Supplementary materials

Quantum activation functions for quantum neural networks

Marco Maronese

Istituto di Fotonica e Nanotecnologie, Consiglio Nazionale delle Ricerche, Piazza Leonardo da Vinci 32, I-20133 Milano, Italy and Dipartimento di Fisica, Universitá degli Studi di Milano Bicocca, Piazza della Scienza, I-20126 Milano, Italy

Claudio Destri

Dipartimento di Fisica, Universitá degli Studi di Milano Bicocca, Piazza della Scienza, I-20126 Milano, Italy

Enrico Prati

Istituto di Fotonica e Nanotecnologie, Consiglio Nazionale delle Ricerche, Piazza Leonardo da Vinci 32, I-20133 Milano, Italy

(Dated: January 2022)

I. SUPPLEMENTARY FIGURES

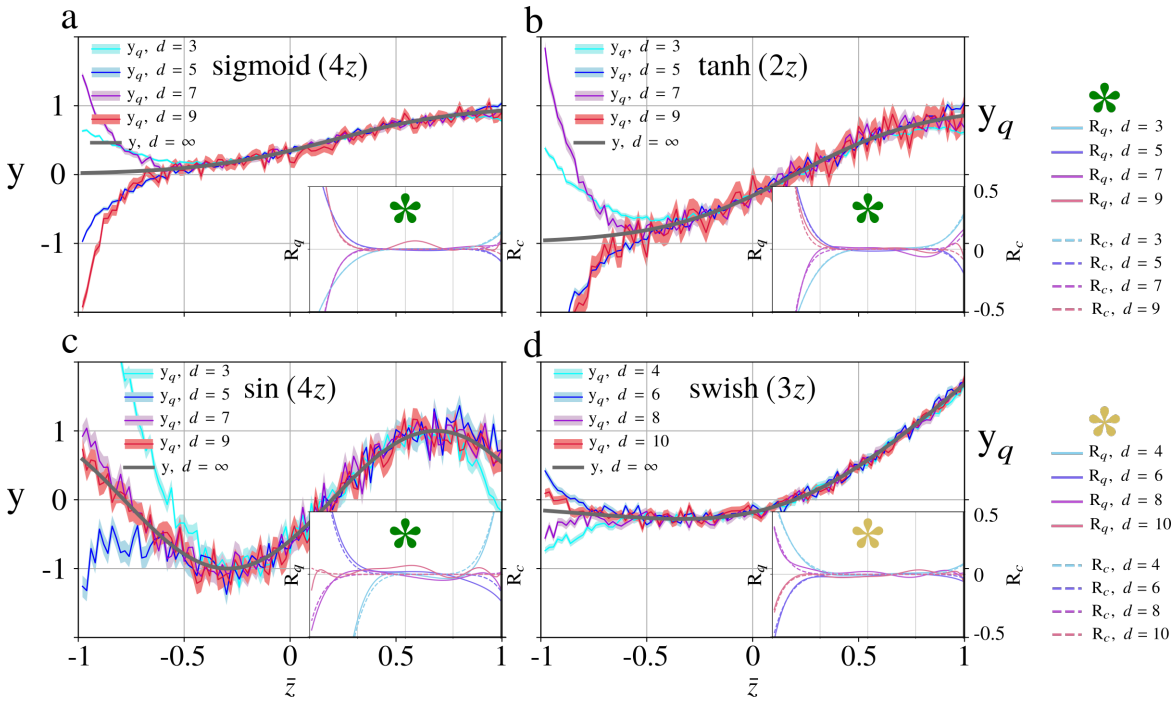


FIG. 1: **Output in function of $\bar{z}$ of a quantum one-layer perceptron for four major analytical activation functions.** In such case $w = (1, 1, 1, 1)$, $\vec{x} = \bar{z}(1, 1, 1, 1)$ and a non-zero bias b is fixed. We have fixed $b = -0.79$ as example, therefore, $\bar{z} = (5z - b)/4 = (5z + 0.79)/4$. The output of the classical perceptron y (gray lines) is computed in the case of the following activation function f: a hyperbolic tangent $tanh(z)$ **(a)**, a logistic function $sigmoid(z)$ **(b)**, a sine function $\sin(z)$ **(c)** and the swish function $z \cdot sigmoid(z)$ **(d)**. The estimation of the quantum perceptron output y_q approximates y at the polynomial order of expansion of $d = 3$ (cyan line) computed with $s = 2^{16}$ samples, $d = 5$ (blue line) with $s = 2^{18}$, $d = 7$ (violet line) with $s = 2^{20}$ and $d = 9$ (red line) with $s = 2^{22}$ in the cases **(a)**, **(b)** and **(c)**. For the swish function the same colors are used at the next order $(d + 1)$ for the same number of samples. In the lower right graphs the plots of R_c and R_q are shown in function of e and at different order d.

