An artificial neuron model implemented on the IBM quantum processor

Dario Gerace

Dipartimento di Fisica, Università di Pavia

Workshop on QC and HPC – CINECA Bologna, 18/12/2018
Coworkers & Acknowledgments

F. Tacchino

D. Bajoni

C. Macchiavello

* The views expressed in this work do not reflect the official policy or position of IBM and the IBM-Q Team
Artificial neural networks (ANN)

possible model of brain: feed-forward network of interconnected signal processing elements

Each node mimics the functionality of a single neuron
The classical perceptron as a model of artificial neuron

\[ \sum_{j} i_{j}w_{j} \]

output

Activation function

the weights and the threshold function can be adjusted during learning phase

Rosenblatt, Psychol. Rev. 65, 386 (1958)
Applications of ANN

- single perceptron
  → a linear classifier

- ANN can classify complex data by recognizing patterns
  E.g. → image and speech recognition

- Evolution into A.I. → translate text, control vehicles, play games, ...
Quantum neural network models

➢ Idea: exploit quantum mechanics to enhance neural network computing capabilities

\[ |\psi\rangle = a|0\rangle + b|1\rangle \]

➢ Most algorithms are difficult to implement on NISQ (Noisy Intermediate Scale Quantum) devices

Schuld et al., Quant. Inf. Proc. 13, 2567 (2014)
Implementing the perceptron on a digital quantum computer

The key function

\[ \sum_j i_j w_j \]

Encoding input and weights

\[ i = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{2^N-1} \end{pmatrix} \quad \text{McCulloch-Pitts neuron} \]

\[ w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{2^N-1} \end{pmatrix} \]

\[ i_j, w_j = -1, +1 \]

\[ |\psi_i\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} i_j |j\rangle \]

\[ |\psi_w\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} w_j |j\rangle \]

Tacchino et al., arxiv:1811.02266 (2018)
Hypergraph states

\[ |\psi_i\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} i_j |j\rangle \]

Real Equally Weighted states

\[ \Downarrow \]

Hypergraph states

Algorithm for the generation of hypergraph states used to implement \( U_i \) and \( U_w \)

Implementing the perceptron on a digital quantum computer

Quantum algorithm: a circuit model

Constraints on the unitaries:

\[ |\psi_i\rangle = U_i |0\rangle^{\otimes N} \]
\[ |1\rangle^{\otimes N} = U_w |\psi_w\rangle \]

\[ |c_{2^N-1}|^2 = |\sum_j i_j w_j|^2 \]

\[ |0\rangle^{\otimes N} |0\rangle_a \rightarrow \sum_{j=0}^{2^N-2} c_j |j\rangle |0\rangle_a + c_{2^N-1} |2^N - 1\rangle |1\rangle_a \]

with \( c_{2^N-1} = \langle \psi_i | \psi_w \rangle \)

Tacchino et al., arxiv:1811.02266 (2018)
Pictorial representation

in vector space:
Pictorial representation

in vector space:

\[ U_w |\psi_w \rangle \]

\[ |11..11\rangle \]

\[ \langle \psi_i | \psi_w \rangle \]

\[ |10..11\rangle \]

\[ |00..00\rangle \]

\[ |01..01\rangle \]

\[ |10..00\rangle \]
Practical application: Recognizing patterns of pixels

\[ N = 2 \]

\[
\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \end{pmatrix} \quad \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}
\]

\[
\begin{array}{cc}
  i_0 & i_1 \\
  i_2 & i_3 \\
\end{array}
\]

+1 = white

-1 = black

\[
|\langle \psi_i | \psi_w \rangle|^2 = 1
\]

It’s me!

\[
|\langle \psi_i | \psi_w \rangle|^2 = 1
\]

It’s still me! (in negative colors)

\[
|\langle \psi_i | \psi_w \rangle|^2 = 0
\]

It’s not me!
Running the algorithm on a real Q-hardware

Exact result (N = 2)

\[
\sum_{i,j} w_{ij}^2 = 1 [0.84]
\]

Experiment (N = 2 + 1 ancilla)

\[
\sum_{i,j} w_{ij}^2 = 0 [0.07]
\]

IBM-Q Experience 5-qubit ‘Tenerife’ processor

Tacchino et al., arxiv:1811.02266 (2018)
Potential advantages

2^N-bit input and weight vectors can be encoded in ±1 factors in a balanced superposition of the computational basis states of N qubits.

Exponential advantage in storage

2^N bits       \rightarrow       N qubits

e.g. 20 qubits are sufficient to process 1024x1024 pixels (i.e. 1 Mpixel)
Summary

- Single perceptron efficiently implemented on 5-qubits IBM-Q hardware

- If scaled, this allows for an exponential scaling of encoding resources

- Further work: multilayer networks, continuously valued input/weight vectors…show quatum advantage