Universal quantum perceptron as efficient unitary approximators

E. Torrontegui and J. J. García-Ripoll
Instituto de Física Fundamental IFF-CSIC, Calle Serrano 113b, 28006 Madrid, Spain

We demonstrate that it is possible to implement a quantum perceptron with a sigmoid activation function as an efficient, reversible many-body unitary operation. When inserted in a neural network, the perceptron’s response is parameterized by the potential exerted by other neurons. We prove that such a quantum neural network is a universal approximator of continuous functions, with the same power as classical neural networks. While engineering general perceptrons is a challenging control problem – also defined in this work –, the ubiquitous sigmoid-response neuron can be implemented as a quasi-adiabatic passage with an Ising model. In this construct, the scaling of resources is favorable with respect to the total network size and is dominated by the number of layers. We expect that our sigmoid perceptron will have applications also in quantum sensing or variational estimation of many-body Hamiltonians.

Quantum computing and machine learning are two computing paradigms that fight the limitations of procedural programming. While the first one is based on a physically different model of computation, the second one reuses von Neumann architectures to build sophisticated approximation models that outperform traditional algorithms. Quantum machine learning merges ideas from both paradigms, to create new quantum algorithms. Quantum machine learning paradigmatically seeks to reproduce classical neural networks using quantum states, other applications of this perceptron include the design of multiqubit conditioned quantum gates, or the design of more general perceptrons with sophisticated response functions that can be applied in quantum sensing. Our perceptron is intimately related to a recent proposal by Cao et al., which implements the nonlinear activation of a qubit using repeat-until-success quantum gates. As discussed later, our perceptron shares the same potential applications with various advantages: scaling of resources, avoidance of phase wrapping (works for arbitrarily large $|x|$) or utility for general nonlinear sensing.

In a feed-forward network setup, the perceptron gate is a qubit with a nonlinear excitation response to an input field $\rho_j(x_j) = \sqrt{1 - f(x_j)} |0_j\rangle + \sqrt{f(x_j)} |1_j\rangle$. (1)

In a feed-forward network setup, the perceptron gate is conditioned on a mean field generated by neurons in earlier layers, $x_j = \sum_{k<j} w_{jk} \sigma_k^z - \theta_j$, with the same weights $w_{jk}$ and biases $\theta_j$ as classical networks. This allows us to prove that a network based on this perceptron is a universal approximator of arbitrary continuous functions. We also prove that the perceptron gate $\hat{U}_j(x_j; f)$ has an efficient implementation as a quasidiabatic passage in an Ising model with transverse field, with a total implementation time that scales as $O(L \times \log(\epsilon/N)/\Omega_f)$, with the number of layers $L$, total number of neurons $N$, gate error $\epsilon$ and activation step size $\Omega_f$ [cf. Fig. 1]. In addition to reproducing classical neural networks using quantum states, other applications of this perceptron include the design of multiqubit conditioned quantum gates, or the design of more general perceptrons with sophisticated response functions that can be applied in quantum sensing.

As discussed later, our perceptron shares the same potential applications with various advantages: scaling of resources, avoidance of phase wrapping (works for arbitrarily large $|x|$) or utility for general nonlinear sensing.

FIG. 1. (a) Quantum perceptron as a qubit that excites coherently according to (1) with a probability $P_j = \frac{1}{2}(1 + \langle \sigma_j^z \rangle) = f(x_j)$ that grows nonlinearly with the activation potential $x_j$. (b) When this perceptron is integrated in a feed-forward neural network, the potential depends on neurons in earlier layers, e.g. $x_{10} = \sum_{k=5}^8 w_{10,k} \sigma_k^z + \theta_{10}$.
Classical neural networks.— First models of neurons were based on the McCulloch and Pitts 24 idea of a two-state system, with active \( s = 1 \) or resting \( s = 0 \) states. The activation of a neuron \( s_i \), called perceptron 24, is determined by the influence of connected partners, as an update mechanism

\[
s'_i = \begin{cases} 
1, & \text{if } \sum_{j=1}^{N} w_{ij} s_j \geq \theta_i \\
0, & \text{otherwise}.
\end{cases}
\]  

(2)

The weights \( w_{ij} \) determine the network architecture 24–25, which we restrict to feed-forward networks, \( w_{ij} \neq 0 \Rightarrow w_{ji} = 0 \) with a precise activation order, organized in layers, as sketched in Fig. 1.