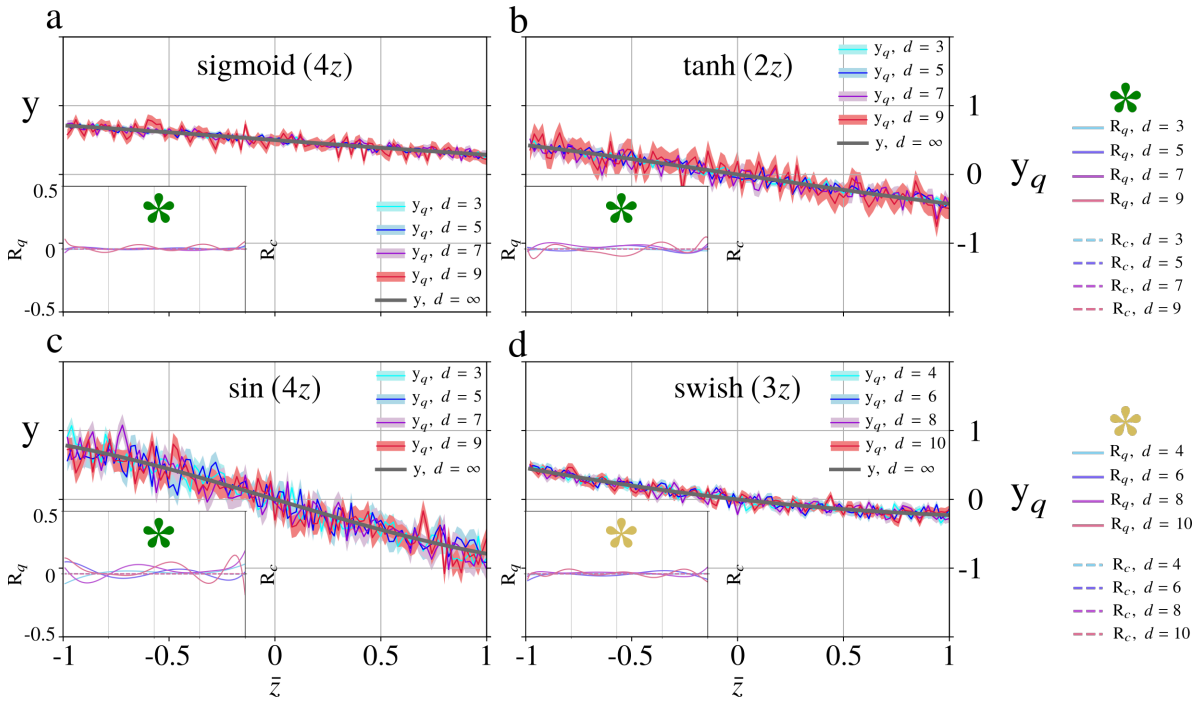


FIG. 2: **Output in function of $\bar{z}$ of a quantum one-layer perceptron for four major analytical activation functions.** In such case $\vec{x} = \bar{z}(1,1,1,1)$, $b = 0$ and a weight vector $\vec{w} \neq (1,1,1,1)$ is fixed. We have fixed $w = (-0.98, -0.45, 0.53, -0.24)$, as example, therefore, $\bar{z} = 5z/(w_0 + w_1 + w_2 + w_3) = 5z/(-1.14)$. The output of the classical perceptron y (gray lines) is computed in the case of the following activation function f: a hyperbolic tangent $tanh(z)$ **(a)**, a logistic function $sigmoid(z)$ **(b)**, a sine function $\sin(z)$ **(c)** and the swish function $z \cdot sigmoid(z)$ **(d)**. The estimation of the quantum perceptron output y_q approximates y at the polynomial expansion's order of $d = 3$ (cyan line) computed with $s = 2^{16}$ samples, $d = 5$ (blue line) with $s = 2^{18}$, $d = 7$ (violet line) with $s = 2^{20}$ and $d = 9$ (red line) with $s = 2^{22}$ in the cases **(a)**, **(b)** and **(c)**. For the swish function the same colors are used at the next order $(d + 1)$ for the same number of samples. In the lower right graphs the plots of R_c and R_q are shown in function of e and at different order d.

