0em

CS Assignment: Predicting Mango Types with Neural Nets

Using Length, Mass, and Both as Features

Objective

The goal of this assignment is to build intuition for neural networks by working with a simple mango dataset. Your dataset contains three columns:

- `length` (cm)
- `mass` (grams)
- `type` (categorical label, e.g., mango variety)

You will design Python functions that attempt to predict mango type using:

1. Length only
2. Mass only
3. Both length and mass

—

File Structure

Your project should follow this structure:

```
project/
   data/
        mango_data.csv
   src/
        dataset_loader.py
        mango_model.py
        train.py
   README.md
```

—

Part 1: Dataset Loader

In `dataset_loader.py`, write a function to load the CSV.

[language=Python] def load$_d$ataset($path : str =$ "$data/mango_data.csv$") : $readcsvextractX(features)$

—

Part 2: Prediction with Length Only

Part 4: Prediction with Length + Mass

Extend to accept both features together.

[language=Python] def mango$_predict_two_features(length, mass, weight1, weight2, bias)$: $weighted sum with two inputs apply activation return prediction$

—

Part 5: Loss Function

Implement Mean Squared Error (MSE) in `mango_model.py`.

[language=Python] def compute$_loss(y_true, y_pred)$: $implement MSE return loss$

—

Part 6: Training Script

In `train.py`, write a loop that:

1. Initializes weights and bias
2. Loops over the dataset
3. Runs predictions
4. Computes loss
5. Adjusts weights using gradient descent
6. Prints loss each epoch

—

Deliverables

- `mango_data.csv`
- Functions:
  - `mango_predict_length()`
  - `mango_predict_mass()`
  - `mango_predict_two_features()`
  - `compute_loss()`
- `train.py` script that runs training

—

Stretch Goals

- Normalize your inputs (scale length/mass before using them)
- Add plots of the loss curve
- Implement a softmax classifier for multiple mango types