Quantum Vision Transformer:
a study on the Quantum Orthogonal Transformer

Università degli Studi di Firenze

Intelligenza Artificiale

Quantum Machine Learning

Presented by Giovanni Maccioni

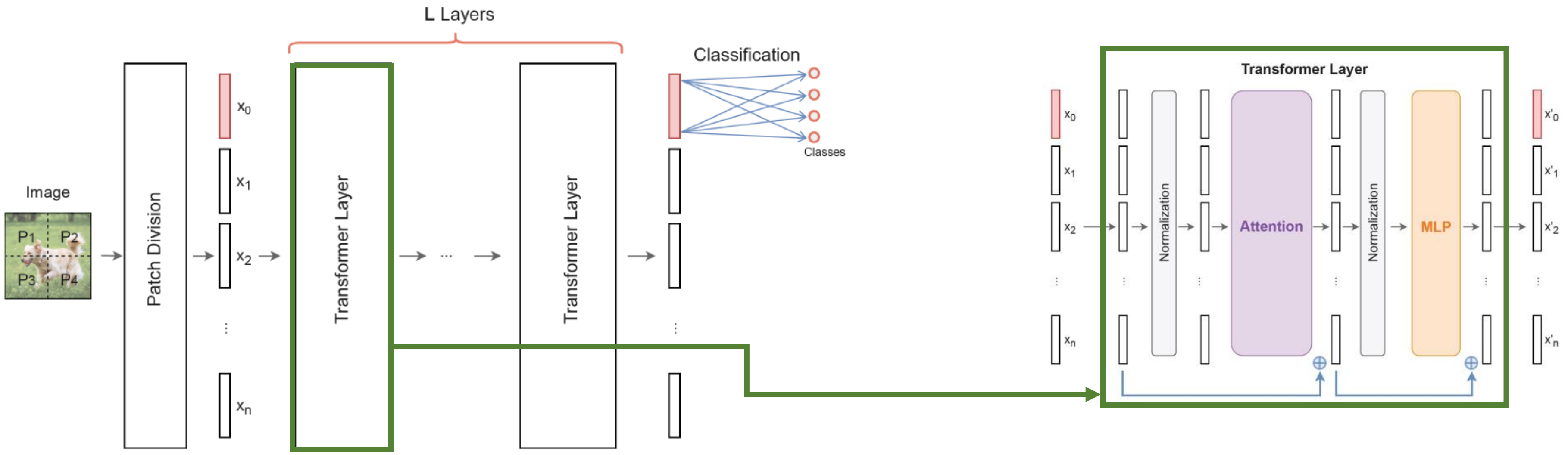
Introduction

- This work is a study on Cherrat et al. [1];

- They developed different versions of Quantum Vision Transformers;

- In particular in this project we reimplemented the Quantum Orthogonal Transformer, in which the attention is partially computed with quantum circuits;

- An image is divided into patches
- Each patch is embedded into a vector, forming a set
- A class token is added to the set of vectors
- The set of vectors is the input to the Transformer



Attention

- Let’s consider a set, where each of its elements have a **Value** and we can access them by consulting a corresponding **Key**.

- Let’s say we have a request, a **Query**, to retrieve an element, under that request conditions, from this set.

- Better the element answers to the request, higher the **similiarity** between the Query and the Key will be.

- We can see those similarity scores as weights, and the answer to the Query as a **weighted average** of the values of each element in the set.

