



HAL
open science

TRIZ, a Systematic Approach to Create Quantum Activation Function for Deep Learning's Hidden Layers, in Order to Make AI Explainable with Quantum Computer

Adehi Guehika

► To cite this version:

Adehi Guehika. TRIZ, a Systematic Approach to Create Quantum Activation Function for Deep Learning's Hidden Layers, in Order to Make AI Explainable with Quantum Computer. 19th International TRIZ Future Conference (TFC), Oct 2019, Marrakesh, Morocco. pp.371-387, <10.1007/978-3-030-32497-1_30>. <hal-02905521>

HAL Id: hal-02905521

<https://inria.hal.science/hal-02905521v1>

Submitted on 23 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

TRIZ, a systematic approach to create quantum activation function for Deep Learning's hidden layers, in order to make AI explainable with quantum computer.

Adehi Guehika¹[0000-0003-4078-5669]

¹ HEC Montréal , 3000 Chemin de la Côte-Sainte-Catherine QC H3T2A7, Montréal, CANADA

Adehi.guehika@hec.ca

Abstract. Artificial Intelligence (AI)'s market is growing very fast all over the world, along with Deep Learning (DL) technologies requiring more and more data and speed to process them, in healthcare, agriculture, automotive, security, and in among several other industries. However, despite this AI rapid and increasing market expansion, there are numerous related challenges need to be tackled. In this paper, the purpose is to show through another structured approach derived from ARIZ algorithm and some principles coined by TrizStartup movement [1], some of the problems deep learning industries are trying to handle. This paper shows how TRIZ can be applied to fix one of those AI problems. In fact, due to DL neural networks (NN) hidden layers, their outputs are not reliable. If an error occurs, how could human reproduce the issue? Human are not able to identify activated neurons to analyse root causes of this malfunction. Therefore, reliability issue from hidden layers versus rapid AI market skyrocket, data and speed processing, generates innovation problems. In order to resolve this, principle 35-parameter changes with Althshuler Matrix, little people, Su-Field analysis, seventy-six standards solutions, have been used to generate quantum functions. The result described in this paper is a new function called QuantumReLU (QReLU), created with Quantum computer in order to extend classical activation function ReLU. It is then possible to fire neuron with activation function QReLU, and to use quantum states to identify activated neurons within hidden layers. Thus, Triz systematic approach led switching neural networks algorithms from classical to quantum computer, and therefore to build deep learning NN on quantum computer, based on the new QReLU activation function.

Keywords: TRIZ, Deep Learning, Quantum computer, QuantumReLU

1 Introduction

This paper's purpose is not to discuss all deep learning and neural networks mathematical details and complexities. This was already well done by Goodfellow, Bengio and Courville [2]. Based on ARIZ algorithm and TrizStartup principles, this paper will be structured on nine main points, see figure 1 and table 1, to have a big picture.

Brief overview on the method or approach we used

We first started by hearing deep learning market voice (P1) in order to understand what the market is, who is leading the market, and in which region it is most likely used. This shows us how far African regions and others are let behind this ongoing AI and deep learning revolution and market expansion. Secondly, we defined some specific problems (P2) this deep learning market is facing in terms of needs or improvement issues. We selected one of those issues and focus on it for the next steps. Then, in the third point, we took decision (P3) to generate typical problems based on this chosen specific problems, in order to generalize it. This brought us to the fourth point concerning the typical problem definition (P4). And then, in the fifth one, we got some insights from innovators, experts or experiences around the world, out of deep learning scope (P5), in terms of grabbing ideas on what else could have been done elsewhere, outside DL markets in order to fix this typical problem with innovative and typical solution (P6). Then in the seventh section, we went through training phases (P7) to master some deep learning tools, technics software from current trends, to transition from typical solutions to specific solution. Then, in the point height, we proposed a specific solution (P8) to apply and fix the initial specific problem we point out earlier. This proposal was tested and validated internally in a short scale. The section nine, was the last point. It covered how the proposed solution could transform deep learning market (P9). We opened discussions and concluded.



Fig. 1. Perpetual innovation's transformation wheel, adapted from TrizStartup principles

Table 1. High level organization of the steps used for analysis on this current paper.