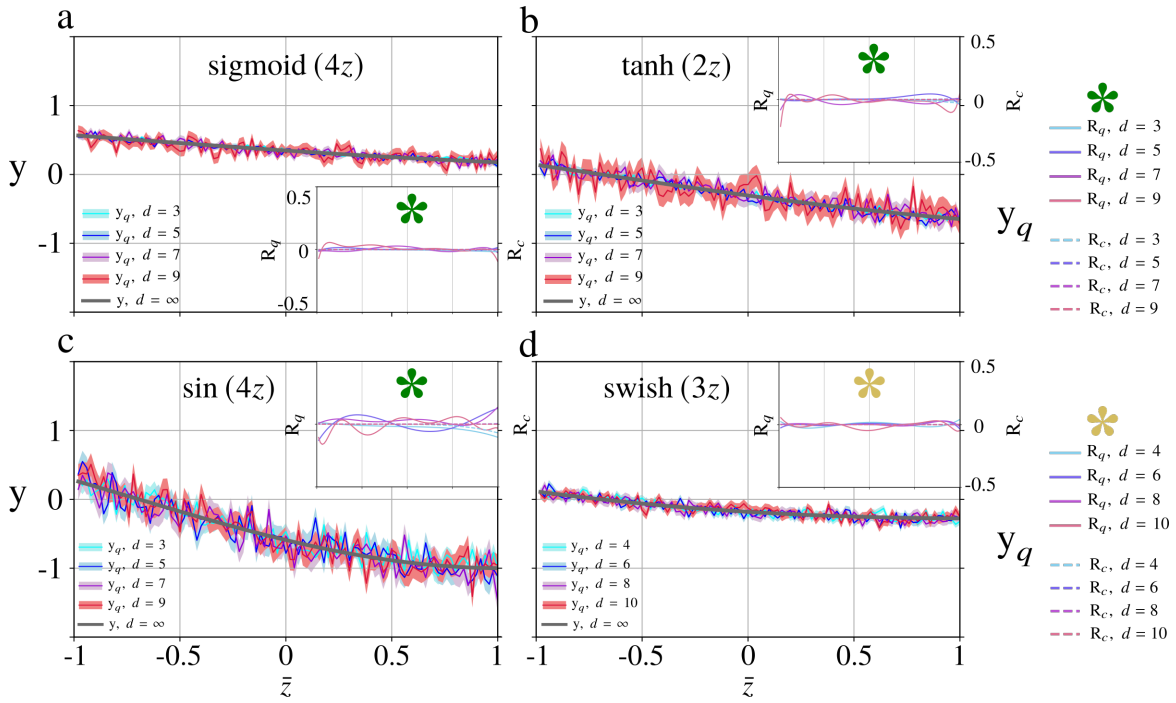


FIG. 3: **Output in function of $\bar{z}$ of a quantum one-layer perceptron for four major analytical activation functions.** In such case $\vec{x} = \bar{z}(1, 1, 1, 1)$, $w = (-0.98, -0.45, 0.53, -0.24)$ and $b = -0.79$. Therefore, $\bar{z} = (5z - b)/(w_0 + w_1 + w_2 + w_3) = (5z + 0.79)/(-1.14)$. The output of the classical perceptron y (gray lines) is computed in the case of the following activation function f: a hyperbolic tangent $tanh(z)$ **(a)**, a logistic function $sigmoid(z)$ **(b)**, a sine function $\sin(z)$ **(c)** and the swish function $z \cdot sigmoid(z)$ **(d)**. The estimation of the quantum perceptron output y_q approximates y at the polynomial order of expansion of $d = 3$ (cyan line) computed with $s = 2^{16}$ samples, $d = 5$ (blue line) with $s = 2^{18}$, $d = 7$ (violet line) with $s = 2^{20}$ and $d = 9$ (red line) with $s = 2^{22}$ in the cases **(a)**, **(b)** and **(c)**. For the swish function the same colors are used at the next order $(d + 1)$ for the same number of samples. In the lower right graphs the plots of R_c and R_q are shown in function of e and at different order d.

II. SUPPLEMENTARY NOTES

A. Supplementary Note 1: Quantum–state initialization

Given a unit vector $\vec{v} \in \mathbb{C}^N$, with $N = 2^n$, a quantum-state initialization transforms a state of the computational basis of n qubits, typically $|0\rangle$, into a vector state $|\psi_v\rangle$ which is a superposition of vectors of the computational basis, where the coefficient v_i is the probability amplitude associated to $|i\rangle$, for any $i = 0, \ldots, N-1$. In a nutshell, this transformation is realized by a unitary transformation $\mathcal{U}(\vec{v})$ such that $\mathcal{U}(\vec{v})|0\rangle = |\psi_v\rangle$. An algorithm to realize one such $\mathcal{U}(\vec{v})$ is given by [1]. There, the modules of the coefficients of the vector $\vec{v}$ are stored in the quantum state by applying a series of multi-controlled R_y gates while the relative phases are added at a second stage by applying a multi-controlled R_z gates series. Typically, the circuit depth is $\mathcal{O}(N)$ to store the modules of $\vec{v}$ and $\mathcal{O}(N)$ to introduce the relative phases. Moreover, for classical data, a suitable preprocessing of the data is needed to compute the angles of the R_y and R_z rotations. The details of the calculus of such angles is reported in Supplementary Note **2**.

B. Supplementary Note 2: Algorithm to initialize a quantum state

In this section we show how to encode classical data in a quantum state, a procedure needed in the algorithm discussed in the main text. Let $\vec{v} = (v_0, v_1, \ldots, v_{N-1})$ be a vector in $\mathbb{C}^N$ such that $|\vec{v}| = 1$ and let q be a quantum register of $n = \log_2 N$ qubits. Then there exists a quantum circuit that transforms $|0\rangle$ into the arbitrary state vector

$$|\psi_v\rangle \equiv \sum_{i=0}^{N-1} v_i |i\rangle \tag{1}$$

The algorithm actually starts from the target state $|\psi_v\rangle$ and construct a unitary transformation that evolves it back to $|0\rangle$. The initialization circuit is therefore the reverse of this. To transform $|\psi_v\rangle$ to $|0\rangle$, one can proceed recursively by disentangling one qubit at a time. Consider first the least significant qubit q_0 in the register. We exploit the following theorem from Ref. [1].

