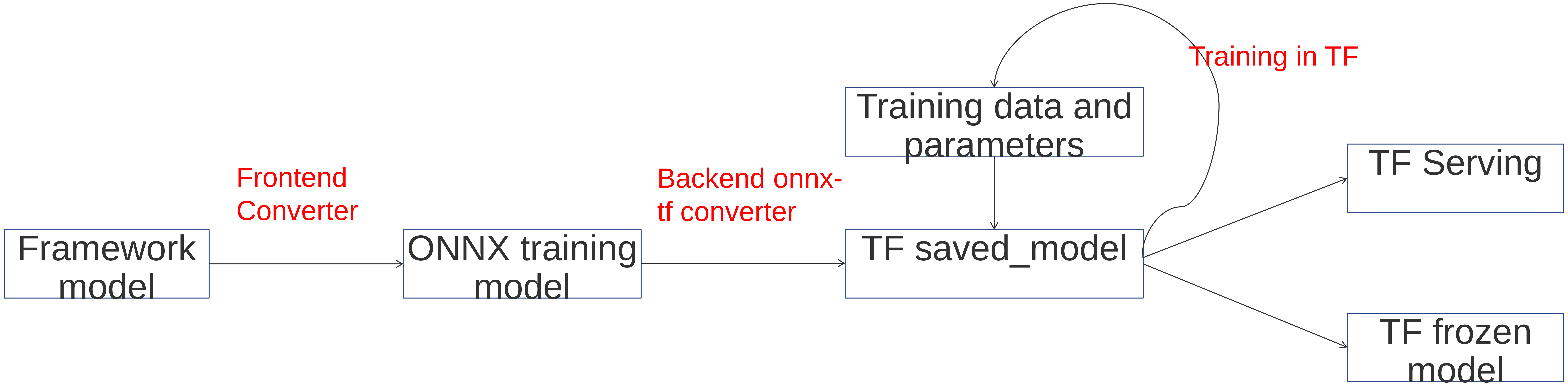
ONNX Training Prototype For Tensorflow

Workshop 8/23/2019

Goals

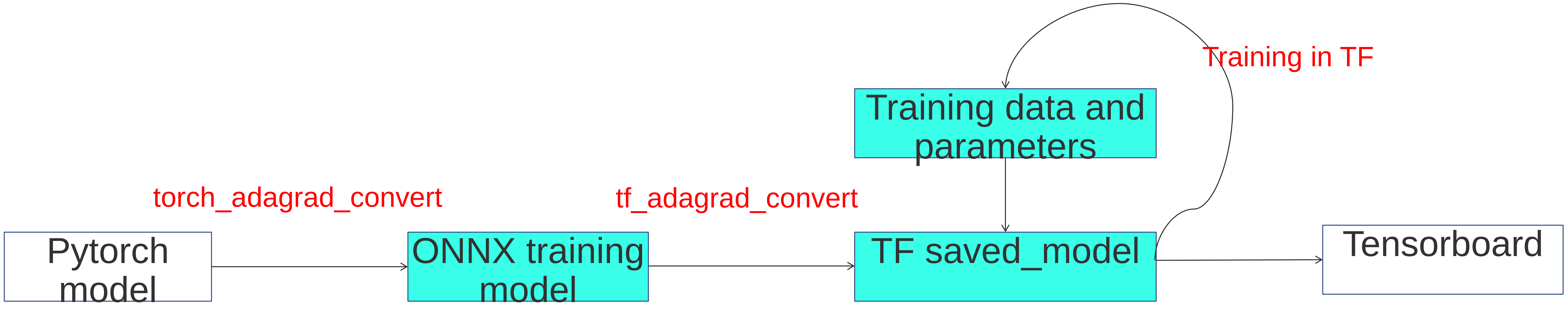
- Identify the proposed ONNX training spec can be practically generated and used in Tensorflow training by
    - Investigate and provide prototype code for conversion from ONNX training IR to Tensorflow trainable model format
    - Investigate and provide prototype code for conversion from Tensorflow trainable model format to ONNX training IR

Flow chart: converting from onnx to tf



Prototype Demo

(prototype, https://github.com/chinhuang007/onnx-tf-training
onnx branch, https://github.com/wschin/onnx/tree/training-info)



Training Info Mapping, converting from onnx to tf

ONNX	Example	Tensorflow
graph.node	A node with op_type = "MyInferenceFunction" pointing to the function with inference graph (nodes)	**N/A**
training_info.initializer	Iteration count, Learning rate (R), H	tf.train.AdagradOptimizer(**Lr**)
training_info.input	Label	tf.losses.mean_squared_error(**labels=y_true**, predictions=y_pred)
training_info.output	Loss value, New H, New W	**N/A**
training_info.algorithm	"MyTrainingFunction"	**N/A**
training_info.algorithm.input	Label, Weight, T, R…	tf.losses.mean_squared_error(**labels=y_true**, predictions=y_pred)

training_info.algorithm.node	Loss node (op_type=MSE), inference node (op_type=MyInferenceFunction), gradient nodes (op_type=gradient), optimizer node (op_type=Adagrad)	loss = tf.losses.mean_squared_error(…)
		optimizer = tf.train.AdagradOptimizer(Lr)
		train = optimizer.minimize(loss, var_list=tf.trainable_variables())

training_info.update_binding	(“W”, “New W”), (“H”, “New H”)	**N/A**

Investigation needed…

- How to handle pseudo nodes like MyInferenceFunction?
- How to determine use of constants vs variables?
- How to support additional operators in training graph (for ex. add or subtract between gradients in algorithm)?
- Where to apply update binding?