Steps	Description	Flow used
P1	Hearing deep learning's market voices	Decision flow
P2	Define deep learning specific problem	Action flow (ARIZ)
P3	Take critical or management decisions	Decision flow
P4	Generate typical problem	Action flow (ARIZ)
P5	Get some insights by learning from innovators, experts, or from elsewhere out of DL scope and environment.	Decision flow
P6	Define typical solution based on P5	Action flow (ARIZ)
P7	If necessary, train your team to master TRIZ, otherwise call TRIZ experts for assistance	Decision flow
P8	Find your specific solution based on P6 and P7	Action flow (ARIZ)
P9	Now, with your specific solution on P8, go and transform your DL market, and if possible, impact the World	Decision and action flow (P9 and (P2) to reinitiate the wheel

2 Analysis

2.1 P1 - Hearing deep learning market voices: the rapid AI market's skyrocket expansion and processing speed increased

PricewaterhouseCoopers [3] shows that global World GDP could be increased to \$15.7 trillion, by 2030, due to Artificial intelligence technologies. See table 2 to notice how far Chinese AI GDP is, compare to the other countries, and even regions of the World. In this, \$1.2 trillion would be added for Africa and others .

Table 2. Importance of Chinese AI GDP compare to Others, adapted from PwC.

Regions	GDP in Trillion \$ US by 2030
China	7,0
North America	3,7
Northern Europe	1,8
Africa, Oceania and other Asian Market	1,2
Developed Asia (without China)	0,9
Southern Europe	0,7
Latin America	0,5

Inside AI demand growth, there are also some initiative which are led by deep learning figure 2. And concerning Deep Learning per say, some researches conducted by companies such as Markets and Markets (MM) [4], and Market Research Future (MRF) [5], expressed some projections. Deep learning's market was worth USD 2.28 Billion in 2017, then 3.18 Billion in 2018 and is expected to reach USD 18.16 Billion by 2023. For the MRF, the research was structured by component, application (Image recognition, signal recognition, data mining), End-User industry (automotive, healthcare, manufacturing, retail, agriculture, security, marketing). And for MM: The research methodology was based on studying annual and financial reports, presentations, press releases related to DL top players. They also discussed with experts and analysed some specialized white papers such as The Zettabyte era regarding the trends and Analytics. They both came out with similar results. **Deep learning compound annual growth rate (CAGR) will jump quickly to reach 41.7% from 2018 to 2023.** According to the previous studies, there are three segments in deep learning market. The first segment is related to component industry, and is also sub-segmented into hardware, software and services. The second one is about application, and the last one concern end-user industry. From the component segment, the software sub-segment will hold the biggest share during the forecast period 2018-2023. But in the meantime, services sub-segment will be growing with highest rate. In the software sub-segment for example, emphasis will be made on software frameworks and platforms/APIs with algorithms and codes available for hardware to perform deep learning programs. The following non-ordered list shows some of the main open platforms to test deep learning algorithms, assertions, and codes: TensorFlow [6], PyTorch [7], Keras [8], Apache MxVet [9], Caffee [10], Caffee2 [11], DeepLearning4J [12], and so on. Figure 3 shows how DL market is shared in different regions trough the World:

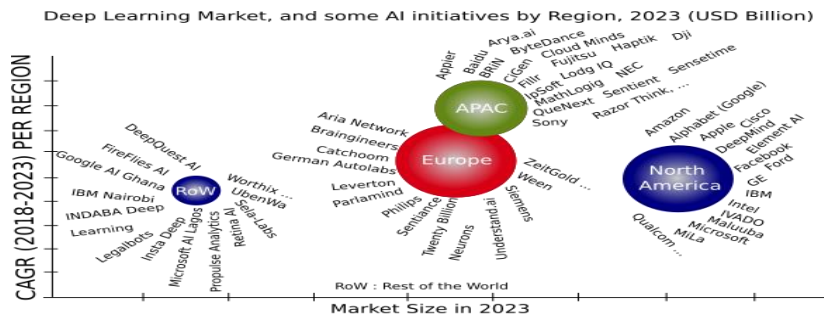


Fig. 2. DL market segments by region, based on MM

Figure 2 shows that deep learning initiatives are still so weak in rest of the World (RoW) region including Africa, and Latin America, even though, those regions could hugely work to generate their own DL applications based on biases adapted to cultural and

local realities. Indeed, for emerging countries whose economies are based on agriculture, oil and gas exportations, for example, there are numerous possibilities to create deep learning applications to leverage their development. In agriculture, DL is widely used in several areas such as plant diseases early detections and diagnosed with image recognition, weed identification, plant classification, weather forecasting, yield prediction, fruit counting [13]. And DL could also be used for land and cover classification, soil moisture estimation, animal research and so on. In the next section we discussed some important challenges deep learning's market and industry are currently facing.

2.2 P2 - Defining deep learning specific problem

Quick overview on deep learning neural networks main principle

Machine learning and Deep Learning are both branches of Artificial intelligence. Deep learning is a kind of crude imitation of how human brain works. It is also a kind of representation learning. And this one is also a sort of machine learning. See how figure 3 is showing the Venn diagram describing the relations between those concepts.