Attention

- Our input set is the embedded image patches, plus the class token

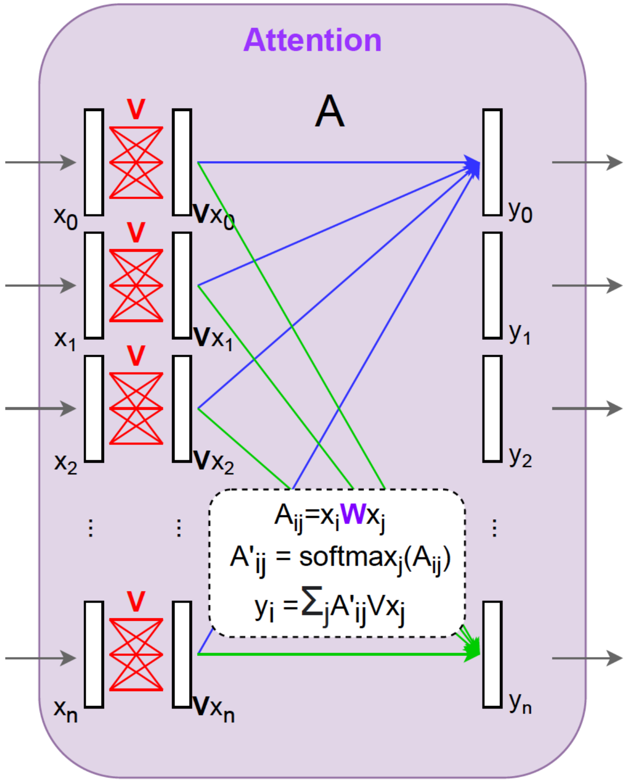
$$Attention(Q,K,U) = softmax(QK^T)U$$

$$softmax(QW^Q(KW^K)^T)UV$$

$$softmax(QWK^T)UV$$

$$softmax(XWX^T)XV$$

$\boxed{Q,K,U \in \mathbb{R}^{n+1\times d}}$ $\boxed{W^Q,W^K,V \in \mathbb{R}^{dXd}}$



Why quantum

- The complexity of classical attention computation is:

$$O(n^2 d + nd^2)$$

- Using quantum circuit we can lower the complexity, measured in number of parametrised gates;

- Also we can lower the number of parameters to train, that in classic attention is:

$$O(2d^2)$$

Quantum Machine Learning

- Parametrised quantum circuit, in which we have two group of parameters: the **inputs** and the **weights**;

- These type of circuits are usually composed by two sub-circuits:

  - **Feature Map**: Vector Loader

  - **Ansatz**: Orthogonal Layer

- For the inputs we need an encoding strategy, to load classical data into a quantum circuit;

Unary Amplitude Encoding

- Given a vector, x, the unary amplitude encoding is given by:

$$x = (x_0, \cdots, x_{d-1}) \qquad |x\rangle = \frac{1}{\|x\|} \sum_{i=0}^{d-1} x_i \, |e_i\rangle$$

- This choice would require in general $n = \lceil log(d) \rceil$ qubits to encode x. The choice of the basic component for the circuits ahead will require d qubits circuits instead;

Gradient Computation

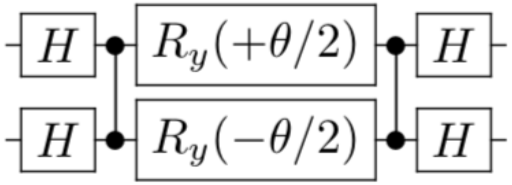
- We need an algorithm to estimate the derivatives to proceed with the training procedure;
- We will use the **Parameter Shift Rule**;
- For these circuits there is a specific algorithm in [4];

RBS gate

- The RBS gate is the foundation for all the circuits ww need

$$RBS(\theta) : \begin{cases} |01\rangle \mapsto \cos\theta\,|01\rangle - \sin\theta\,|10\rangle \\ |10\rangle \mapsto \sin\theta\,|01\rangle + \cos\theta\,|10\rangle \end{cases}$$

- A possible decomposition:



- It has the property to preserve the number of ones and zeros in any basis state;

Vector Loader

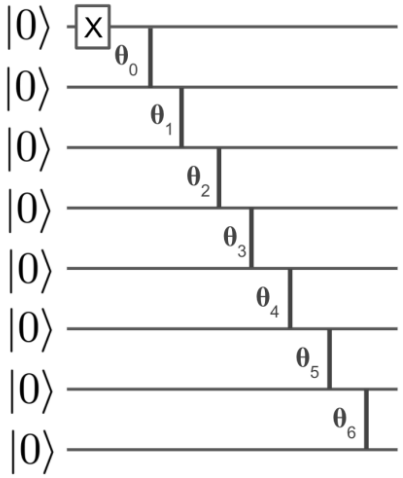
- It will be our Feature Map;

- It is a circuit composed of always d-1 RBS gates, but with different topologies;

- The parameters are computed classically and the algorithm is defined by the disposition of the gates;

Vector Loader: Diagonal Loader

- Assumes Nearest-Neighbour connectivity between qubits
- The depth is **O(d)**



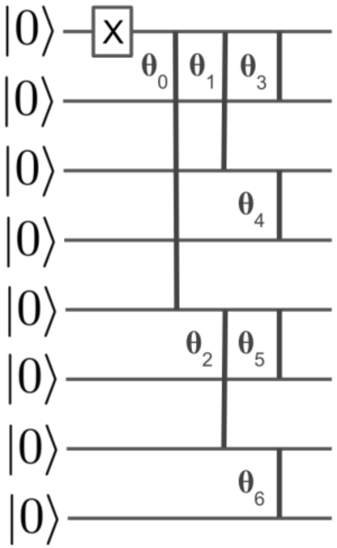
Algorithm 1: Compute parameters for the diagonal vector loader

