Scaling Machine Learning

Section 1 – Introduction and Overview

- Machine learning (ML) – Statistical learning
- Artificial intelligence (AI)
- Deep learning (DL)
- Reinforcement learning (RL)
- Data science (DS)
- Big data (BD)
- Data mining (DM)
- Data analytics (DA)
- Business intelligence (BI)
- Predictive analytics (PA)
- Statistical analysis (SA)
- Pattern recognition (PR)
- Knowledge discovery in databases (KDD)

Author Team: (Institution Location)

- Author 1 (Institution 1)
- Author 2 (Institution 2)
- Author 3 (Institution 3)

Section Overall Goals

- Introduce the topic area
- Motivate the topic area
- Provide a brief overview of the topic area
- Provide a brief overview of the tutorial
- Provide a brief overview of the tutorial

Section 1 – Introduction and Overview

- Section 1.1 – Introduction
- Section 1.2 – Overview
- Section 1.3 – Conclusion

Section 1.1 – Introduction

- Machine learning (ML)
  - Statistical learning
  - Artificial intelligence (AI)
  - Deep learning (DL)
  - Reinforcement learning (RL)
  - Data science (DS)
  - Big data (BD)
  - Data mining (DM)
  - Data analytics (DA)
  - Business intelligence (BI)
  - Predictive analytics (PA)
  - Statistical analysis (SA)
  - Pattern recognition (PR)
  - Knowledge discovery in databases (KDD)
- Scope of ML
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning
- Applications of ML that can be scaled
  - Computer vision (CV) – Image classification
  - Speech recognition (SR)
  - Natural language processing (NLP)
  - Text mining
  - Recommender systems
  - Ad targeting
  - Search ranking
  - Gene sequence
  - Digital assistants (Siri, Alexa, etc.)
- Scaling ML
  - Model parallelism
  - Data parallelism
  - Distributed stochastic gradient descent (SGD)
- Open ML challenges
  - Ease of use
  - Hyper-parameter optimization (HPO) and neural architecture search (NAS)
  - Ease of deployment
  - Explainable AI (XAI)
  - Fairness, accountability, transparency, and ethics (FATE)
  - Hardware challenges

Section 1.2 – Overview

- Tutorial Goals and Non-goals
- Tutorial Materials
  - Slides
  - Hands-on exercises (Google Colaboratory)
- Tutorial Organization

Section 1.3 – Conclusion

- Hands-on exercises (Google Colaboratory)
- Open discussion

Section 2 – Hardware Trends Impacting Scaling

- Introduction
- Single Machine Trends
- Distributed Hardware Trends
- Hardware Acceleration Trends
- Summary and Conclusions

Section 2.1 – Introduction

- Moore's Law
- Dennard scaling
- Amdahl's law
- Parallel computing
- Warehouse scale computing (WSC)
- Cloud computing

Section 2.2 – Single Machine Trends

- Pipelining
- Instruction level parallelism (ILP)
- Superscalar
- Out-of-order execution (OOO)
- Speculative execution
- Vectorized instructions
- Multi-core
- Simultaneous multithreading (SMT)
- Non-uniform memory access (NUMA)
- Memory hierarchy
- Storage hierarchy (disk, flash, 3D-Xpoint)
- Network
- Impact on software performance

Section 2.3 – Distributed Hardware Trends

- Ethernet
- Infiniband
- Remote direct memory access (RDMA)
- Storage area network (SAN)
- Network attached storage (NAS)
- Impact on distributed machine learning (DML)

Section 2.4 – Hardware Acceleration Trends

- Graphics processing units (GPU)
- Tensor processing units (TPU)
- Field programmable gate arrays (FPGA)
- Neuromorphic computing
- Quantum computing
- Impact on machine learning (ML)

Section 2.5 – Summary and Conclusions

- Hardware trends
- Implications for ML/DL/DS/BD/AI
- Implications for software

Section 3 – Scaling Machine Learning

- Introduction
- Single machine scaling
- Distributed scaling
- Hardware acceleration
- Open challenges
- Conclusion

Section 3.1 – Introduction

- Why scale machine learning (ML)
  - Faster time-to-solution (TTS)
  - Larger datasets
  - Larger models
  - Higher cluster utilization
- Ease of use
- Ease of deployment
- Ease of maintenance
- Hardware trends
- Software trends
- Scaling ML
  - Model parallelism
  - Data parallelism
  - Distributed stochastic gradient descent (SGD)
- Open challenges
  - Ease of use
  - Hyper-parameter optimization (HPO) and neural architecture search (NAS)
  - Ease of deployment
  - Explainable AI (XAI)
  - Fairness, accountability, transparency, and ethics (FATE)
  - Hardware challenges

Section 3.2 – Single Machine Scaling

- Pipelining
- Instruction level parallelism (ILP)
- Superscalar
- Out-of-order execution (OOO)
- Speculative execution
- Vectorized instructions
- Multi-core
- Simultaneous multithreading (SMT)
- Non-uniform memory access (NUMA)
- Memory hierarchy
- Storage hierarchy (disk, flash, 3D-Xpoint)
- Network
- Impact on software performance

Section 3.3 – Distributed Scaling

- Ethernet
- Infiniband
- Remote direct memory access (RDMA)
- Storage area network (SAN)
- Network attached storage (NAS)
- Impact on distributed machine learning (DML)

Section 3.4 – Hardware Acceleration

- Graphics processing units (GPU)
- Tensor processing units (TPU)
- Field programmable gate arrays (FPGA)
- Neuromorphic computing
- Quantum computing
- Impact on machine learning (ML)