Theorem 1 (Shende-Bullock-Markov – Disentanglement of a qubit) *Given a register q of n qubits in an arbitrary state $|\psi\rangle \in \mathcal{H}_q^{\otimes n}$, there exist a set of angles $\{\theta_j, \varphi_j\ , j =$*

$0, 1, \ldots, N,\ N = 2^{n-1}\}$, *such that the block–diagonal unitary matrix*

$$U_n(\{\theta_j\}, \{\varphi_j\}) = \bigoplus_{j=0}^{N-1} R_y\left(-\theta_j\right) R_z\left(-\varphi_j\right) \tag{2}$$

transforms $|\psi\rangle$ *into the direct product* $|\psi'\rangle \otimes |0\rangle$*, where* $|\psi'\rangle \in \mathcal{H}_q^{\otimes n-1}$.

It is clear now that another set of angles $\{\theta'_j, \varphi'_j\ , j = 0, 1, \ldots, N',\ N' = 2^{n-2}\}$ and unitary transformation $U_{n-1}(\{\theta'_j\}, \{\varphi'_j\})$ can be found that disentangles q_1 and yields a state of the form $|\psi''\rangle \otimes |00\rangle$. The procedure is then repeated until all qubits are disentangled and the state of the most significant qubit q_n is rotated into $|0\rangle$ (possibly up to an irrelevant overall phase), eventually yielding the state $|0\rangle^{\otimes n}$. Alltogether, we need n sets of angles of the form

$$\{\theta_{j,m}, \varphi_{j,m},\ j = 0, \ldots, 2^{n-m-1} - 1,\ m = 0, 1, \ldots, n-1\} \tag{3}$$

with the associated operators

$$U_{n,m}(\{\theta_{j,m}, \varphi_{j,m}\}) \equiv U_{n-m}(\{\theta_{j,m}, \varphi_{j,m}\}) \otimes \mathbb{1}^{\otimes m} \tag{4}$$

where the definition of the transformation U_{n-m} is given by the Equation 2.

Therefore, the inverse of the operator product

$$\mathcal{U}_n(\{\theta_{j,m}, \varphi_{j,m}\}) \equiv \prod_{m=1}^{n} U_{n,n-m}(\{\theta_{j,m}, \varphi_{j,m}\}) \tag{5}$$

initializes a register of n qubits in the state $|\psi_v\rangle$, once a preprocessing phase is run to compute the angles $\{\{\theta_{j,m}\}, \{\varphi_{j,m}\}\}$ from the vector $\vec{v}$. For the case of quantum data the angles of the Bloch spheres of the qubits are already given therefore a preprocessing phase to compute them is not required.

1. Special case of real data

If $\vec{v} = (v_0, v_1, \ldots, v_{N-1})$ is a real vector as in the case of the vectors $\vec{v}_x$ and $\vec{v}_{w,b}$ in the qubit–based perceptron model of the main text, then the unitary matrix $\mathcal{U}(\vec{v})$ that transforms $|0\rangle^{\otimes n}$ into $|\psi_v\rangle$ can be factored into two matrices according to

$$\mathcal{U}(\vec{v}) = Ph_n(\vec{v})\, \mathcal{U}_n^{\dagger}(\{\theta_{j,m}\}, \{\varphi_{j,m} = 0\}) \tag{6}$$

where, since $\mathcal{U}_n^\dagger(\{\{\theta_{j,m}\},\{\varphi_{j,m}\}\})\,|0\rangle^{\otimes n} = |\psi_v\rangle$,

$$\mathcal{U}_n^\dagger(\{\theta_{j,m},\varphi_{j,m}=0\})\,|0\rangle^{\otimes n} = |\psi_{v_+}\rangle \equiv \sum_{i=0}^{N-1} |v_i|\,|i\rangle \tag{7}$$

and

$$Ph_n(\vec{v})|\psi_{v_+}\rangle = \sum_{i=0}^{N-1} \frac{v_i}{|v_i|} v_i\,|i\rangle = |\psi_v\rangle \tag{8}$$

is the diagonal transformation that introduces the correct negative signs of the coefficients v_i.

2. Computation of the angles $\theta_{j,m}$

The transformation $\mathcal{U}_n^\dagger(\{\{\theta_{j,m}\},\{\varphi_{j,m}=0\}\})$ depends only by a set of angles $\{\theta_{j,m}\}$ and, considering the Equation 5 with $\varphi_{j,m}=0$ for all the pairs j,m

$$\mathcal{U}_n^\dagger(\{\{\theta_{j,m}\},\{\varphi_{j,m}=0\}\}) = \prod_{m=0}^{n-1}\left[\bigoplus_{j_m=0}^{2^m-1} R_y\left(\theta_{j,m}\right)\otimes \mathbb{1}^{\otimes(n-m)}\right] \tag{9}$$

