#pragma once

#include <twml/Tensor.h>

#include <twml/RawTensor.h>

#include <twml/ThriftWriter.h>

namespace twml {

// Encodes a batch of model predictions as a list of Thrift DataRecord

// objects inside a Thrift BatchPredictionResponse object. Prediction

// values are continousFeatures inside each DataRecord.

//

// The BatchPredictionResponseWriter TensorFlow operator uses this class

// to determine the size of the output tensor to allocate. The operator

// then allocates memory for the output tensor and uses this class to

// write binary Thrift to the output tensor.

//

class BatchPredictionResponse {

private:

uint64\_t batch\_size\_;

const Tensor &keys\_;

const Tensor &values\_; // prediction values (batch\_size \* num\_keys)

const Tensor &dense\_keys\_;

const std::vector<RawTensor> &dense\_values\_;

inline uint64\_t getBatchSize() { return batch\_size\_; }

inline bool hasContinuous() { return keys\_.getNumDims() > 0; }

inline bool hasDenseTensors() { return dense\_keys\_.getNumDims() > 0; }

inline uint64\_t getPredictionSize() {

return values\_.getNumDims() > 1 ? values\_.getDim(1) : 1;

};

void encode(twml::ThriftWriter &thrift\_writer);

template <typename T>

void serializePredictions(twml::ThriftWriter &thrift\_writer);

public:

// keys: 'continuousFeatures' prediction keys

// values: 'continuousFeatures' prediction values (batch\_size \* num\_keys)

// dense\_keys: 'tensors' prediction keys

// dense\_values: 'tensors' prediction values (batch\_size \* num\_keys)

BatchPredictionResponse(

const Tensor &keys, const Tensor &values,

const Tensor &dense\_keys, const std::vector<RawTensor> &dense\_values);

// Calculate the size of the Thrift encoded output (but do not encode).

// The BatchPredictionResponseWriter TensorFlow operator uses this value

// to allocate the output tensor.

uint64\_t encodedSize();

// Write the BatchPredictionResponse as binary Thrift. The

// BatchPredictionResponseWriter operator uses this method to populate

// the output tensor.

void write(Tensor &result);

};

}