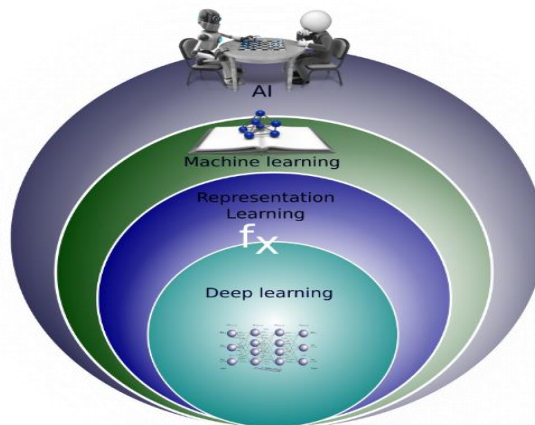


Fig. 3. AI to Deep Learning, adapted from Venn diagram, Goodfelow et al

Difference between Machine learning (ML) and Deep learning (DL)

The difference between them resides in the following. Machine learning uses algorithms to split data into significant parts and process them to take decision. Machine learning (ML) include supervise learning in which a human feed the machines and teach them with correct input-output. ML include also un-supervised learning through which a machine deduced its own output from recognized pattern, while given input were provided. ML use also reinforcing learning to observe gathered information while interacting with its environment and take actions in order to reduce risk and to optimise its rewards. DL is a subfield of ML. DL uses a layered structured algorithm so-called artificial neural network (NN), inspired by human brain's biological network, in order to analyse data with a kind of human logic, to draw its conclusions. But in case of

inaccurate predictions, while ML need engineers or human operators to make some adjustments, DL can verify by it own if its prediction is accurate or not, by making intelligent human-like decisions. Basically, Deep learning classifies input information using layered neural networks. And those neural network (NN) are composed with multiple k-hidden layers of neurons fully connected between the input and output nodes.

Deep learning, speed versus reliability issues

There are numerous challenges discussed by AI experts on deep learning. But for the purpose of this paper, we will focus on some of them. In fact, details on deep learning limits and challenges were provided by Gary Marcus [14], professor at New York University and former AI head at Uber. There are also other issues such as the ones explained by professor Kunle Olukotun [15] from Stanford University whose explanation could be also benefitted to understand current Deep Learning limits.

- Deep Learning need data, thousands, billions, and now Zettabytes of data, in order to train their neural networks algorithms and to easily classify their outputs. (Problem 1)
- Speed of computation is now limited, according to Olukotun: «You’ve got these huge [computational] demands, but you have the slowing down of Moore’s law,” Olukotun said. “The question is, how do you meet these demands while Moore’s law slows» [16]. And for Greg Diamos [17] senior researcher at Baidu : «Today the job of training machine learning models “is limited by compute, **if we had faster processors, we’d run bigger models**» (Problem 2)
- Deep learning lacks a mechanism to understand abstraction explicit verbal definition. It works better when there are thousands, millions or even billions of training examples, said Marcus (Problem 3).
- Deep learning is opaque, and **there is no transparency in the decision they make** based on patterns and correlations they generate through millions of data. This opacity can be critical in several scenario, and for example, medical treatment for patient, if any mistakes are made. **For Marcus, this lack of transparency is a potential liability** for financial trades, medical diagnosis, in which **human would like to understand how a given system made a given decision.** (Problem 4)

Our selected deep learning specific problem

We reformulate all those issues as follow: As per Diamos, we want to run big models with neural networks, therefore we need to increase processors speeds. But in the meantime, we need to overcome this lack of transparency and any potential liability, as per Marcus, thus we need to preserve reliability. So, we want to improve the computing speed, but by doing so, the reliability of the neural network outputs becomes more and

more worst, since this lack of transparency will be increased. How can we fix this innovative issue? Which decisions could we take in order to have this issue been resolved?

2.3 P3 - Tacking decision: Where to go? Or which innovation directions to use so that we can fix this specific problem?

In this section we used innovation's 3 dimensional representations [18] shown in figure 5, in order to decide which direction was helpful to fix the specific problem defined in the previous section.