Let's see how we can compute such angles. Let's rewrite the state $\left|\psi_{\vec{v}_+}\right\rangle$ as:

$$\begin{aligned}\left|\psi_{\vec{v}_+}\right\rangle &\equiv \sum_{i=0}^{\frac{N}{2}-1} |i\rangle\otimes\Big(|v_{2i}|\,|0\rangle + |v_{2i+1}|\,|1\rangle\Big) = \\ &= \sum_{i=0}^{\frac{N}{2}-1} |i\rangle\otimes\Big(|v_{i_0}|\,|0\rangle + |v_{i_1}|\,|1\rangle\Big)\end{aligned} \tag{10}$$

In brief, the indices $2i$ and $2i+1$ are replaced with i_0 and i_1 $\forall i = 0,1,\ldots,\frac{N}{2}-1$. Let's show how the angles $\{\theta_{j,m}\}$ are computed by the coefficients $|v_i|$.

As first step, let's find the relationship between the angles $\{\theta_{j,0}\}$ and the coefficient $|v_i|$. From the Shende-Bullock-Markov Theorem 1, since the the coefficients $|v_i|$ have a null phase, one has that the operator

$$U_n = \bigoplus_{i=0}^{N-1} R_y\left(-\theta_i\right) \tag{11}$$

transforms the state $|\psi_{\vec{v}}\rangle$ in a separate state. Let's define a state

$$|\rho\rangle = \sum_{i=0}^{\frac{N}{2}-1} \rho_i\,|i\rangle \in \mathcal{H}^{\otimes(n-1)} \tag{12}$$

such that $U_n \left|\psi_{\vec{v}_+}\right\rangle = |\rho\rangle |0\rangle$ then, since each rotation $R_y(-\theta_i)$ is defined on the subspace $\mathcal{H}_i = \{|i\rangle \otimes |0\rangle, |i\rangle \otimes |1\rangle\}$, we obtain $\frac{N}{2}$ equations for the angles $\{\theta_{i,0}\}$

$$\overbrace{\begin{pmatrix} \cos\frac{\theta_i}{2} & \sin\frac{\theta_i}{2} \\ -\sin\frac{\theta_i}{2} & \cos\frac{\theta_i}{2} \end{pmatrix}}^{R_y(-\theta_i)} \begin{pmatrix} |v_{i_0}| \\ |v_{i_1}| \end{pmatrix} = \rho_i \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \forall i \tag{13}$$

The correspondent system of equations is:

$$\begin{cases} |v_{i_0}| \cos\frac{\theta_i}{2} + |v_{i_1}| \sin\frac{\theta_i}{2} = \rho_i \\ -|v_{i_0}| \sin\frac{\theta_i}{2} + |v_{i_1}| \cos\frac{\theta_i}{2} = 0 \end{cases} \quad \forall i \tag{14}$$

By using the second equation of the system it is possible to obtain the angles θ_i in the case of $v_{i_0} \neq 0$

$$\begin{aligned} & -|v_{i_0}| \sin\frac{\theta_i}{2} + |v_{i_1}| \cos\frac{\theta_i}{2} = 0 \\ \Rightarrow\ & \frac{|v_{i_1}|}{|v_{i_0}|} = \tan\frac{\theta_i}{2} \\ \Rightarrow\ & \theta_i = 2 \arctan\frac{|v_{i_1}|}{|v_{i_0}|} \end{aligned} \tag{15}$$

If $v_{i_0} = 0$ we can easily take $\theta_i = \pi$ because in such way the matrix in the equation 13 allows to transform the state $|1\rangle$ in $|0\rangle$.

Once obtained the angles as above, we have to explicitly evaluate ρ_i in order to iterate the algorithm to separate the next qubit in the register. To do that the first equation in the system in Eq.14 is used:

$$\rho_i = \left(|v_{i_0}|^2 \cos^2\frac{\theta_i}{2} + |v_{i_1}|^2 \sin^2\frac{\theta_i}{2} \right)^{\frac{1}{2}} \tag{16}$$

Therefore, the values ρ_i are positive defined.

Now we need an operator U_{n-1} to separate the less significant qubit in the $(n-1)$-qubits state $|\rho\rangle$. In order to find the angles appropriate to build the operator U_{n-1} we repeat the computation from the Eq.13 with the coefficient of the state $|\rho\rangle$ taken two by two like the coefficient v_{i_0} and v_{i_1}.

$$|\rho\rangle = \sum_{i=0}^{\frac{N}{4}-1} |i\rangle \otimes \left(\rho_{i_0} |0\rangle + \rho_{i_1} |1\rangle \right) \tag{17}$$

Defining a state $|\alpha\rangle \in \mathcal{H}^{\otimes(n-2)}$, such that $U_{n-1}(\{\theta_{i,1}\}) |\rho\rangle = |\alpha\rangle |0\rangle$, it is possible to find the relationship between $\theta_{i,1}$ and the coefficients ρ_{i_0} and ρ_{i_1} and iterate the calculus to find all the angles $\{\theta_{i,m}\}$ $\forall i, m$.

