Assignment 4: Build a Fully Connected ANN Architecture

40 Points

Deliverables: Answers to background questions; Python code in an. ipynb file.

Background Questions

B1: Why are activation functions generally included in artificial neural network architectures? (4 Points)

B2: Explain the difference between weights and biases for a neuron in a fully connected architecture. (4 Points)

B3: Why are batch normalization layers commonly included in modern ANN architectures? (4 Points)

B4: Explain the number of output nodes required for (1) a regression task, (2) a binary classification task, and (3) a multiclass classification task. (4 Points)

B5: Explain what operation(s) need to be added to make an architecture return class probabilities as opposed to class logits for a multiclass classification problem. (4 Points)

B6: Explain why it is generally necessary to normalize or re-scale all predictor variables to a consistent scale before they are provided as input to an ANN architecture. (4 Points)

Tasks (PyTorch):

T1: By subclassing nn.Module, build a fully connected artificial neural network architecture with the following characteristics or that meets the following criteria: (4 Points)

1. Accepts 7 input predictor variables.
2. Predicts logits for three classes.
3. Contains three hidden layers with the following input and output sizes.
    a. In = 7; Out = 64
    b. In = 64; Out = 128
    c. In = 128; Out = 3
4. Includes ReLU and batch normalization layers between all the fully connected layers other than the last.

T2: Generate some random example data with the correct shape as a tensor using PyTorch. Instantiate a model and predict the new data to test that the model is working. (4 Points)

T3: List the number of trainable parameters in your model. Break the parameters into the following categories: (4 Points)

1. Fully connected layer weights

2. Fully connected layer biases
3. Batch normalization gains
4. Batch normalization shifts