Towards provably efficient Quantum Algorithms for large-scale machine-learning models

Authors: Junyu Liu, ..., Liang Jiang (In pre-print:2023)

Manish Kumar

Quantum Tech. (M. Tech) IISc Bengaluru

October 28, 2023

Table of Contents

Table of Contents

How the Quantum algorithm is used in ML setting?

Maching Learning Set-up

- Learning paradigm:Classical neural network (classical data)
- Network training: Gradient-based [say, Stochastic Gradient Descent (SGD)]
- Network constraint: Network shall admit sparse training. This is achieved via model (parameter) pruning

Quantum algorithm used in the above set-up

- Training via SGD amounts to solving ordinary differential equations(ODE) in an iterative fashion
- If the network possesses certain requirements, a Quantum advantage is possible in the training part of the network
- Why Quantum advantage exist?: Detailed on the next slide
- How much advantage?: Superpolynomial

Rationale for Quantum advantage

How Network training and ODE solver are related?

- Case I: A dissipative nonlinear ODE has an efficient quantum algorithm for certain specified cases. [J-P Liu, ..., AM Childs 2021]
- Case II: Dissipation is observed in the early stage of the network training. Loss functions are known to be non-linear.
- The Rest of the paper is built upon when the results of case I could be used to get a quantum algorithm for case II.

How much speed up would be possible?

- If certain criteria are met, then Theorem 1 in the paper implies a Quantum algorithm with a runtime of:

$$\mathcal{O}[T \cdot poly(log(n), \frac{1}{\epsilon})]$$

- $T \rightarrow$ number of iterations; $n \rightarrow$ size of model parameters
- Classical algorithms have runtime $poly(n)$

Requirements to be met by Machine Learning model

Necessary condition for Theorem 1

- Sparsity of the ML model: Some practically useful ML models has this property.
- Sparsity of the weight vectors: Possible for sparse training (model pruning).
- Dissipative assumptions: Need to be checked by analysis of the eigenvalues of Hessian.
- Complexity of uploading/downloading quantum states: There exists an efficient methods for sparse quantum states.

Important Note:

Claim: The above requirement creates the opportunity to apply HHL-based algorithm in this dissipative non-linear system.

Learning process diagram

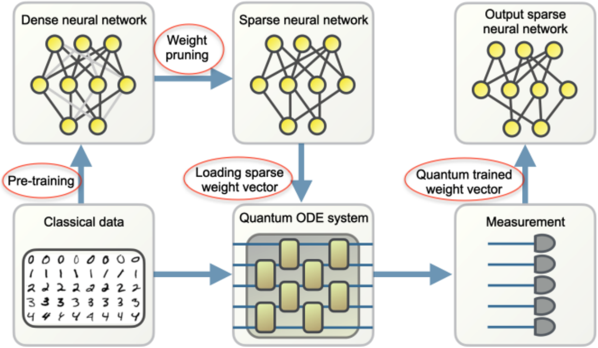

Figure: A possible learning process in large-scale models, which might use sparse training, whose early stage in learning might admit possible quantum advantage.

Table of Contents

Stochastic Gradient Descent

Key parameters of SGD for a Machine Learning Model

- Let the model be defined by loss function $\mathcal{L}_{\mathcal{A}}$
- Weight vector be $\theta \in \mathbb{R}^n$ and its componets be θ_μ
- Steps in model training be $t = 0, ..., T$
- Iteration equation: $\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{d\mathcal{L}_{\mathcal{A}}(\theta(t))}{d\theta}$
- The above eqation is iteratively solved as per the prescription of SGD

What makes it non-linear ODE?

- Loss function $\mathcal{L}_{\mathcal{A}}$ has non-linearity due to construction of the model. Example: Cross entropy loss function.
- Hence, it must be linearized to make any Quantum method applicable.

Structure of the algorithmic steps for SGD

Input:

- Input: Initial weight vector $\theta(t=0)$
- Machine Learning Architecture $\mathcal{L}_{\mathcal{A}}$ with size n
- Maximal number of iteration T

Output:

- Output: terminal weight vector $\theta(t=T)$.

Key challenge to translate from Classical to Quantum domain

- Uploading and downloading sparse vector efficiently
- Effectiveness of linearization of non-linear loss function

Quantum Algorithm for the problem

Algorithmic steps

- Step-I: Convert the non-linear equation to the system of linear equations. Method: Use Quantum Carleman linearization.
- Step-II: Upload sparse weight vector $\theta(t = 0)$ as state vector to Qauntum device.[*]
- Step-III: Perform Quantum linear system solver to produce the quantum state vector $\theta(t = T)$.
- Step-IV: The last step yields a system of linear equations (SLE). Check if SLE fulfils the requirement for an efficient algorithm for Quantum linear system Algorithms (QLSA).
- Step-V: Use Quantum State Tomography to get (classical) weight vector $\theta(T)$.[*]

[*] See the caveat on the next page

Summary of requirements and a few caveats

Requirements for Quantum advantage

- The requirements of Theorem 1 is necessary to gain the Quantum advantage
- The solution shall not decay "too rapidly". (Quantified in the paper)

Important Caveats

- The algorithm produces a state vector that encodes the desired solution without specifying how to extract information from it.
- This is a common caveat to QSLA-based algorithms. In many cases, this is used as a subroutine to other quantum algorithms that require such a state in their intermediate process.

Table of Contents

Conclusions

- If certain requirement are fulfilled by an (classical) ML model, then Quantum can speedup its training
- Carleman linearization of nonlinear ODE has been used as the main crucial tool
- This is in line with using Quantum machines to simulate classical phenomena rather than the Qauntum phenomena. The former is known to have a Quantum advantage over any (known) classical computation.
- This requires fault tolerant Quantum machine to run HHL-based technique
- It is most useful in the early stages of the Model training. Because during those time the requirements are met with high certainty.

Thank You for Your Attention!