TensorFlow: A System for Large-Scale Machine Learning

By Martín Abadi et. al (2016)

Presentation by Luou Wen (lw658)
Background

• Motivation:
  • Improvement to DistBelief for large-scale distributed computing
  • DistBelief: Parameter server architecture – stateless worker, stateful parameter server
  • Also allow training and using models on smaller scale machines (single GPU machines and mobile CPUs)
  • More flexibility to model training
Related works

• DistBelief:
  • Limitations of parameter server architecture
  • Lack of flexibility to refine optimisation functions
  • Fixed execution pattern – fails for more advanced models

• Single-machine frameworks:
  • Caffe – difficult to add new layers
  • Theano – similar structure
  • Torch – less portable

• Batch dataflow systems:
  • Require data to be immutable
  • Update step must process larger batches slowing convergence
Approach

• Model represents individual mathematical operators
• Deferred execution
  • 1\textsuperscript{st} phase defines program as a symbolic dataflow graph
  • 2\textsuperscript{nd} phase executes an optimised version of the program
Evaluation

• Similar performance to MXNet
• Neon and Caffe optimised differently
• Torch and TensorFlow use the same version of cuDNN
• Evaluation of Language Modelling not compared against other systems
Strength and Weaknesses

• Strength:
  • Distributable
  • Optimised for large-scale model training

• Weaknesses:
  • Static dataflow graph implementation limits training of deep reinforcement learning algorithms
    • PyTorch seems to be the more popular tool for this
Impact

• Widely adopted in the industry for machine learning engineering
• Used in many research projects
References