ONNX Training Working Group Update

Svetlana Levitan, PhD
Senior Developer Advocate in Chicago
Center for Open-source Data and AI Technologies
IBM Cloud and Cognitive Software

April 9, 2020
Outline

• Background
• Overview of ONNX training approach
• New features in ONNX IR
• New operators
• Next steps
ONNX Training working group history and status

Working group created in February 2019

Led by IBM, meetings on Tuesdays at 10:30 am US Pacific time.

Gitter room: https://gitter.im/onnx/training

Several Pull Requests merged into Master for ONNX 1.7 release “Preview”.

Converters and open source ONNX RT do not yet support this.

Wei-Sheng Chin (Microsoft) created the proposal, with inputs from others.
Background: Why ONNX Training?

Sometimes training is a part of deployment (model refinement)

Create training spec (or possibly partially trained model) in one framework and train in another or in ONNX Runtime

More flexibility for computation-intensive workloads

Attractive for hardware manufacturers
ONNX Training Approach as described in PR #2314

1. Added a protobuf message, **TrainingInfoProto**, to store training information.
2. In TrainingInfoProto, the user can store training algorithm in algorithm field as a GraphProto.
3. Can also store initialization algorithm for resetting the model in TrainingInfoProto.**initialization**.
4. ModelProto.graph is callable inside TrainingInfoProto.algorithm.
5. ModelProto.graph.initializer is visible to nodes in TrainingInfoProto.algorithm.node.
6. Also introduced a **Gradient** operator to differentiate a function represented by a (sub-) graph and GraphCall operator to call the inference graph.
7. Defined new operators for most widely used loss functions and optimizers.
New features in ONNX IR (version 7)

onnx/onnx.proto  Added lines 210-316 and 367-377

message TrainingInfoProto {
  optional GraphProto initialization = 1;
  optional GraphProto algorithm = 2;
  repeated StringStringEntryProto initialization_binding = 3;
  repeated StringStringEntryProto update_binding = 4;
};
message ModelProto {
  ...
  repeated TrainingInfoProto training_info = 20;
};
New operators (and PR #'s)

**Loss functions**
- Negative LogLikelihood #2551
- Mean Squared Error #2570
- Softmax Crossentropy #2573

**Optimizers**
- Adagrad #1955
- SG with momentum #1959
- Adam #1970

**CORE**
- Gradient #2314
- GraphCall #2314

Domain: ai.onnx.training
Next steps

• Wei-Sheng finished ADAM PR #1970, but not in 1.7 release
• Add more details into current primitives, **define gradient behavior for each operator**
• Helper functions:
  Create TrainingInfoProto
  To go from inference to training graph and back
• Work with Converters teams to help them to support ONNX training
• How are users doing auto-diff? Need it in ONNX? Need your answers!

• Longer term: Get feedback on the spec and update
Contact me

slevitan@us.ibm.com

@SvetaLevitan