) on train and validation sets.

    Args:
        x_train: Input features for the train set. Has shape (num_train, )
        y_train: Targets (labels) for the train set. Has shape (num_train, )
        x_val: Input features for the validation set. Has shape (num_val, )
        y_val: Targets (labels) for the validation set. Has shape (num_val, )
        n: The degree of the polynomial fit. See the above equation.

    Returns:
        (train_mse, val_mse), tuple of MSE on train and validation sets.
    """
    # Fit the model on the train set.
    pass

    # Generate model predictions for the train set and calculate the MSE.
    y_predict_train = ...
    train_mse = mse(y_train, y_predict_train)

    # Similarly, calculate the MSE on the val set.
    val_mse = ...

    return train_mse, val_mse
```

(c) (5 points) Use the above function to calculate and plot train and validations MSEs against $n = 1, 2, \ldots, 10$. Include this graph in your answer sheet. Which n value would you pick for your final model?

(d) (5 points) Use the model selected above to make predictions on the test set. Include your predictions in a file named `<your-index-number>_y_predict_test.txt`. For example, `180000X_y_predict_test.txt`.

Deliverables: Upload a single .zip file containing exactly two files: a PDF containing answers to questions (a) to (c), and a plain text file containing the answer to (d), with the correct file name. Name the zip file as: `<your-index-number>_assignment3.zip`. You can use an interactive environment like Jupyter Notebook or Google Colab to complete the exercise. Uploading the full source code as `.py` files is not needed.