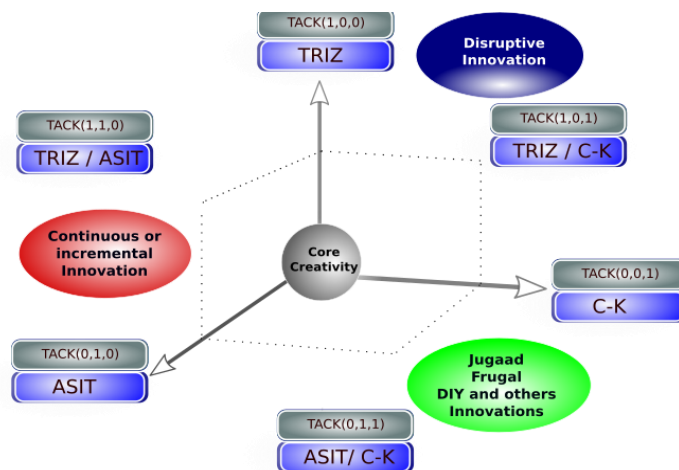


Fig. 4. Innovation directions, adapted from TrizStartup movement

Here:

- C-K [19] is defined as Concept – knowledge theory, a design theory to create the unknown,
- ASIT [20] is a method to help companies use their in-box creativity in order solve innovation problems. ASIT comes from TRIZ
- TRIZ 'Teoriya Resheniya Izobreatatelskikh Zadach' is the (Russian) acronym for the "Theory of Inventive Problem Solving." Initiated by G.S. Altshuller. [21]

The decision to use TRIZ, ASIT, or C-K, or any combination of those innovation and creativity theories, tools, and method depend of the following scenarios:

- Scenario 1: If the issue needs to be resolved by something based on completely unknown expression, then we use C-K to create that unknown by digging alternately

from Concept to Knowledge, and vice-versa, until we reach knowledge or concept that can be used to fix the issue.

- Scenario 2: If the issue is formulated by a known expression, and if this issue required focusing on companies or organizations' internal resources to leverage their in-box creativity, by finding solutions from the problem's zone, then ASIT will be the direction to use to innovate.
- Scenario 3: If the formulated issue is with known expression, and if this time, we need to go out of the box, beyond our internal resources, and our engineering comfort zone, to grab some insights elsewhere in order to have the issue been resolved, then we will use TRIZ, as innovation direction.
- Between those extreme scenarios, we defined vector TACK (X, Y, Z), to position in 3 Dimensions, the decision regarding which innovation direction will be used to solve the issue. In fact, TACK stand for Triz Asit and C-K. From those TACK vectors we create 3 orthogonal innovation plans as following:

Breakthrough innovation plan:

Led by TRIZ: Tack (1,0,0) and C-K: Tack (0,0,1), through the following PLAN:

$$PLAN_{Breakthrough\ Innovation\ (TRIZ;C-K)} = P(Tack(1,0,0); Tack(0,0,1)) \quad (1)$$

Continuous or incremental innovation plan:

Led by ASIT: Tack (0,1,0) and TRIZ: Tack (1,0,0), through PLAN:

$$PLAN_{Incremental\ Innovation\ (TRIZ;ASIT)} = P(Tack(1,0,0); Tack(0,1,0)) \quad (2)$$

Other innovation Jugaad, frugal innovation and others initiative such as DIY etc., pan:

Led by ASIT: Tack (0,1,0) and CK: Tack (0,0,1), through PLAN:

$$PLAN_{Jugaad;Frugal;Others\ (ASIT;C-K)} = P(Tack(0,1,0); Tack(0,0,1)) \quad (3)$$

As reminder, the specific issue was described as follow: *we want to improve the speed for computations for data processing. But, by doing so, the reliability of the neural network output becomes more and more worst, since this lack of transparency will be increased.* This issue in this case is about known expression with concepts such as «transparency» or «opacity» issue in neural networks due to hidden layers. So, we can use ASIT or TRIZ. In addition, this issue is a global issue, and it is not an issue that need to be dealt by a single company or organization, attempting to resolve it with their in-box creativity. Therefore, ASIT will not be used in priority. Instead, we will be starting with TRIZ to fix this issue. And if we want to generate a breakthrough innovation, we can use only TRIZ, or combine it with C-K, in the $PLAN_{Breakthrough\ Innovation\ (TRIZ;C-K)}$. And in this case any combination of TRIZ:

Tack (1,0,0) vector and C-K: Tack (0,0,1) vector could be used. But for concision in this paper, we used, only TRIZ to fix this specific DL speed versus reliability issues.

2.4 P4 - Generating typical problem and suggestions: Reliability issue

In every single field where Deep Learning is used, it is making significant strides. This is true in speech recognition with deep recurrent neural network [22]. It is also true in image search without tag performed with deep convolution neural network. Today's search ranking are made with deep neural network, email are smartly replied with feed-forward neural network combined with deep recurrent neural network. All those achievements made with deep learning, demonstrate how far and fast DL is expanding. We first used reinvention technics from Michael Orloff [23], to model and represent the specific issue. Secondly, we also used Su-Filed technique to represent another model of the same issue.