Modern works with artificial neural networks allow for the perceptron response to take real values in a bounded interval \( s_i \in [0,1] \) 24. The step function (2) becomes a graded response \( s'_i = f \left( \sum_{j=1}^{N} w_{ij} s_j - \theta_i \right) \) that is usually determined by the sigmoid logistic function \( f(x_j) = 1/(1 + e^{-x_j}) \) [cf. dashed line in Fig. 1]. The power of such continuous neural networks lays in the fact that already two layers can approximate any complex function 22—hence, their capacity to classify complex data and reveal unknown patterns—. This result is derived from the “universal approximation theorem”, which we now recall in the form by Cybenko 29.

**Theorem 1.** Let \( I_M = [0,1]^M \) be the \( M \)-dimensional unit cube and \( C(I_M) \) the space of continuous functions on \( I_M \). Let the function \( \eta \) be continuous and sigmoidal—i.e. \( \eta(\infty) \rightarrow 1, \eta(-\infty) \rightarrow 0 \). Then, finite sums of the form

\[
Q(s) = \sum_{j} \alpha_j \eta \left( \sum_{k=1}^{M} w_{jk} s_k - \theta_j \right)
\]  

are dense in \( C(I_M) \). In other words, given any \( q \in C(I_M) \) and \( \varepsilon > 0 \), there exists a sum \( Q(s) \) with \( N_c \) terms, for which \( |Q(s) - q(s)| \leq \varepsilon \) for all \( s \in I_M \).

Following this theorem, we can design a two-layer neural network with \( M \) input and \( N_c \) output neurons to approximate any function \( q(s) \in C(I_M) \). The \( M \) input neurons will be assigned argument of the function we wish to compute \( s^{\text{in}}_i = s_i \). We will use the graded response update to determine the values of the \( N_c \) output neurons \( s^{\text{out}}_j = f \left( \sum_{k=1}^{M} w_{jk} s^{\text{in}}_k - \theta_j \right), \ j = 1, \ldots, N_c \). Finally, we approximate the function as \( q(s) \simeq \sum_{j=1}^{N_c} \alpha_j s^{\text{out}}_j \). Determining the values of \( \alpha_j \), \( w_{jk} \) and \( \theta_j \) for an specific function amounts to training the network.

Quantum perceptron.— Earlier proposals for quantum neurons encode the perceptron signal in a qubit space, \( |s_i\rangle \), and implement the discrete transformations as maps \( |s_1, \ldots, s_N\rangle \rightarrow |s'_1, \ldots, s'_N\rangle \) that follow the nonlinear transformations above. Unfortunately, most of such maps are not invertible and do not correspond to unitary evolution 29. One solution is to use auxiliary qubits 29 that are changed to ensure unitarity \( |s\rangle |\text{anc}\rangle \rightarrow |s', \text{anc}'\rangle \). Another interesting approach is to use repeated measurements with auxiliary qubits to implement nonlinear conditional gates 29. Our approach is similar to this last option but slightly more general: our perceptron is a quantum device that undergoes a coherent transformation 1 which can be implemented as a SU(2) rotation parameterized by a general input field \( x_j \).

\[
\hat{U}_j(x_j; f) = \exp \left[ i f(x_j) \hat{\sigma}^z_j \right].
\]  

(4)

The unitary operator \( \hat{U}_j(x_j; f) \) is fully defined by an activation function \( f(x_j) : \mathbb{R} \rightarrow [0,1] \), the weights \( w_{jk} \) and the thresholds \( \theta_j \). The angle \( \hat{f}(x_j) = \arcsin(f(x_j)^{1/2}) \) depends nonlinearly on the input field \( x_j = \sum_{k<j} w_{jk} \theta_k - \theta_j \), and this manifests in the Heisenberg picture

\[
\langle \hat{\sigma}^z_j \rangle = 2 \left( \sum_{k<j} w_{jk} \theta_k - \theta_j \right) - 1.
\]  

(5)

This relation mimics the one in the classical neural network and can be used to prove that our construction is a universal approximator of continuous functions.

