Zach Tatlock
Let’s Get in the Wayback Machine
Let’s Get in the Wayback Machine
Challenges for Deep Learning IRs

- State-of-the-art models increasingly depend on:
  - Datatypes - lists, trees, graphs
  - Control flow - branches, loops, recursion
  - Whole-program analyses and optimizations
  - Any one feature “easy to bolt on”
  - Folklore suggests full, expressive IR will be slow

```
let encode = λ st.
  if(...):
    encode(step(st))
  else:
    ...
```
Challenges for Deep Learning IRs

- State-of-the-art models increasingly depend on:
  - Datatypes - lists, trees, graphs
  - Control flow - branches, loops, recursion
  - Whole-program analyses and optimizations
  - Any one feature “easy to bolt on”
  - Folklore suggests full, expressive IR will be slow
The Relay IR

- Relay generalizes NNVM
- Retains graph-level optimizations
- Provides more expressive features
  - Datatypes, control flow, code re-use
- Functional semantics to simplify analysis
- Automatic differentiation + optimizations

~ “OCaml for ML”
Relay: Expressiveness + Performance

- High-level Relay models match NNVM in traditional vision inference
• High-level Relay models match NNVM in traditional vision inference
Relay: Expressiveness + Performance

- Low-cost abstraction enabled by:
  - Tensor shape inference and specialization
  - High-level operator fusion
  - Whole-program partial evaluation
Relay: Expressiveness + Performance

- Low-cost abstraction enabled by:
  - Tensor shape inference and specialization
  - High-level operator fusion
  - Whole-program partial evaluation
  - But most of all by extensible, composable optimization framework!
Relay Win: Support for New Models

• High-level Relay models for RNNs and LSTMs can outperform the rest
Relay Win: Support for New Models

- High-level Relay models for RNNs and LSTMs can outperform the rest

Plus support for new/improved targets via high-level transformations:

<table>
<thead>
<tr>
<th>Model</th>
<th>Raspberry Pi 3</th>
<th>RK3399</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18</td>
<td>323.0</td>
<td>161.0</td>
<td>466.0</td>
</tr>
<tr>
<td>MobileNet</td>
<td>288.0</td>
<td>184.0</td>
<td>451.0</td>
</tr>
<tr>
<td>Inception</td>
<td>122.0</td>
<td>85.0</td>
<td>509.0</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>139.0</td>
<td>78.0</td>
<td>673.0</td>
</tr>
<tr>
<td>TreeLSTM</td>
<td>113.0</td>
<td>83.0</td>
<td>641.0</td>
</tr>
</tbody>
</table>

### Quantization Scheme
- float32
- int8/int32
- int8/int16

### Hardware Design
- Single-Batch
- Multi-Batch
Relay Win: Support for New Models

- High-level Relay models for RNNs and LSTMs can outperform the rest

Plus support for new/improved targets via high-level transformations:
New Features

**Relay in Production**

Relay is a high-level, high-performance, data-parallel programming language designed to be an expressive interface for high-speed execution of machine learning systems. Relay supports algebraic data types, closures, control flow, and recursion, allowing it to directly represent more complex models than computation graph-based RIs (e.g., MXNet) can. In TVM v0.8, Relay is in stable phase and is ready for production.

- Algebraic Data Types (ADT) support (#2342, #2375). ADT provides an expressive, efficient, and safe way to realize recursive computation (e.g., RNN). Refer to https://dlib.cs.umass.edu/clangref/relay.adt.html for more information.
- Pass manager for Relay (#2368, #2329, #2334, #2391).
- Most frameworks have been supported in Relay, including ONNX, Keras, Tensorflow, Caffe2, CoreML, MXNNMv1, MXNet (#21246).
- Explicitly manifested memory and tensor allocations in Relay. (#33050)

**Relay Virtual Machine**

The Relay Virtual Machine (Relay VM) is the new generation of runtime to strike a balance between performance and flexibility when deploying and executing Relay programs. Previously, the graph runtime is able to utilize the fully static nature of the input graphs to perform aggressive optimization such as fully static allocation, and optimal memory reuse. When we introduce models which make use of control-flow, recursion, dynamic shapes, dynamic allocation, we must change how execution works.
Relay + You!

- Relay merged in to TVM mainline
- Documentation, tutorials, examples
- Add your own analyses and optimizations
- Target new accelerators
- Support new models
- Tons of community support!

+ many more amazing folks!
Relay + You!

• Relay merged into TVM mainline
• Documentation, tutorials, examples
• Add your own analyses and optimizations
• Target new accelerators
• Support new models
• Tons of community support!

+ many more amazing folks!