Gluon API for Scala/ Clojure/ Java packages

Original Dev Thread https://lists.apache.org/thread.html/aecfae9381286c8864f0b0a63211f2ac498eafb1ffb6fcdd863e67a6@%3Cdev.mxnet.apache.org%3E

Background & Motivation

Currently the Scala package supports the Module and the Feed Forward API. The Clojure package builds on the Scala package and supports the Module API. The Gluon API is only supported so far in the Python package, however it is more full featured and all of the newer documentation and books and online resources like Dive into Deep Learning use the Gluon API. Supporting this API would allow the JVM packages to grow and to eventually share a common API for documentation and tutorials.

For example - Dive Into Deep Learning could have parallel texts and exercises for Scala and Clojure (and others ...)

Challenges

- Keeping the MXNet JVM packages coordinated, so there is a minimum of code duplication and no divergence in the jni bindings which would complicate deployment. However allow interested contributors to pick up areas of development that interest them independently.
- The Gluon API is large so finding a good area to tackle.
- The Current Scala native bindings might not be sufficient to support the Gluon API.
- How to build out pieces one by one instead of all at once.
- Some syntax will be different in different langs.

Proposal

- Have guidelines about developing Gluon API for JVM langs.
  - Any new JNI binding development needs to go into the Scala package first to avoid divergence and packaging issues in other packages.
  - If a contributor wants to add a package to the Clojure package first (for example https://mxnet.incubator.apache.org/api/python/gluon/model_zoo.html) that doesn't touch the JNI bindings, that development can happen in Clojure first and then when it arrives in the Scala package can be converted over if desired to have a single implementation. This will allow some decoupling of development.
  - The API should follow as closely as possible to the Gluon API and live under the same namespace.
    - Example: The model zoo API in the Clojure package would be *org.apache.clojure-mxnet.gluon.model-zoo*
  - Use the existing code when possible to build out the Gluon API.
  - If implementing gluon.model.zoo or gluon.data before core functionality, have it work with existing module code.

Questions

- Autograd is an interesting area. Currently there are no good JVM autograd options. To get this to work would be exciting for many users. But need to look into how it could actually work.
  - Would definitely take JNI changes
  - Large Effort

TODO: Gluon API Analysis

Gluon package	Description	Comments	Can be supported with current JNI?	Covered in Dive into Deep Learning?	Benefit	Effort
gluon.base	Base Package	Dynamic Graph support? Autograd	?	Just about everywhere		
gluon.model_zoo	Provides pre-defined and pre-trained models to help bootstrap machine learning applications		yes	• chapter_computer-vision/neural-style.md • chapter_computer-vision/fine-tuning.md • chapter_computer-vision/fcn.md • chapter_computer-vision/kaggle-gluon-dog.md	High	Low

gluon.data	Provides useful dataset loading and processing tools, as well as common public datasets.			• chapter_deep-learning-basics/fashion-mnist.md • chapter_recurrent-neural-networks/sequence.md • chapter_crashcourse/naive-bayes.md		
gluon.loss	Commonly used loss functions in neural networks			• chapter_deep-learning-basics/linear-regression-gluon.md • chapter_recurrent-neural-networks/sequence.md		
gluon.nn	Neural network blocks			• chapter_recurrent-neural-networks/sequence.md		
gluon.rnn	Recurrent neural network API			• chapter_recurrent-neural-networks/sequence.md		
gluon.contrib	Provides many useful experimental APIs for new features.			None		