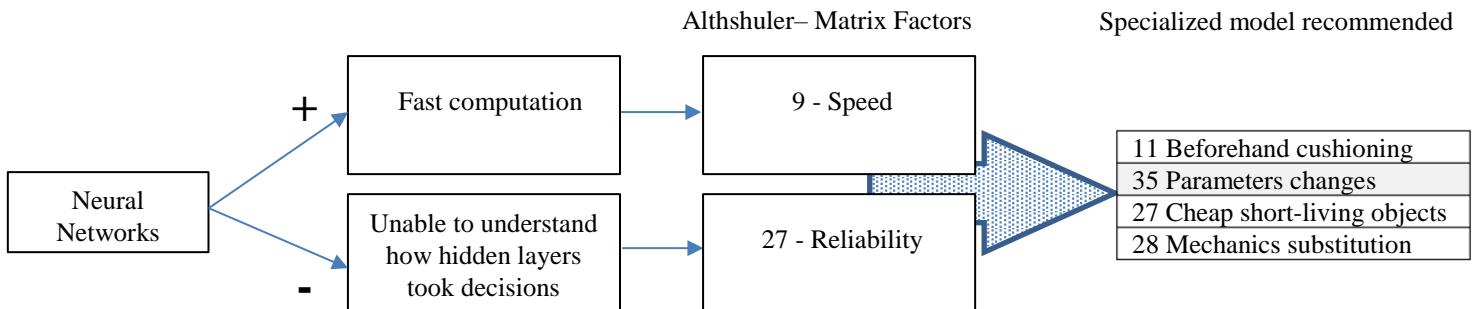
First model of the typical problem adapted from reinvention technics from Orloff [23]

Reinvention: eliminating neural network layers' opacity

Trends Deep Learning neural networks required huge amount of data to perform their training and classification. And, to do so, the computing process consume time and energy. Therefore, the processor speed needs to be improved. But in the meantime, it is completely not feasible to see what happening inside those neural networks' algorithm. The hidden layers and their corresponding neurons reduce output's results reliability.

Reducing Macro-FIM : By doing X-Thing, the initial system is modified. It doesn't create new problems, and assures that the following result is obtain:
IFR: [The neural networks hidden layers are visible]

Standard contradictions (SC), or Technical contradictions



Radical contradiction (RC), or physical contradiction

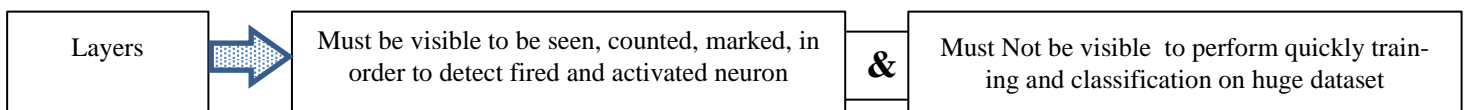


Table 2. Zooming expected for typical solution

Expected Zooming on the target typical solution	
Contradictions removed	Yes (this is a target decision we expected to get at the end)
Super-effects	The layers are visible, and now the outputs are reliable.
Negative effects	The targeted typical solution will require digging on something hidden and invisible for human.
Development trends	Possibility to create a path inside the neural network for any single output, from input feature to last output feature. And this could be done on all activated neurons inside the hidden layers. And any single neuron path will therefore make those layers become visible.
Changes in the Ambient system	Activated neuron are visible, hidden layers are visible, and can be plot, extract, and interpretable.
Extended uses	We can define new kind of jobs, for example, as Deep Learning Auditors, in case of trouble, they will be able to extract those paths, like working in a Plane Black-box after a crash, to analyze the reliable output, and at which point, on which layer, the issue emerged, and which neuron were involved in the issue, so that correctives actions could be taken for the current situation, but also to avoid them in the future.
Beauty of the solution	Now the invisible world become visible in the deep learning universe.