3. Implementation of $Ph_n^\dagger(\vec{v})$

As shown in the Equation 8, $Ph_n^\dagger(\vec{v})$ is a diagonal matrix with diagonal terms 1 or -1. There are many ways to implement such operator on a near-term quantum computer. Let's introduce in a few lines a method based on the quantum hypergraph state [2].
Considering n qubits in a state $|+\rangle^{\otimes n}$, where $\sqrt{2}\,|+\rangle = |0\rangle + |1\rangle$, the quantum hypergraph state corresponds to the hypergraph $g_{\leq n} = \{V, E\}$, where V is a set of n vertices and E is a set of hyper-edges of any order k from $k = 1$ and $k = n$, and it is defined as

$$|g_{\leq n}\rangle = \prod_{k=1}^{n} \prod_{\{i_1,\ldots,i_k\}\in E} C^k Z_{i_1,\ldots,i_k} \, |+\rangle^{\otimes n} \tag{18}$$

Let's define now the set of Real Equally Weighted (REW) states A REW state is defined as follows

$$|f\rangle = \frac{1}{2^{n/2}} \sum_{i=0}^{N-1} (-1)^{f(i)} \, |i\rangle \tag{19}$$

where $f : \{0,1\}^n \to \{0,1\}$ is a boolean function. The set of the RAW state of n qubits and the set of the hypergraph state on n qubits coincide [2]. Since $Ph_n^\dagger(\vec{v}) = diag(\frac{v_0}{|v_0|}, \ldots, \frac{v_{N-1}}{|v_{N-1}|})$ it follows that the state

$$Ph_n^\dagger(\vec{v}) \, |+\rangle^{\otimes n} = \sum_{i=0}^{N-1} \frac{v_i}{|v_i|} \, |i\rangle \tag{20}$$

is a RAW state and, therefore, there exists a correspondent hypergraph state $|g_{\leq n}\rangle$ as defined in the Equation 18. Considering the Equations 18 and 20 it results that there exists an hypergraph on n vericies $g_{\vec{v}} = \{V_{\vec{v}}, E_{\vec{v}}\}$

$$Ph_n^\dagger(\vec{v}) = \prod_{k=1}^{n} \prod_{\{i_1,\ldots,i_k\}\in E_{\vec{v}}} C^k Z_{i_1,\ldots,i_k} \tag{21}$$

Given a RAW state, there is a method to obtain the correspondent hypergraph state [2].

4. Example of a 3-qubit state initialization

Let's show, for example, the implementation of an initialization of a 3-qubits state with real and positive amplitudes. We need to implement the inverse of the operator $\mathcal{U}_3$ and to do it we have to compute only 7 angles. Indeed an arbitrary 3-qubits quantum state is a superposition of 2^3 eigenstates so 8 is the number of the amplitudes but since the state must

respect the unitarity constraint we need only 7 independent parameters to define it. Let's consider the arbitrary 3-qubits state

$$\begin{aligned} |\psi_{\vec{v}}\rangle &= \sum_{i=0}^{3} |i\rangle \otimes \Big(v_{i_0} |0\rangle + v_{i_1} |1\rangle \Big) \\ \vec{v} &= (v_{0_0}, v_{0_1}, v_{1_0}, v_{1_1}, v_{2_0}, v_{2_1}, v_{3_0}, v_{3_1}) \in \mathbb{R}^8 \end{aligned} \tag{22}$$

Being $\vec{v}$ a real values vector, the transformation $\mathcal{U}(\vec{v})$, such that $\mathcal{U}(\vec{v}) |0\rangle^{\otimes 3} = |\psi_{\vec{v}}\rangle$, is

$$\begin{aligned} \mathcal{U}(\vec{v}) =& Ph_3^\dagger(\vec{v}) \mathcal{U}_3^\dagger(\{\{\theta_{j,m}\}, \{\varphi_{j,m} = 0\}\}) \\ =& Ph_3^\dagger(\vec{v}) \prod_{m=0}^{2} U_{3,m}^\dagger(\{\theta_{j,m}\}, \{0\}) \end{aligned} \tag{23}$$

Therefore, from the definition in the Equation 4, the transformation in the Equation 23 becomes, implying the identity transformation,

$$\mathcal{U}(\vec{v}) = Ph_3^\dagger(\vec{v}) U_3^\dagger(\{\theta_{j,0}\}) U_2^\dagger(\{\theta_{j,1}\}) U_1^\dagger(\{\theta_{j,2}\}) \tag{24}$$

Since we have 4 pairs (v_{i_0}, v_{i_1}), the operator U_3 depends on 4 angles, U_2 depends on 2 angles and U_1 by one. To simplify the notation, let's rewrite the set of angles $\{\theta_{j,0}\}$ as $\{\theta_0, \theta_1, \theta_2, \theta_3\}$, the set $\{\theta_{j,1}\}$ as $\{\theta_4, \theta_5\}$ and $\theta_{0,2}$ as θ_6. The angles $\{\theta_0, \theta_1, \theta_2, \theta_3\}$ are computed by the Equation 15 imposing that

$$U_3 \left|\psi_{\vec{v}_+}\right\rangle = \left[\sum_{i=0}^{1} |i\rangle \otimes \Big(\rho_{i_0} |0\rangle + \rho_{i_1} |1\rangle \Big) \right] \otimes |0\rangle \tag{25}$$