**Theorem 2.** Any continuous function \( q(s) \in C(I_M) \) can be approximated on a sublattice \( I_{M,k} = \{ 0, 2^{-k}, 2 * 2^{-k}, \ldots, 2^{-k} \}^M \) up to an error \( \varepsilon \) using \( k \times M + N_c \) quantum neurons. The protocol is as follows: (i) Determine the parameters \( w_{ij}, \theta_j \) and \( \alpha_i \) to approximate this function using a classical network of \( M + N_c \) continuous valued neurons. (ii) For any \( s \in I_{M,k} \), write the coordinates in binary form \( s_i = b_0 b_1 \cdots b_{N-1} = \sum_{n=0}^{k-1} 2^n b_{i,n} \). (iii) Prepare a qubit circuit with \( kM \) input qubits, labeled \( \{ \hat{\sigma}^{z,\text{in}}_{j,n} \}_{j=1}^{M} \) and \( N_c \) output neurons \( \{ \hat{\sigma}^{z,\text{out}}_{i,n} \}_{i=1}^{N_c} \) in the product state \( |\psi(s)\rangle := \otimes_{j=1}^{M} \otimes_{n=0}^{k-1} |b_{i,n}\rangle \otimes_{i=1}^{N_c} |0\rangle \).

(iv) Apply simultaneously the unitary transformation

\[
\hat{U} = \prod_{i=1}^{N_c} \exp \left[ i \sum_{j=1}^{M} w_{ij,n} 2^n \hat{\sigma}^{z,\text{in}}_{j,n} - \theta_i \right] \hat{\sigma}^{z,\text{out}}_{i,n}.
\]  

(6)

(iv) Reconstruct the approximation as \( Q(s) = \sum_{i=1}^{N_c} \alpha_i (1 + \langle \hat{\sigma}^{z,\text{out}}_i \rangle) \). This requires averaging over \( K_c \) realizations, so that the total statistical error \( \mathcal{O}(N/\sqrt{K_c}) \), remains bounded.

We prove this result in the supplementary material 30. The most important remark is that the number of neurons \( k \times M + N_c \) scales similarly as ordinary implementations of neural networks in classical computers: the factor \( k \) arises simply from the need to digitize the universal approximation theorem from Theorem 1 using bits instead of real numbers with arbitrary precision.
Implementation.——We construct the perceptron gate evolving a qubit with the Hamiltonian
\[ \hat{H}(t) = \frac{\hbar}{2} [-\Omega(t) \hat{\sigma}_x - x_j \hat{\sigma}_y] . \] (7)

The qubit is controlled by an external transverse field \( \Omega(t) \), has a tunable energy gap and interacts with other neurons through \( x_j = \sum_{j<k} w_{jk} \hat{\sigma}_k^+ - \theta_j \). The instantaneous ground state of \( \Omega \)
\[ |\Phi(x_j/\Omega)\rangle = \sqrt{1 - g(x_j/\Omega)} |0\rangle + \sqrt{g(x_j/\Omega)} |1\rangle \] (8)
has a sigmoid excitation probability [cf. Fig. 1, solid]
\[ g(x) = \frac{1}{2} (1 + x/\sqrt{1 + x^2}) . \] (9)

This suggests implementing the gate (1) in three steps: (i) set the perceptron to the superposition \( |+\rangle = \hat{H}(0) = 1/\sqrt{2} \langle 0 | + 1 \rangle \) with a Hadamard gate; (ii) instantaneously boost the magnetic field \( \Omega(0) = \Omega_0 \gg |x_j| \); (iii) adiabatically ramp-down the transverse field \( \Omega(t_f) = \Omega_f \) in a time \( t_f \), to do the transformation \( \mathcal{A}(x_j)|+\rangle \simeq |\Phi(x_j/\Omega_f)\rangle \).

As sketched in Fig. 2, the energy gap in this protocol is larger than \( \langle \Omega(t) \rangle \), ensuring many quasiadiabatic strategies \( \langle \Omega(t) \rangle \) to approximate \( \hat{U}_f(x_j, g) \simeq \mathcal{A}(x_j) \mathcal{H} \) for \( |x_j| \leq |x_{\text{max}}| \ll \langle \Omega_0 \rangle \). We compared two: a linear ramp \( \Omega(t) = \Omega_0 (1 - t/t_f) + \Omega_f t/t_f \), and a FAQUAD (Fast-Quasi-Adiabatic passage) control \( \Omega \) that limits nonadiabatic errors \( \delta \). As figure of merit we use the average fidelity
\[ \mathcal{F} = \int_{-x_{\text{max}}}^{x_{\text{max}}} \mathcal{F}[\Phi(x_j), \phi(t_f, x_j) \rangle |dx_j, \]
(10)

with \( \mathcal{F}(\Phi, \phi) = \langle |\Phi(x_j/\Omega_f)\rangle |\phi\rangle \rangle^2 \) and \( \phi \) the final dynamical state driven by \( \Omega(t) \).