Second model used to generate typical problem by using Sub-Field analysis

The current system needs to be improved. By digging in the 76 standard solutions, Class 2 and Class 3 are dominant classes to use to get a typical solution for this typical problem. In fact, as per Terninko, Domb, and Miller [24], Class 2 has some model to improve the system by changing it. And Class 2, section 2.2.2 gives us typical solution. This is also true for Class 3 as well. Typical solution from Class 2 is about: Developing the Substance-Field System. And its section 2.2.2 is about: changing substance from a macro level to a micro level. Typical solution from Class 3 is about the System Transitions. And its section 3.2.1. is about: Transition to the Micro-Level

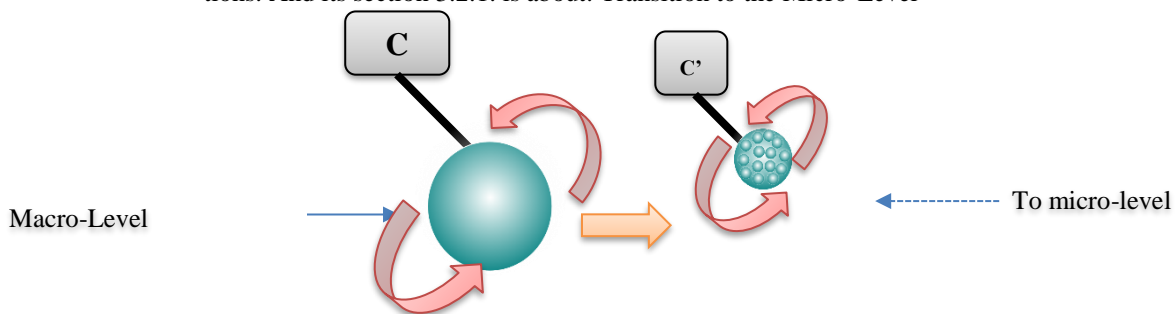
**Fig. 5.** Applying Class 3, section 3.2.1

Table 3. Summary of suggested Typical Solution determination

Technic	Type	Suggested typical solution	Code
Contradiction	Standard (SC)	Change an object's physical state	Suggestion 1
	Radical (RC)	Move to the super-system or the sub-system	Suggestion 2
Su-Field Analysis	Class 2 / 2.2.2	Changing substance from a macro level to a micro level.	Suggestion 3
	Class 3 / 3.2.1	Transition to the Micro-Level	Suggestion 4

2.5 P5 - Getting some insights by learning from innovators, experts, or from elsewhere out of DL scope and environment and grabbing suggestions for typical solution.

The idea behind getting those insights is to narrow the suggested Typical solutions in order to focus to the feasible one, based on what else have been done elsewhere, we selected the following suggestion: Selected suggestion: Change an object's physical state. Here we have: Neural Networks -> Layers -> Neurons, which are built on classical computer algorithm. Those ones are running programs based on bits 0 and 1 which are lowest information unit used in the system. So, changing the physical state of the Su-filed will be related to changing the physical state of this single unit of the system. So how can we change a physical state of something that is binary having states only equal to 0 or 1?

3 The Results

3.1 P6 - Talking about typical solution: Quantum information and Qubits, the solutions?

IBM lunch IBM Q [25], a free platform to test and write freely coded on quantum programming. Google asked, last year, for NASA's help to prove quantum supremacy within a matter [26]. The idea behind quantum supremacy is the challenge to prove quantum computer with sufficiently powerful should be able to perform easily where powerful classical machines failed to do so on certain mathematical calculation. At NASA, some benchmarking test on both quantum computer and on supercomputer, were conduct by Benjamin Villalonga [27] at the Quantum Artificial Intelligence Lab at NASA. In this way, they were able to compare performances and to conclude the supremacy of quantum computer. IBM provide also Qiskit, an open-source framework for quantum computing. Several other platforms exist to test quantum computer with ease. We emphasis on Qubit, because Today's computers are working on bits of

information. Those bits have only two states: 0 or 1, as we mentioned before. But quantum computers possess more than two states; they use quantum bits so called Qubits, to encode information. Those Qubits can represent electrons, photons, atoms, ions. All those are working together to produce the computer processor and memory. Quantum computer can contain all these multiple states simultaneously. Thus, those quantum computers are fare and millions of times more powerful than most popular supercomputers.

3.2 P7 - Training the team to transition from typical to specific solution

According to the previous explanation provided on this paper, related to suggested typical solution (suggestion 1,2,3, and 4), we decided to put our emphasis on trying to find a specific solution in quantum world. But to deal with quantum computing combined to deep learning, we used TRIZ nine's windows, DTC operators to break the psychological inertia [28]. Moreover, concerning deep learning and quantum computing programming, we used tools and software such as python 3.7 [29], pythorch [30]; and for Quantum computing, online resources such as Qiskit [31] [32].

3.3 P8 – Building the specific solution.

The solution: Inventing QReLU activation function, the ReLU equivalent for Quantum Computers

We moved from micro level to quantum word. We used Qubits instead of bits to code in Qiskit, the information. Since qubits have more than two states, we parse a pointer as controller in the other available states. This parsed pointer is dynamically created each time a neuron is activated or fired. The pointer is located on the Bloch sphere [33] and we have several options to control those parses pointers with Ion traps, Optical traps, Quantum dots, Semiconductor impurities, Superconducting circuits, etc.