Data: $x = (x_0, \cdots, x_{d-1})$
Result: $\theta_0, \cdots, \theta_{d-2}$
$\theta_0 \leftarrow \arccos(x_0)$;
$i \leftarrow 1$;
while $i \neq d-1$ **do**
  $\theta_i \leftarrow \arccos\left(\frac{x_i}{\prod_{j=0}^{i} \sin(\theta_j)}\right)$;
  $i \leftarrow i + 1$;
end

$$RBS(\theta) : \begin{cases} |01\rangle \mapsto \cos\theta\,|01\rangle - \sin\theta\,|10\rangle \\ |10\rangle \mapsto \sin\theta\,|01\rangle + \cos\theta\,|10\rangle \end{cases}$$

- Assumes All-to-All connectivity between qubits;

- It has a depth of **O(log d)**



Algorithm 2: Compute parameters for the parallel vector loader

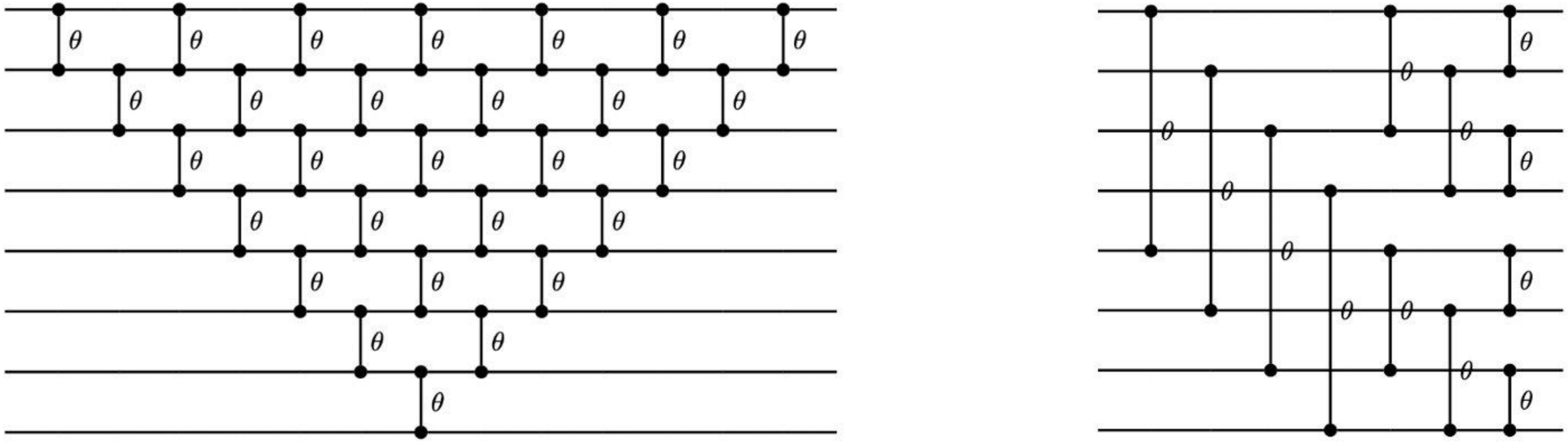
Data: $x = (x_0, \cdots, x_{d-1})$
Result: $\theta_0, \cdots, \theta_{d-2}$
$i \leftarrow 0$;
while $i \neq \frac{d}{2}$ **do**
  $r_{\frac{d}{2}+i-1} \leftarrow \sqrt{{x_{2i+1}}^2 + {x_{2i}}^2}$;
  $i \leftarrow i + 1$;
end
$i \leftarrow \frac{d}{2} - 2$;
while $i \geq 0$ **do**
  $r_i \leftarrow \sqrt{{r_{2i+2}}^2 + {r_{2i+1}}^2}$;
  $i \leftarrow i - 1$;
end
$i \leftarrow 0$;
while $i \neq \frac{d}{2}$ **do**
  $\theta_i \leftarrow \arccos(\frac{r_{2i+1}}{r_i})$;
  $i \leftarrow i + 1$;
end
$i \leftarrow 0$;
while $i \neq \frac{d}{2}$ **do**
  if x_{2i+1} *is positive* **then**
    $\theta_{\frac{d}{2}+i-1} \leftarrow \arccos(\frac{x_{2i}}{r_{\frac{d}{2}+i-1}})$;
  else
    $\theta_{\frac{d}{2}+i-1} \leftarrow 2\pi - \arccos(\frac{x_{2i}}{r_{\frac{d}{2}+i-1}})$;
  end
end