Figure 3 compares the linear and FAQUAD strategies to modify the transverse field. In Fig. 3(a) we observe that for the same time \( t_f \) the FAQUAD protocol is more accurate; alternatively, given an error tolerance \( \epsilon = 1 - \mathcal{F} \), the FAQUAD design is 2-3 orders of magnitude faster than the linear ramp. From approximate fits, we estimate that the total time for a perceptron gate to have an error smaller than \( \epsilon \) scales as \( t_{f,\epsilon} = O(\log(\epsilon)^{1/0.15} \Omega_f^{-1}) \). When we have multiple neurons \( N \) spread over \( L \) layers, the gates of a single layer can be parallelized, keeping the total time bounded, but errors accumulate exponentially with the number of qubits. A more realistic scaling that takes this into account is \( T_{f,\epsilon} = O(L \times \log(\epsilon/N)^{1/0.15} \Omega_f^{-1}) \).

We can compare this performance with a proposal for implementing a quantum perceptron using auxiliary qubits, conditioned rotations and measurements \( 23 \). The gate implemented in that work is a rotation \( \hat{U} = \exp(i q(k)(x) \hat{\sigma}_y) \) with a nonlinear angle \( q(k)(x) = 2 \arctan(\tan^{\alpha_k}(x)) \) that converges to a step-wise function in the interval \( x \in [-\pi/4, \pi/4] \). This gate requires about \( k \) auxiliary qubits, a circuit depth \( O(14^k) \) and the total gate time scales polynomially \( O((n/\delta)^2) \) with the number of neurons per layer \( n \) and the step width \( \delta \simeq \Omega_f \) of the network. An important point in the work by Cao et al. is that it demonstrates algorithmic applications for neural networks that are perfectly discriminating —rotation angles take values close to \( \pi/2 \) or 0 and \( P_j \) is either 0 or 1, as in Eq. 2: those applications can also be reproduced with our own perceptron by a suitable design of

---

**FIG. 2.** Energy levels of the two-level system \( \Phi \) as a function of the activation potential \( x_j \). The perceptron gate begins with large transverse field, \( \Omega_0 \gg |x_j| \), such that the ground state is the approximate superposition \( |+\rangle \simeq |0\rangle + |1\rangle \). When the transverse field is decreased, the state converges to \( |\Phi(x_j/\Omega_f)\rangle \) given by \( \Phi \).

**FIG. 3.** (a) Transverse field \( \Omega(t) \) for the linear ramp (dashed) and FAQUAD (solid) protocols to implement the perceptron gate. (c) Average infidelity \( 1 - \mathcal{F} \) as a function of the total ramp time \( t_f \), for the two ramp protocols. The FAQUAD process is fitted by \( \sim c_0 \exp[-c_1 (\Omega_f t_f)^{c_2}] \), \( c_0 = 26.838 \), \( c_1 = 6.577 \), and \( c_2 = 0.150 \) (black circles).
The model of a quantum perceptron that we have introduced has other important ramifications, such as the design of complex controlled operations or the connection to quantum sensing sketched above. In particular, the image of the multi-layer perceptron circuit as a quantum sensor opens many interesting questions. For instance, how to define and optimize the sensitivity of these sensors? Can these threshold sensors be combined with other unitary operations, quantum states, etc? If so, what are the quantum limits of threshold sensing vs. ordinary sensing of classical fields? We expect to address these and other questions in future works.

We acknowledge funding from MINECO/FEDER Project FIS2015-70856-P, CAM PRICYT project QUITEMAD+CM S2013-ICE2801, and Basque Government (Grant No. IT986-16).
Supplementary material for “Universal quantum perceptron as efficient unitary approximators”

II. FAST QUASIADIABATIC DYNAMIC

Given the same boundary values $\Omega_0$ and $\Omega_f$ as for a linear ramp, we can engineer a rather fast control of $\Omega(t)$ that still achieves the target state $|\phi_1(t)\rangle$ for all $x_j$. The need to produce single controls independently on one Hamiltonian parameter automatically discards many of the existing methods that speed up adiabatic passages $|\phi_1(t)\rangle$.