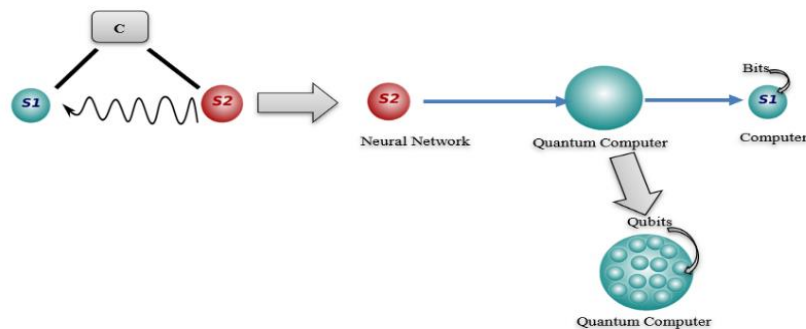


Fig. 6. Neural Network: from classical to quantum computer

How do we finally come to the solution?

We adapted artificial neural network structure combined with Bloch Sphere. We used Little people technic [34] to project ourselves inside the neuron, and in hidden layers:

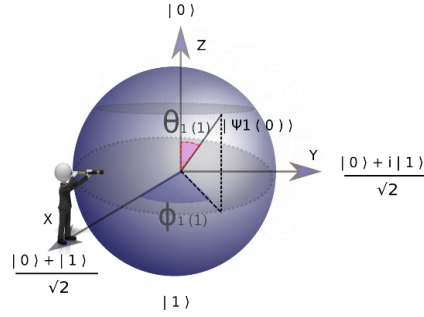


Fig. 7. Little people looking inside neuron as on a Bloch Sphere

We understood that little man, when inside the neuron, can see how this neuron structure is newly designed shaped with Bloch Sphere attributes. Same things inside the hidden layers.

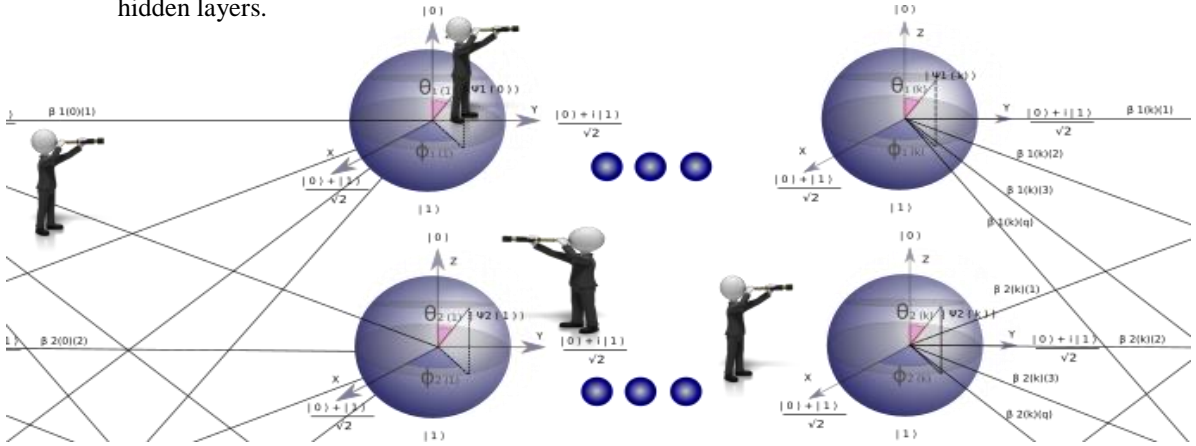


Fig. 8. Little people looking inside neural networks and hidden layers as in front of multiple Bloch spheres

Our new QReLU function: The mathematical formula

The newly activated function will be call QReLU (Quantum ReLU or Quadratic ReLU), where:

$$|\Psi_j(0)\rangle \text{ are input, for } j = 1 \text{ to } n, \quad (4)$$

$$|\Psi_j(k)\rangle = QReLU(\sum_{i=1}^p |\Psi_j(k-1)\rangle \cdot \beta_{i(k-1)(k)} + b_{jk}), j=1 \text{ to } q \quad (5)$$

b_{jk} , are complex biases added to activation function $|\Psi_j(k)\rangle$ for $j=1$ to q