Quantum Orthogonal Layer

- This will be our Ansatz;
- The number of parametrised gates, differently from the Vector Loaders, varies;
- Also the topology of the circuit does;

Quantum Orthogonal Layer: Pyramid and Butterfly Layers

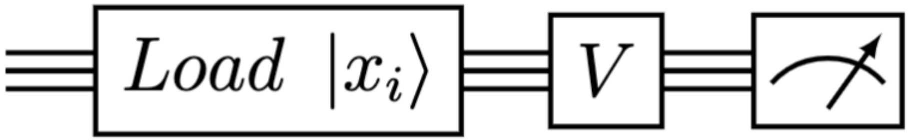
- On the left, the Pyramid Layer; the Butterfly Layer on the right;



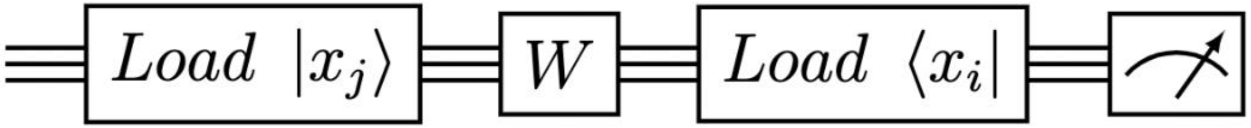
Circuit	Hardware Connectivity	Depth	# Gates
Pyramid	NN	$2d-3$	$\frac{d(d-1)}{2}$
Butterfly	All-to-all	$log(d)$	$\frac{d}{2}log(d)$

Quantum Attention Block

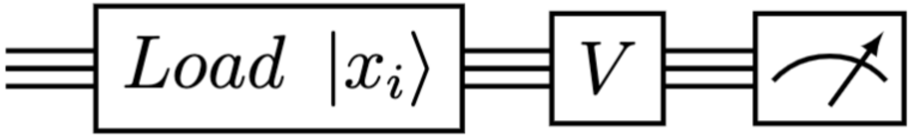
- Matrix-Vector Product



- Vector-Matrix-Vector Product

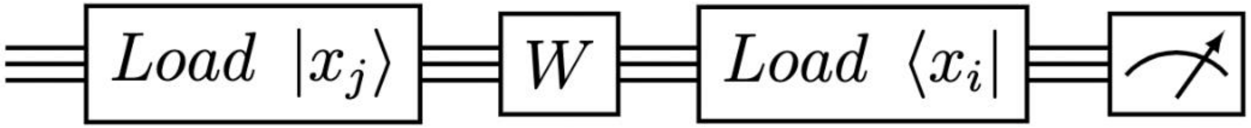


Matrix-Vector Product



- Measuring this circuit will give us the Matrix-Vector prooduct
- We will need **n circuits**
- The amplitudes of only states with only one qubit different from 0 will be potentially non-zero (thanks to the RBS gate property)

Vector-Matrix-Vector Product



- We need the Adjoint Vector Loader circuit;
- The probability of measuring 1 in the first qubit will be:

$$A_{ij}{}^2 = |x_i W x_j|^2$$

- We worked with positive attention coefficients;
- We need n^2 circuits;

On the complexity

- As a reminder the complexity of the classical attention computation complexity and the number of parameters are:

$$O(n^2 d + nd^2) \qquad\qquad O(2d^2)$$

- Taking the best case scenario in the quantum implementation:

