Framework for Coupling ANN trained on Python to C/C++ Code

Description –

The developed framework includes a header file, ANN.H, which contains the declaration and definition of a class named ANN. This class provides essential functions to integrate a trained artificial neural network (trained in Python) into C/C++ code. It also offers the functionality needed to perform inference (prediction) on the loaded ANN for specific user input points. Additional functions within the class facilitate preprocessing and postprocessing (currently limited to normalization) of input data points when the minimum and maximum values used during training are provided. The class includes multiple constructors that enable users to load ANNs by specifying the number of inputs/outputs (depending on the architecture) or by providing paths to preprocessing text files.

The ANN.H header file serves as a lightweight interface that utilizes ONNX Runtime to load ANNs and perform inference. It provides an accessible framework for users, abstracting the complexities of the ONNX Runtime API. This framework was developed using onnruntime-linux-x64-1.14.1, and it is recommended that this specific version be used in conjunction with the framework. Since ONNX Runtime is employed, all trained models must be converted to the ONNX format. For models developed in Python using PyTorch, they can be directly saved and converted to ONNX format through PyTorch. For TensorFlow users, the tf2onnx package should be installed to facilitate the conversion to ONNX format.

A sample test case, titled `runANN.cpp`, has been developed to demonstrate the use of the framework. This test case illustrates how to load and use a trained neural network model named `C12_N2_TML_PR.onnx`, which features 4 inputs and 3 outputs. Additionally, two text files containing preprocessing data, `mins_data.txt` and `maxs_data.txt`, are provided and utilized within the test case.

To compile the test case, a `Makefile` is also provided. The `Makefile` specifies the necessary details, including the compiler to be used, the required include and library files, their paths, and the name of the resulting executable.In order to run the test case, please run the following commands –

```
make
./runANN
```

Files –

For Coupling - ANN.H, ANN.C
Open-source files – onnruntime-linux-x64-1.14.1
Test cases – runANN.cpp