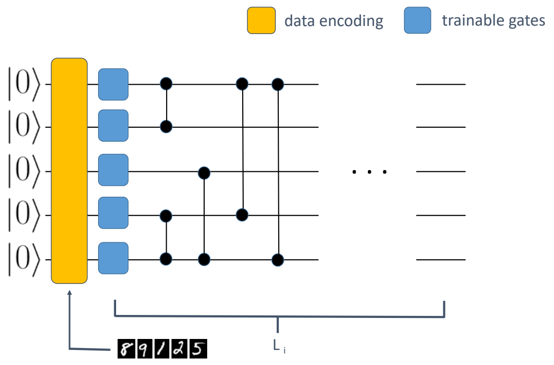
Layerwise learning for quantum neural networks

Andrea Skolik[1,2], Jarrod R. McClean[3], Masoud Mohseni[3], Patrick van der Smagt[4], Martin Leib[1]

[1]Volkswagen Data:Lab, Ungererstraße 69, 80805 Munich, Germany
[2]Ludwig Maximilian University, Faculty of Mathematics, Informatics and Statistics, Theresienstraße 39, 80333 Munich, Germany
[3]Google Research, 340 Main Street, Venice, CA 90291, USA
[4]Volkswagen Group Machine Learning Research Lab, Ungererstraße 69, 80805 Munich, Germany

Quantum neural networks

Quantum neural networks (QNNs) are parametrized circuits that are optimized to accomplish a learning task. The trained parameters are the rotation angles of the single-qubit gates.



- One layer consists of single-qubit rotation gates and arbitrary combinations of CZ gates on all qubits, e.g. 2D or all-to-all.
- A classical gradient-based optimizer is used to calculate parameter updates based on a loss function that is calculated over the outputs of the QNN.

Barren plateaus in QNNs

Completely random circuits will converge to a 2-design as a function of the number of qubits and the circuit depth (McClean et al. (2018)). This causes concentration of measure on the average, which in turn leads to vanishing variance of gradients w.r.t. circuit parameters.

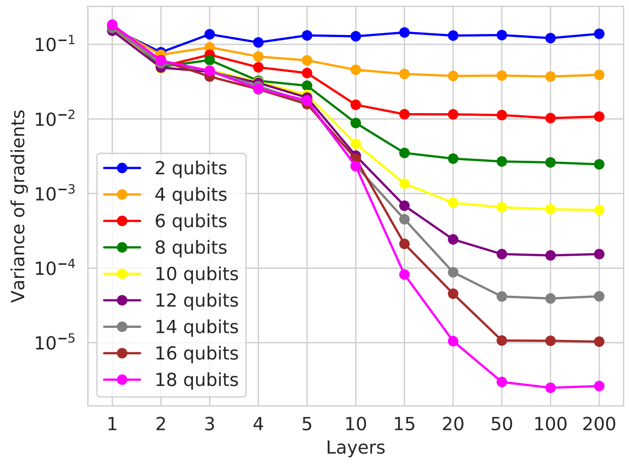


Figure 1: Convergence to a 2-design for randomly initialized circuits of different size and all-to-all connections.

- When training on a near term quantum device, signal-to-noise ratio will prevent efficiently training QNNs as the gradient components get smaller and smaller in magnitude.
- The smaller the expectation value we want to estimate, the larger the number of measurements we need to achieve a certain accuracy.

References

Knill, E., Ortiz, G., and Somma, R. D. (2007). Optimal quantum measurements of expectation values of observables. *Physical Review A*, 75(1):012328.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812.

Gradient sampling

- Estimation of expectation values scales in number of measurements N as $\mathcal{O}(\frac{1}{\epsilon^2})$ with error ϵ on noisy intermediate-scale quantum (NISQ) devices (Knill et al. (2007))
- For getting useful estimates of gradient components g_i, we need $g_i > \epsilon$
- Magnitude of partial derivatives determines how many measurements are needed to achieve good signal-to-noise ratio for classical optimizer
- The larger $|g_i|$, the lower we can go with the number of measurements from the quantum device

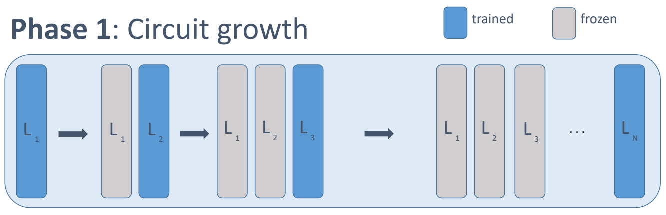
Layerwise learning

Shallow circuits are not affected by barren plateaus. We use this property to more efficiently train QNNs. We do this by training them layer-by-layer, only optimizing part of the parameters in the circuit in one step. With this approach, we

- avoid initialization on a barren plateau
- reduce the number of trained parameters
- reduce the number of samples from the QPU per gradient estimation.