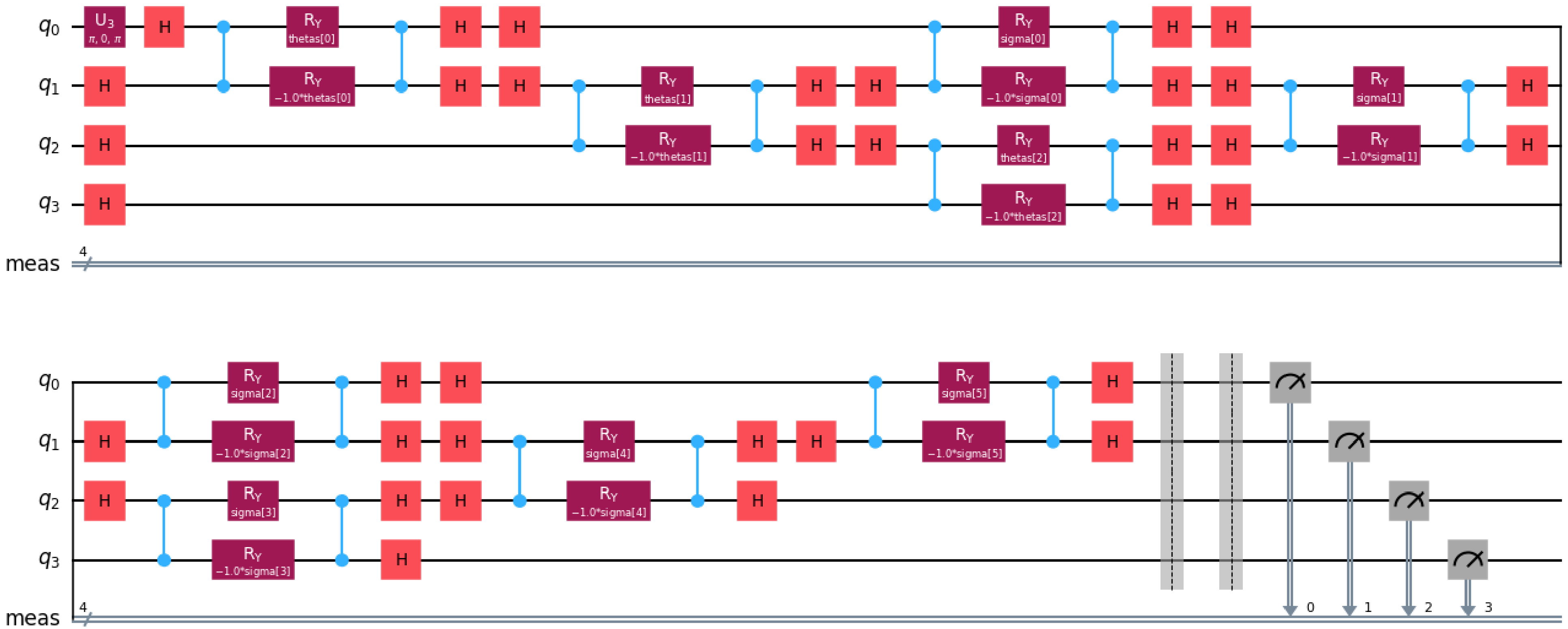
Number of parametrized Gates	Depth	Number of circuits
$O(dlog(d))$	$O(log(d))$	$n + n^2$

- Also there is an overhead to load classical data into quantum circuits;

