Improved Exception Handling in MXNet - Phase 2

Introduction

MXNet currently has support for handling dmlc::Error exceptions in the backend and propagating them to frontend for both the operators and the iterators case.

There is still some work needed w.r.t Exception Handling. This document lists out all the further improvements that can be done for the exception handling feature.

Improved Exception Types for the Backend

Currently, there are only three exceptions available for MXNet C++ backend: dmlc::Error, InferTypeError (inherits from dmlc::Error) and InferShapeError (inherits from dmlc::Error).

There are more scenarios where exceptions are thrown. Here are some examples: MemoryAllocationError, ConfigurationError(GPU_ENABLED), KVStoreError.

Add support for these exceptions in C++ backend. Exceptions that can be shared with dmlc-core should go in dmlc-core.

Here is the implementation of dmlc::Error: https://github.com/dmlc/dmlc-core/blob/master/include/dmlc/logging.h#L31

Here are a few examples for the other exceptions inheriting from dmlc::Error: https://github.com/apache/incubator-mxnet/blob/master/src/operator/operator_common.h#L74

and https://github.com/apache/incubator-mxnet/blob/master/src/operator/operator_common.h#L85

Improved Exception Types for the Frontend Language Bindings - (Python, Scala, Perl..)

Will require a mapping of error codes and corresponding frontend exceptions.

The setting of error code for the frontend exception currently happens here: https://github.com/apache/incubator-mxnet/blob/master/src/c_api/c_api_common.h#L59

Frontend exception checking for Python example: https://github.com/apache/incubator-mxnet/blob/master/python/mxnet/base.py#L137

Support for handling exception thrown from consumed libraries

Currently we are handling only dmlc::Error and terminating the process for other exceptions. Once we have finished the above two tasks,

It should be easy to add exception mapping in the frontend corresponding to std::exception in backend. This will ensure that all exceptions inheriting from dmlc::Error

and std::exception are handled and propagated to frontend.