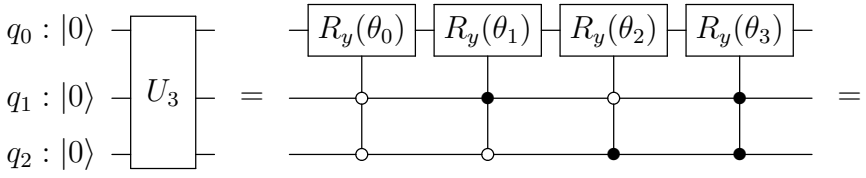
Then, to obtain the operator U_2, we must compute only two angles from the pair ρ_{0_0}, ρ_{0_1} and ρ_{1_0}, ρ_{1_1}. The final state will be of kind

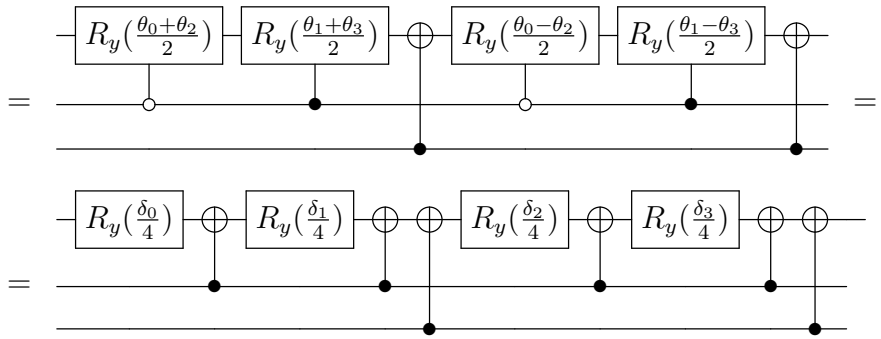
$$(U_2 \otimes \mathbb{1}) U_3 \left|\psi_{\vec{v}_+}\right\rangle = \Big(\alpha_{0_0} |0\rangle + \alpha_{0_1} |1\rangle \Big) \otimes |00\rangle \tag{26}$$

We will use such coefficients α_{0_0} and α_{0_1} to compute the 7-th and last angle to obtain the operator U_1 which will be only a one-qubit gate.

Now, the goal is to build a quantum circuit which implements U_3 and U_2 only with one-qubit and CX gates using the angles obtained above.

Later it is shown decomposition [**?**] of the transformation U_3 in one-qubit and CX gates.





The angles δ_i are defined in the following way:

$$\begin{aligned}
\delta_0 &= \theta_0 + \theta_2 + \theta_1 + \theta_3 \\
\delta_1 &= \theta_0 + \theta_2 - \theta_1 - \theta_3 \\
\delta_2 &= \theta_0 - \theta_2 + \theta_1 - \theta_3 \\
\delta_3 &= \theta_0 - \theta_2 - \theta_1 + \theta_3
\end{aligned} \tag{27}$$

C. Supplementary Note 3: Evaluation of systematic error in presence of simulated noise

Here, the performance of the perceptron in terms of its resilience to noise is shown for the major case of the *tanh* activation functions. In Figure 4 there is an example of the output of a quantum perceptron both free and affected by noise, added as simulated error. The quantum operations are simulated with *qasm_simulaotr* with a noise model based on the device *ibmq_16_melbourne* implemented by the functions present in *QisKit Aer* .

As shwon in Figure 4, the simulated error reduces to a systematic error which can be corrected by a linear transformation. Therefore, given $\bar{y}_q(z)$ being the output of the perceptron in function of $z = \vec{w} \cdot \vec{x} + b$ obtained with a noise model and the expected output $y_q(z)$, it is possible to correct the error derived from the simulated noise with a post-processing operation following the formula

$$\bar{y}_q(z) \to y_q(z) = c_1 \bar{y}_q(z) + c_2 \tag{28}$$

where c_1 and c_2 are the constant coefficients of the linear transformation: c_1 is the slope while c_2 is the intercept.

The fits for some cases are systematically analysed and summarized in the Table I. We remind that the *qasm_simulator* can simulate only the noise of a given real quantum device