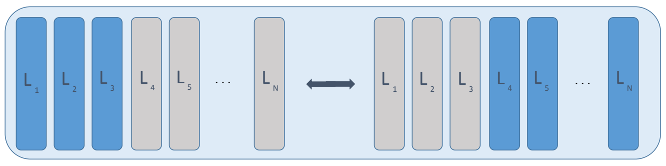
Training is divided into two phases. In the first phase, we start training with a small number of layers and train those for a fixed number of epochs. After that, we add another set of layers and freeze the parameters of the previous step's layers. We repeat this process until the desired depth is reached. In phase two, we perform additional optimization sweeps over larger subsets of the layers using the final circuit configuration from phase one. The parameters from this circuit give us a good starting point to optimize quarters, halves, or even the full circuit without initializing on a barren plateau.

Phase 1: Circuit growth



- start training with a small number of layers
- train initial layers for a given number of epochs
- add another set of layers
- freeze previous layers' parameters
- train new set of layers
- repeat until desired depth is reached

Phase 2: Circuit refinement



- take trained circuit from phase 1
- divide into larger partitions, i.e. two halves, or even the full circuit
- perform additional optimization sweeps over the new partitions

Binary classification

We demonstrate our approach on a binary classification task on handwritten digits, namely the 6's and 9's in the MNIST data set. Using layerwise learning, we can decrease the number of measurements taken to estimate each gradient component to as low as 10 and still achieve favorable training results. Complete depth learning in this scenario requires much smaller learning rates to successfully learn, leading to a slower convergence rate. As a naive strategy to avoid initialization on a barren plateau, we compare our approach to training the full circuit with an initialization of all-zero parameters. We train the same randomly generated circuit with LL and CDL strategies, which consists of 8 qubits, 21 layers and all-to-all connections between qubits. As shown in figure 1, this is a configuration that has already entered the 2-design regime when randomly initialized. We perform a measurement on the last qubit and use its expectation value as the classification result. The below results show those achieved by the best hyperparameters for both approaches.

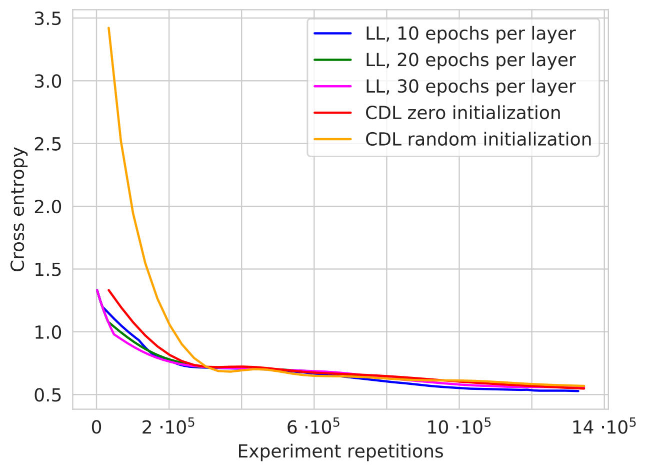


Figure 2: Results for exact gradients computed directly from the circuit's final state corresponding to infinite number of samples.

- all training strategies perform similarly, as low magnitude gradients can be handled by classical optimizer
- both outperform random initialization

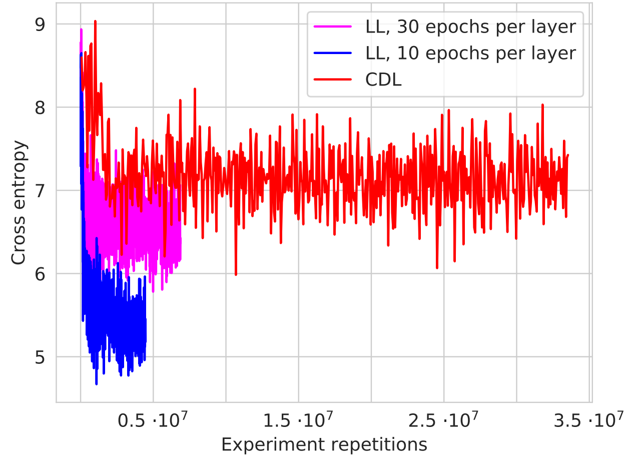


Figure 3: Average test error for training with 10 samples per expectation value estimation for partial derivatives.

- gradients of larger magnitude let LL converge faster and to lower objective function values on average
- larger gradients reduce the variance of measurement uncertainty, and allow us to use higher learning rates to further speeds up the learning process
- the smaller the variance induced by measurement uncertainty, the fewer measurements from the QPU are needed to get sufficient gradient estimates