#pragma once

#include "tensorflow/core/framework/common\_shape\_fns.h"

#include "tensorflow/core/framework/op.h"

#include "tensorflow/core/framework/shape\_inference.h"

#include "tensorflow/core/framework/op\_kernel.h"

#include "tensorflow/core/platform/env.h"

#include "tensorflow/core/lib/io/random\_inputstream.h"

#include <twml.h>

#include <string>

using tensorflow::int64;

using tensorflow::Status;

using std::string;

class BlockFormatReader : twml::BlockFormatReader {

public:

explicit BlockFormatReader(tensorflow::io::InputStreamInterface \*stream)

: twml::BlockFormatReader() , stream\_(stream) {

}

// Read the next record.

// Returns OK on success,

// Returns OUT\_OF\_RANGE for end of file, or something else for an error.

Status ReadNext(string\* record) {

if (this->next()) {

return stream\_->ReadNBytes(this->current\_size(), record);

}

return tensorflow::errors::OutOfRange("eof");

}

uint64\_t read\_bytes(void \*dest, int size, int count) {

uint64\_t bytesToRead = size \* count;

std::string current;

// TODO: Try to merge ReadNBytes and the memcpy below

// ReadNBytes performs a memory copy already.

Status status = stream\_->ReadNBytes(bytesToRead, &current);

if (!status.ok()) {

return 0;

}

memcpy(dest, current.c\_str(), bytesToRead);

return count;

}

private:

tensorflow::io::InputStreamInterface \*stream\_;

TF\_DISALLOW\_COPY\_AND\_ASSIGN(BlockFormatReader);

};