CS 744: BIG DATA SYSTEMS

Shivaram Venkataraman
Fall 2018
ADMINISTRIVIA

- Assignment 1 Grading
- Assignment 2
- Course Project
SYSTEMS FOR MACHINE LEARNING
OPTIMIZATION

\[
\min_{w \in \mathbb{R}^d} \sum_{i=1}^{N} f(w, z_i) + P(w)
\]

Function

Data (Examples)

Model

Regularization
DEEP LEARNING

ReLU

\[ \text{ReLU} \left( \max \left( \sum x_i w_i, 0 \right) \right) \]

Non-linearity
DEEP LEARNING

Stack them together!
DEEP LEARNING

ResNet18

Convolution
ReLU
MaxPool
Fully Connected
SoftMax
...

7x7 conv
54 filters
Norm
ReLU
Pool

3x3 conv
64 filters
Norm
ReLU

3x3 conv
64 filters
Norm
ReLU
Sum

drop out

3x3 conv
64 filters
Norm
ReLU
Sum

drop out

3x3 conv
128 filters
Norm
ReLU
Sum

drop out

3x3 conv
128 filters
Norm
ReLU
Sum

drop out

3x3 conv
128 filters
Norm
ReLU
Sum

drop out

3x3 conv
256 filters
Norm
ReLU
Sum

drop out

3x3 conv
256 filters
Norm
ReLU
Sum

drop out

3x3 conv
512 filters
Norm
ReLU
Sum

drop out

3x3 conv
512 filters
Norm
ReLU
Sum

drop out

3x3 conv
512 filters
Norm
ReLU
Sum

drop out

3x3 conv
512 filters
Norm
ReLU
Sum

drop out

FC (84) Softmax
MODEL TRAINING

\[ w^{(k+1)} = w^{(k)} - \alpha_k \nabla f(w^{(k)}) \]

Initialize \( w \)
For many iterations:
  Compute Gradient
  Update model
End

Stochastic Gradient Descent
Gradient using backprop
Compute Intensive!
DESIGN PRINCIPLES

- Dataflow graphs of primitive operators
- Deferred execution: Symbolic dataflow graph
- Heterogeneous accelerators
EXECUTION MODEL

Dataflow graph

Multiple concurrent executions of overlapping sub-graphs

Vertices have mutable state; shared between executions
GRAPH ELEMENTS

Tensors
- N-dimensional arrays, *dense* representation by default
- Operations take in tensors and return tensors

Variables
- Stateful operations
- Read/AssignAdd into shared buffer
- Queues for back-pressure [Volcano]
**EXECUTION MODES**

**Partial Execution**
- Input batches from queue
- Concurrent training steps
- Shared model
- “Horizontal” parallelism?

**Distributed Execution**
- Operations placed on devices
- Account for colocation
- Manual placement decisions?
- Send-Recv to stitch subgraphs
CONTROL FLOW

- Support for RNNs, LSTMs
- **Switch** and **Merge** operators to support conditionals
- **Enter**, **Exit**, **NextIteration** to support while loops

```python
input = ...  # A sequence of tensors
state = 0    # Initial state
w = ...     # Trainable weights

for i in range(len(input)):
    state, out[i] = f(state, w, input[i])
```
EXTENSIONS

Automatic Differentiation
- Given a symbolic expression, generate its gradient
- Also extend to control flow operations

Fault Tolerance
- User-level checkpointing
- Save and Restore operations in graph
- Not necessarily consistent?
SYNC VS ASYNC

(a) Asynchronous replication

(b) Synchronous replication

(c) Synchronous w/ backup worker
QUESTIONS / DISCUSSION?