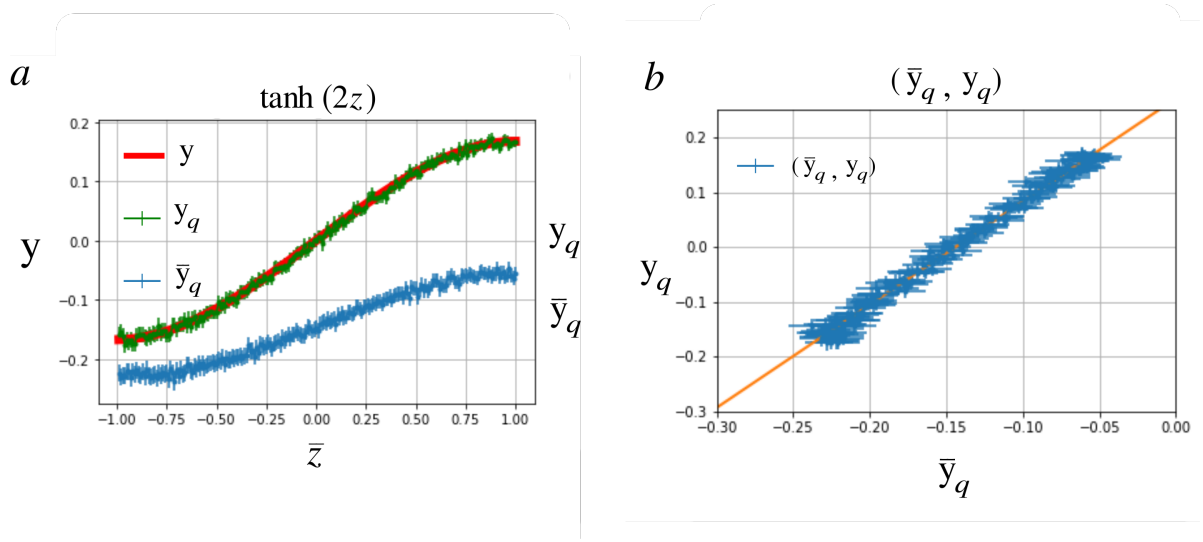


FIG. 4: **Effect of the noise on a quantum one-layer perceptron with hyperbolic tangent activation functions.** The output of a quantum one-layer perceptron involving the *tanh* activation functions is plotted as a function of $\bar{z}$, without (y_q) and with $(\bar{y}_q)$ simulated noise. **(a)**. Here, $\vec{x} = \bar{z}(1,1,1,1)$, $b = 0$ and a weight vector $\vec{w} = (1,1,1,1)$ is assigned. The relation between the two outputs $(\bar{y}_q, y_q)$ **(b)**, fitter by the orange line, shows that the addition of a simulated noise is equivalent to a linear transformation. The outputs were calculated on 200 different values of z with $R = 2^{15}$ circuit shots for each $\bar{z}$.

up to a certain accuracy degree. The error over $\bar{y}_q(z)$ generated by actual noise in the hardware it is not necessarily corrected by a linear transformation like shonw in Equation 28.

At the current stage of development of quantum hardware, at the best of our knowledge, the complexity of the circuit implementing the estimate $y_q(z)$ together with the actual values of T_1 and T_2 in the *ibmq_16_melbourne*, like any other Noise Intermediate Scale Quantum computer device, prevented us to obtain results from quantum hardware comparable to those achieved by using the simulated noise. Simulated noise should better be considered an approximation of the noise of a future generation of quantum computers. Therefore developing a correction strategy for the case of simulated noise can bring some information over the future implementation of the such algorithm of a quantum perceptron with arbitrary activation function.

d	b	c_1	σ_{c_1}	c_2	σ_{c_2}	r factor
3	0	1.8811	$1.6 \cdot 10^{-4}$	0.271996	$3.9 \cdot 10^{-6}$	0.995
3	1	1.8526	$1.7 \cdot 10^{-4}$	0.268245	$3.2 \cdot 10^{-6}$	0.995
3	-1	1.9861	$2.8 \cdot 10^{-4}$	0.284283	$8.5 \cdot 10^{-6}$	0.993

d	b	c_1	σ_{c_1}	c_2	σ_{c_2}	r factor
5	0	2.5827	$2.2 \cdot 10^{-3}$	0.7181	$1.7 \cdot 10^{-5}$	0.968
5	1	2.5973	$2.9 \cdot 10^{-3}$	0.7245	$2.1 \cdot 10^{-3}$	0.960
5	-1	2.4533	$2.9 \cdot 10^{-3}$	0.6792	$2.3 \cdot 10^{-3}$	0.955

d	b	c_1	σ_{c_1}	c_2	σ_{c_2}	r factor
9	0	2.2191	$4.3 \cdot 10^{-3}$	0.7162	$4.5 \cdot 10^{-3}$	0.922
9	1	2.3832	$5.6 \cdot 10^{-3}$	0.7735	$5.8 \cdot 10^{-3}$	0.914
9	-1	2.0946	$3.6 \cdot 10^{-3}$	0.6737	$3.8 \cdot 10^{-3}$	0.927

TABLE I: **Fit of the outputs of a quantum one-layer perceptron with hyperbolic tangent activation functions with and without simulated noise.** The fit between $\bar{y}_q(z)$ and $y_q(z)$ is calculated with different values of the bias b (first column) and with different degrees d of Taylor series approximations of the activation function. d is the power of the Taylor series. In the last column the goodness factor r of the fit is reported and it is always > 0.9 with a number of points equal to 200 for each fit and with $R = 2^{15}$ circuit shots to evaluate the perceptron outputs.

[1] Shende, V. V., Bullock, S. S. & Markov, I. L. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**, 1000–1010 (2006).

[2] Rossi, M., Huber, M., Bruß, D. & Macchiavello, C. Quantum hypergraph states. *New Journal of Physics* **15**, 113022 (2013).