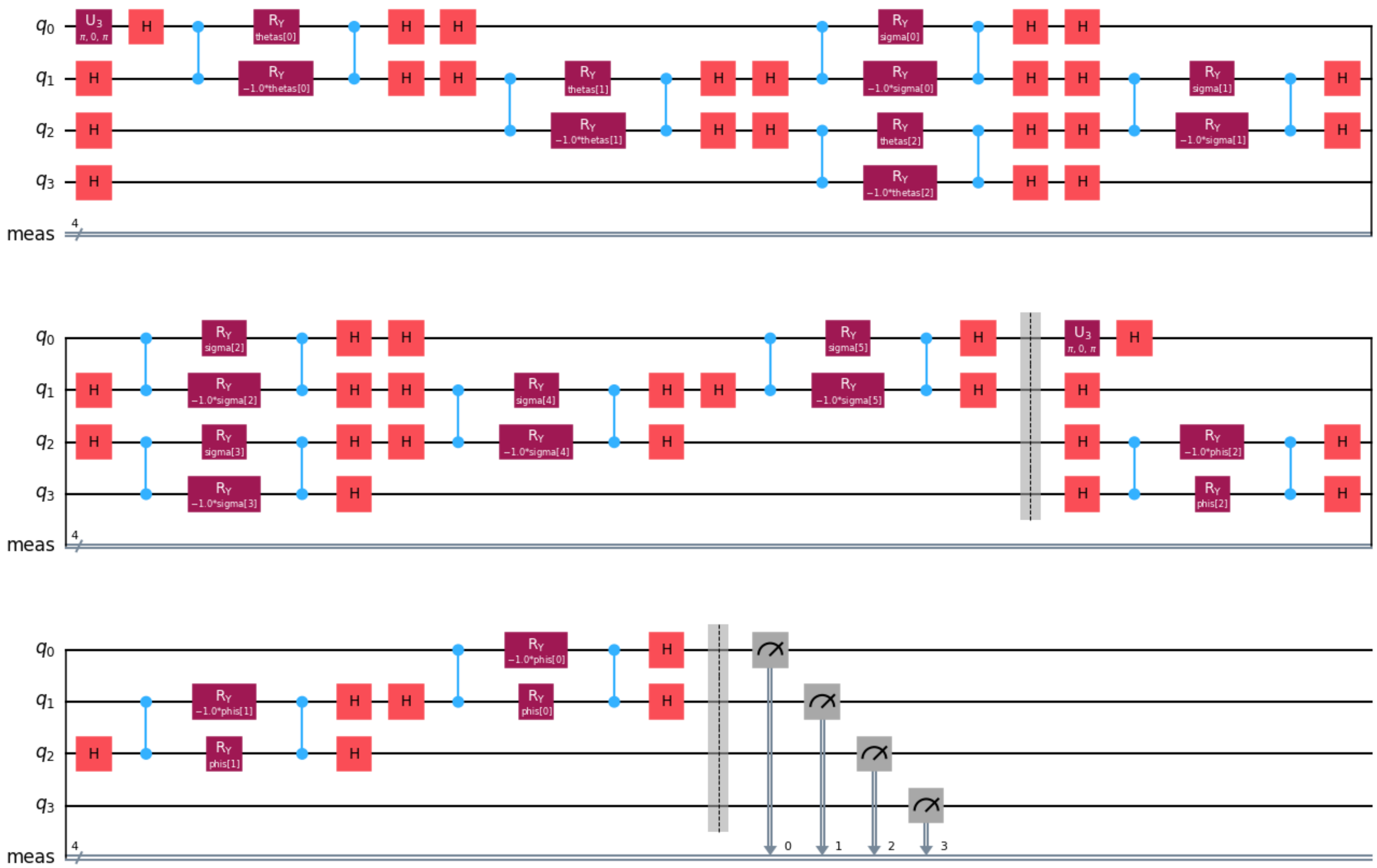
Experiments

- The dataset choosen is MNIST, limited to classes 0 and 1 (about 10.000 samples);
- The two Vision Transformers were trained for 5 epochs;
- The images are divided in 4 patches 14x14, giving us 4 vectors per-image as input to the Transformer;
- The image patches are embedded in 4-dimensional vectors, to allow us using 4-qubit circuits;
- We trained two ViT, each took on average about 30 hours to train;