However, there is one strategy of fast quasiadiabatic dynamics (FAQUAD) [31], which only works with the adiabatic parameter $\mu(t)$

$$\mu(t) = \hbar \left| \frac{\langle \phi_0(t) | \partial_t \phi_1(t) \rangle}{E_1(t) - E_0(t)} \right|$$  \hspace{1cm} (S8)

expressed in terms of the rate of change of the first excited state $|\phi_1(t)\rangle$ of $H(t)$ and the energy separation between the ground and excited states, $E_1 - E_0$ of a quasiadiabatic Hamiltonian. We will generalize this strategy, imposing conditions on $\mu(t)$ that are satisfied for all input fields and states of the neurons $x_j$, thereby designing the optimal controls for implementing this gate.

Our strategy will be to ensure that the adiabatic parameter remains constant $\mu(t) = c$ to delocalize the transition probability along the whole process. If the relation between field and time is invertible $t = t_f(s)$ and define $\tilde{\Omega}(s) := \Omega(s t_f)$ so that $d\tilde{\Omega}(s)/ds = t_f^{-1} d\Omega/ds$. This way,

$$\frac{d\tilde{\Omega}}{ds} = \pm \frac{\tilde{c}}{\hbar} \left| \frac{E_0 - E_1}{\langle \phi_0 | \partial_t \phi_1 \rangle_{\tilde{\Omega}}} \right|,$$

where the sign determines whether $\Omega(t)$ monotonously increases or decreases from $\Omega_0$ to $\Omega_f$. We rescale time according to the total duration $s = t/t_f$ and define $\Omega(s) := \Omega(s t_f)$ so that $d\tilde{\Omega}(s)/ds = t_f^{-1} d\Omega/ds$. This way,

$$\frac{d\tilde{\Omega}}{ds} = \pm \frac{\tilde{c}}{\hbar} \left| \frac{E_0 - E_1}{\langle \phi_0 | \partial_t \phi_1 \rangle_{\tilde{\Omega}}} \right|,$$

where $\tilde{c} = c t_f = \pm \hbar \int_{\tilde{\Omega}(0)}^{\tilde{\Omega}(1)} \frac{d\tilde{\Omega}}{\langle \phi_0 | \partial_t \phi_1 \rangle_{\tilde{\Omega}}}$.  \hspace{1cm} (S10)

To deduce $\tilde{\Omega}(s)$ for the FAQUAD protocol we solve Eq. $|\phi_1(t)\rangle$, choosing $\tilde{c}$ to satisfy $\tilde{\Omega}(0) = \Omega_0$ and $\tilde{\Omega}(1) = \Omega_f$. A different election of $t_f$ corresponds to a scaling of $c = \tilde{c} t_f$.
and \( \Omega(t = st_f) = \tilde{\Omega}(s) \). For the particular Hamiltonian \( H \) the instantaneous eigenstates and energies are given by,

\[
|\phi_i\rangle = \cos(\theta/2) |1\rangle + (-1)^i \sin(\theta/2) |0\rangle, \quad (S12)
\]

\[
E_i = -(-1)^i \sqrt{\Omega^2 + x_j^2/2}, \quad i \in \{0, 1\}, \quad (S13)
\]

where \( \theta = \arccos[-x_j/\sqrt{\Omega^2 + x_j^2}] \). Replacing Eq. \( (S12) \) into Eqs. \( (S10) \), the FAQUAD control \( \Omega(t) \) is deduced. However, this transverse field is different for different \( x_j \) values. The constant adiabatic parameter for the FAQUAD protocol is

\[
\mu = \left| \frac{1/\sqrt{1 + x_j^2/\Omega_0^2} - 1/\sqrt{1 + x_j^2/\Omega_f^2}}{2x_j t_f} \right|. \quad (S14)
\]

For the gate to succeed, we need a single control that does not depend on the neuron input potential \( x_j \). We notice that the largest value of \( |\mu| \) happens at \( |x_j/\Omega_f| \approx 1.272 \) providing us with an optimal definition of \( \mu(t) \) that works for all input neuron configurations. In fig. \( S1 \) the corresponding adiabatic parameter \( \mu \) is plotted for the linear and FAQUAD ramps as a function of \( t_f \). Whereas for the FAQUAD protocol \( \mu \) is constant along the whole interval, \( \mu \) changes in time in the linear ramp of \( \Omega(t) \) taking its maximum value at the end of the process \( t = t_f \) that corresponds to the minimum energy gap \( E_1 - E_0 \). As \( t_f \) increases both protocols become more adiabatic, however, for a fixed \( t_f \) value the FAQUAD strategy is more adiabatic allowing a sigmoidal excitation response in processes 2-3 orders of magnitude faster than with a simple linear ramp.