Section 3.5 – Open Challenges

- Ease of use
- Hyper-parameter optimization (HPO) and neural architecture search (NAS)
- Ease of deployment
- Explainable AI (XAI)
- Fairness, accountability, transparency, and ethics (FATE)
- Hardware challenges

Section 3.6 – Conclusion

- Open discussion

Section 4 – Hands-on Exercises

- Introduction
- Hyper-parameter optimization (HPO) and neural architecture search (NAS)
- Ease of deployment
- Explainable AI (XAI)
- Fairness, accountability, transparency, and ethics (FATE)
- Open discussion

Section 4.1 – Introduction

- Hands-on exercises (Google Colaboratory)
- Jupyter notebooks
- Open source machine learning libraries
- Open source datasets

Section 4.2 – Hyper-parameter Optimization (HPO) and Neural Architecture Search (NAS)

- Hyper-parameter optimization (HPO)
- Neural architecture search (NAS)
- Frameworks
- Software libraries
- Open source datasets

Section 4.3 – Ease of Deployment

- Frameworks
- Software libraries
- Open source datasets

Section 4.4 – Explainable AI (XAI)

- Frameworks
- Software libraries
- Open source datasets

Section 4.5 – Fairness, Accountability, Transparency, and Ethics (FATE)

- Frameworks
- Software libraries
- Open source datasets

Section 4.6 – Open Discussion

- Open discussion

Section 5 – Conclusion

- Open discussion

Section 6 – Bibliography

- Bibliography

Section 7 – Appendix

- Machine learning (ML)
  - Statistical learning
  - Artificial intelligence (AI)
  - Deep learning (DL)
  - Reinforcement learning (RL)
  - Data science (DS)
  - Big data (BD)
  - Data mining (DM)
  - Data analytics (DA)
  - Business intelligence (BI)
  - Predictive analytics (PA)
  - Statistical analysis (SA)
  - Pattern recognition (PR)
  - Knowledge discovery in databases (KDD)
- Scope of ML
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning
- Applications of ML that can be scaled
  - Computer vision (CV) – Image classification
  - Speech recognition (SR)
  - Natural language processing (NLP)
  - Text mining
  - Recommender systems
  - Ad targeting
  - Search ranking
  - Gene sequence
  - Digital assistants (Siri, Alexa, etc.)
- Scaling ML
  - Model parallelism
  - Data parallelism
  - Distributed stochastic gradient descent (SGD)
- Open ML challenges
  - Ease of use
  - Hyper-parameter optimization (HPO) and neural architecture search (NAS)
  - Ease of deployment
  - Explainable AI (XAI)
  - Fairness, accountability, transparency, and ethics (FATE)
  - Hardware challenges
- Hardware trends
- Implications for ML/DL/DS/BD/AI
- Implications for software

Section 7.1 – Machine Learning (ML)

- Statistical learning
- Artificial intelligence (AI)
- Deep learning (DL)
- Reinforcement learning (RL)
- Data science (DS)
- Big data (BD)
- Data mining (DM)
- Data analytics (DA)
- Business intelligence (BI)
- Predictive analytics (PA)
- Statistical analysis (SA)
- Pattern recognition (PR)
- Knowledge discovery in databases (KDD)

Section 7.2 – Scope of ML

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

Section 7.3 – Applications of ML that can be scaled

- Computer vision (CV) – Image classification
- Speech recognition (SR)
- Natural language processing (NLP)
- Text mining
- Recommender systems
- Ad targeting
- Search ranking
- Gene sequence
- Digital assistants (Siri, Alexa, etc.)

Section 7.4 – Scaling ML

- Model parallelism
- Data parallelism
- Distributed stochastic gradient descent (SGD)

Section 7.5 – Open ML Challenges

- Ease of use
- Hyper-parameter optimization (HPO) and neural architecture search (NAS)
- Ease of deployment
- Explainable AI (XAI)
- Fairness, accountability, transparency, and ethics (FATE)
- Hardware challenges

Section 7.6 – Hardware Trends

- Moore's Law
- Dennard scaling
- Amdahl's law
- Parallel computing
- Warehouse scale computing (WSC)
- Cloud computing

Section 7.7 – Implications for ML/DL/DS/BD/AI

- Hardware trends
- Software trends

Section 7.8 – Implications for Software

- Hardware trends
- Software trends

Section 8 – Bibliography

- Bibliography

Section 9 – Acknowledgments

- Acknowledgments

Section 10 – Biography

- Biography

Section 11 – Machine Learning Software

- Frameworks
- Software libraries
- Open source datasets

Section 11.1 – Frameworks

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin
- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin

Section 11.2 – Software Libraries

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin

Section 11.3 – Open Source Datasets

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin

Section 12 – Machine Learning Software

- Frameworks
- Software libraries
- Open source datasets

Section 12.1 – Frameworks

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin

Section 12.2 – Software Libraries

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin

Section 12.3 – Open Source Datasets

- Caffe
- Caffe2
- Chainer
- CNTK
- DeepLearning4J
- Dlib
- DyNet
- Flux
- Gluon
- Keras
- Keras-MX
- Lasagne
- MatlabDL
- MatlabML
- MXNet
- PaddlePaddle
- PlaidML
- PyTorch
- Scikit-learn
- Tensorflow
- Theano
- Torch
- Vowpal Wabbit (VW)
- H2O
- Thrust
- CUB
- ArrayFire
- Boost.Compute
- CLTorch
- TensorFlow.js
- deeplearning.js
- Julia
- Clojure
- Scala
- Spark MLlib
- Spark ML
- Hadoop MR
- Spark DL
- Flink ML
- Angel
- Parameter Server
- Veles
- Spark CoreNLP
- Ecrans
- Tensorspace
- Weka
- Keras.js
- Stanford CoreNLP
- Prophet
- Zhinu
- Zoo
- Marvin