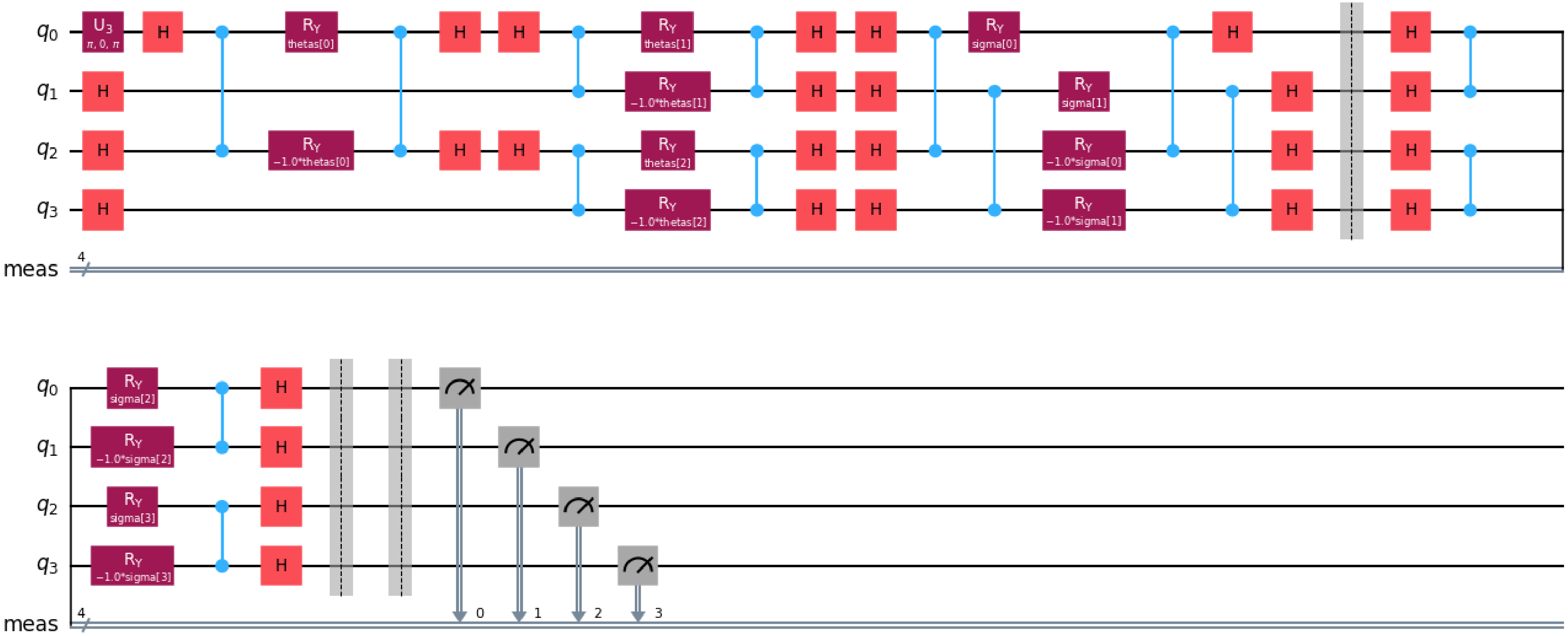
Implementation: Diagonal-Pyramid



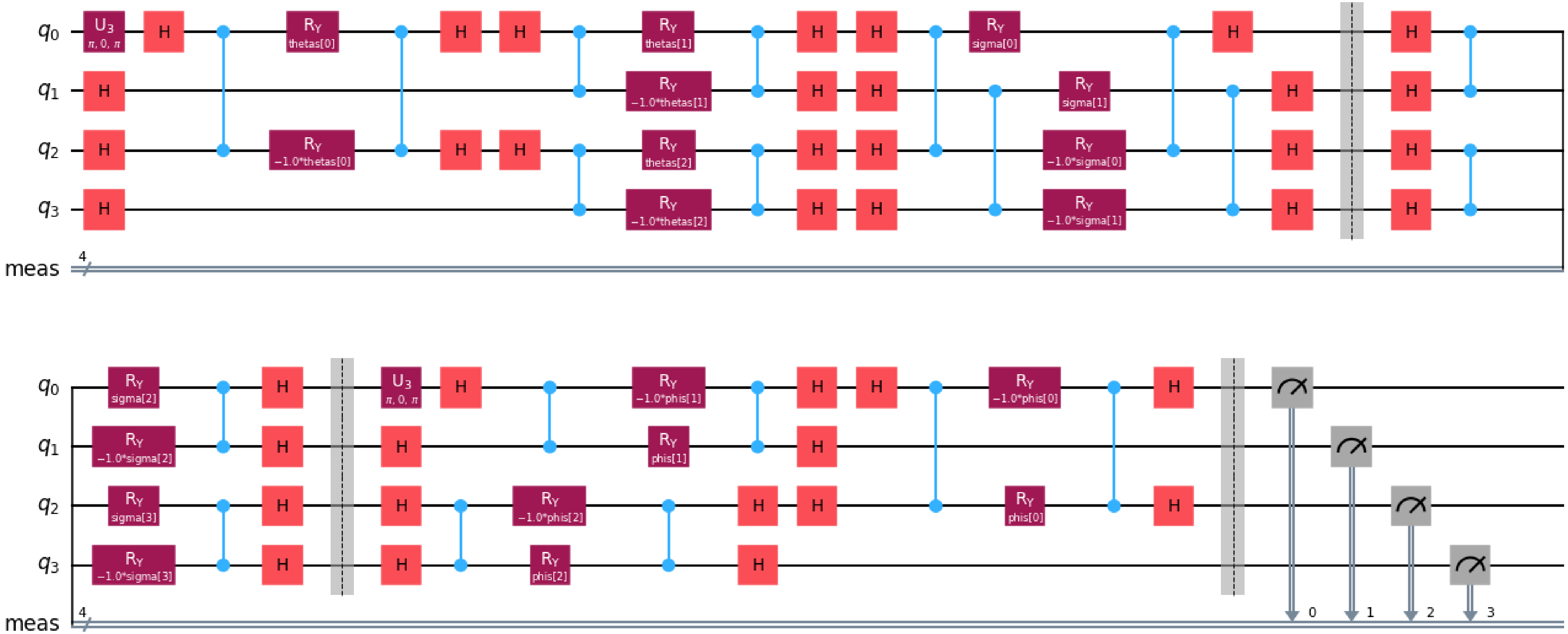
Implementation: Diagonal-Pyramid



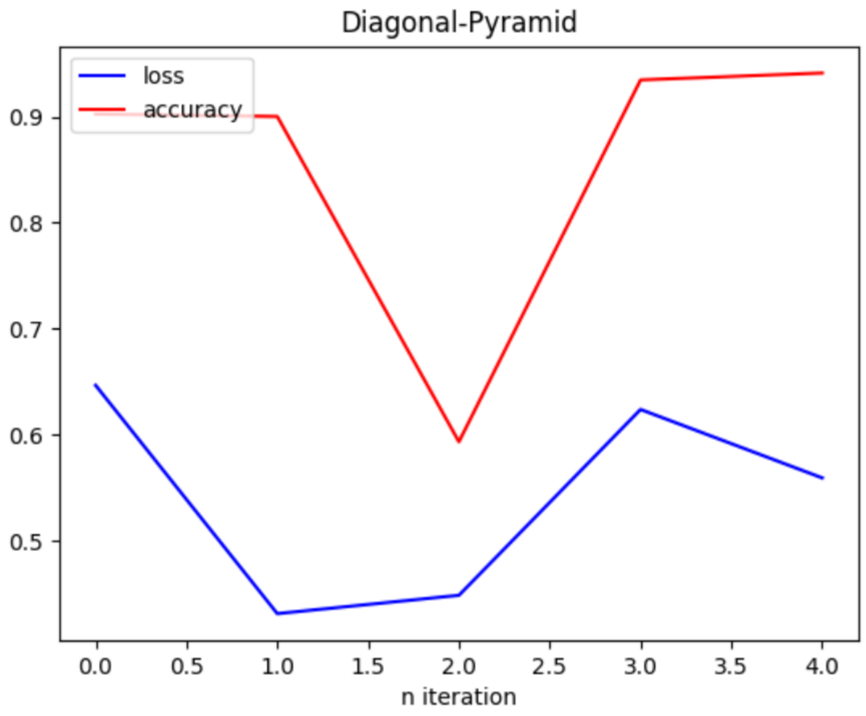
Implementation: Parallel-Butterfly



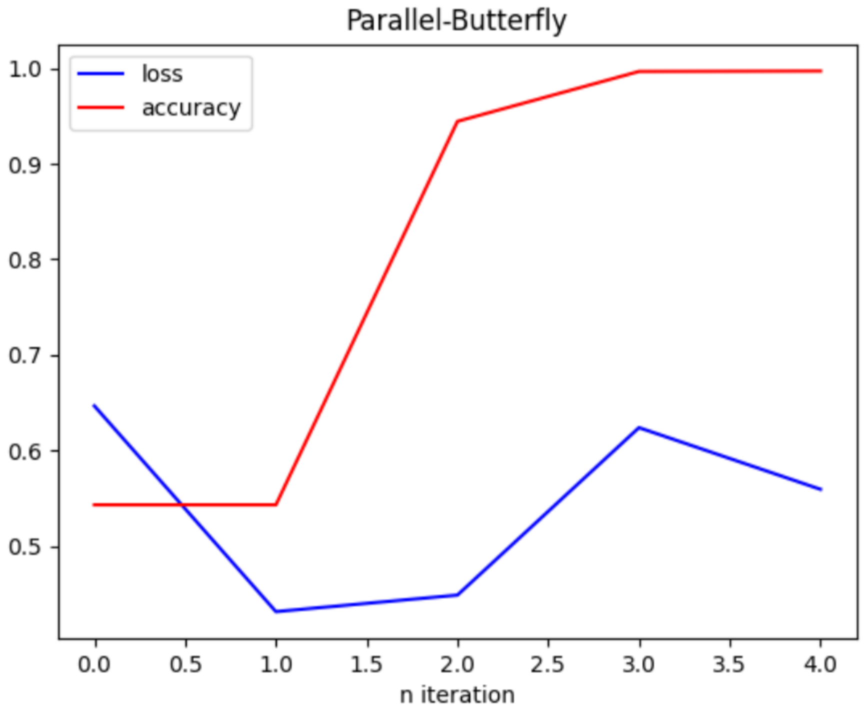
Implementation: Parallel-Butterfly



Results



Diagonal-Pyramid
loss
accuracy
0.9
0.8
0.7
0.6
0.5
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
n iteration



Parallel-Butterfly
loss
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
n iteration

References

1. E. A. Cherrat, I. Kerenidis, N. Mathur, J. Landman, M. Strahm, and Y. Y. Li. Quantum vision transformers. *arXiv preprint arXiv:2209.08167, 2022*.
2. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929, 2020*.
3. S. Johri, S. Debnath, A. Mocherla, A. Singk, A. Prakash, J. Kim, and I. Kerenidis. Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Information, 7(1):122, 2021.*
4. N. Mathur, J. Landman, Y. Y. Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis. Medical image classification via quantum neural networks. *arXiv preprint arXiv:2109.01831, 2021.*
5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems, 30, 2017.*