$\beta_{i(k-1)(k)}$, are complex numbers corresponding weight of previous activated neurons

$$QReLU\left(|\Psi_j(k)\rangle_{(a_{jk}, b_{jk}, c_{jk})}\right) = \begin{cases} \mathbf{0}, & \left\| |\Psi_j(k)\rangle_{(a_{jk}, b_{jk}, c_{jk})} \right\| < \mathbf{1} \\ |\Psi_j(k)\rangle_{(a_{jk}, b_{jk}, c_{jk})}, & \left\| |\Psi_j(k)\rangle_{(a_{jk}, b_{jk}, c_{jk})} \right\| \geq \mathbf{1} \end{cases} \quad (6)$$

And,

$|\Psi_j(f)\rangle$ are output, for $j = 1$ to w

How we created QReLU on a quantum computer, using Qiskit and python?

```

from numpy import linalg as LA
class QReLU():
    def _init_(self):
        pass
    # Let define a simple forward for our basic QReLU
    def forward(self, input):
        #vect = np.vectorize(complex)(input)
        p = LA.norm(input)
        if p < 1 :
            QReLU_forward = 0
        else :
            QReLU_forward = input
            #QParse_pointer_f(QReLU_for-
            ward,QClassifier(QReLU_forward))

        return QReLU_forward

    # Now let define a simple backward function with basic
    QReLU

    def backward(self, input, bwg_output):
        r = LA.norm(input)

        if r > 1 :

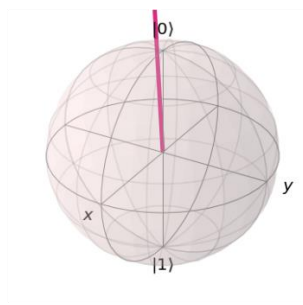
```

```

    QReLU_bw = np.array(input)
    #QParse_pointer_bw(QReLU_bw,QClassifier(QReLU_bw))
else :
    QReLU_bw = 0
return QReLU_bw*bwg_output

# Testing the forward function
qrel1 = QReLU()
b=qrel1.forward([0,-0.25,4.5])
b
# QReLU's forward function Visualisation on a Bloch
sphere:
qiskit.tools.visualization.plot_bloch_vector(np.array(b))

```



#This previous neuron will be fired and activated.

Fig. 9. Visualization of forward function on Bloch sphere

```

# But the following neuron will not be activated
qrel1 = QReLU()
b=qrel1.forward([0,-0.25,0.5])
b
>>>> 0
# Now, testing the backward function:
qrel2 = QReLU()
c=qrel2.backward([-3,0.25,5.05],0.2)
c
# QReLU's backward function Visualisation:
qiskit.tools.visualization.plot_bloch_vector(np.array(c))

```

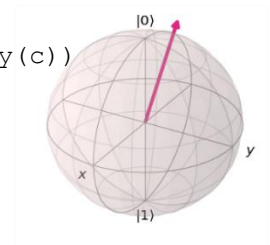


Fig. 10. Visualization of backward function on Bloch sphere

3.4 Discussion on the method contributions and limitations

This method contributes to work on neural networks hidden layers, and therefore open the gates for new opportunities with new kind of jobs created. For example, Deep Learning Auditors, in case of NN malfunction or error in prediction, or any trouble with outputs, they will be able to extract significant paths. And, like working in a Plane Black-box after a crash, DL Auditors could analyze the reliable outputs, and activated neurons involved in the issue, so that correctives actions could be taken in the current situation, but also to avoid similar issues in the future. This reliability will give also opportunity to human to really have control or at least, will be able to investigate the root cause of any issue that will be made by machine. This method also extends ARIZ algorithms with intermediate steps in terms of decisions, directions, and actionable tools to take or to make between ARIZ blocks.

But this method presents some limitations. For example, to come to this QReLU function, we made some assumptions. Since in classical ReLU $(x) = \max(0, x)$ negative x were not fired. Only positive x was fired. So, we assumed that, in a Bloch sphere, with complex values, and qubits, inspecting negative values is useless. Instead, only neuron x with Frobenius [35], or others $\text{norm}(x) \geq 1$ will be fired and activated. We used qubits to construct these neurons, and to evaluate their norms in order to activate or not the neuron. So how could we deal with neuron x inside the Bloch sphere near border where $\text{norm}(x) = 1 - \epsilon$, and $\epsilon > 0$, and with ϵ , a float number close to 0?

Conclusion:

In contrast to other researches on Deep Learning using trial and errors, we used TRIZ ARIZ algorithm, and TrizStartup methodology, to structure our thinking on this subject. This allows us to dig on Speed versus Reliability issues, and to come with a so-called QReLU, Quantum version of the classical ReLU function. TRIZ tools and technics could, therefore, be used for further systematic researches on deep learning and on AI development for helping open in systematic ways, new frontiers with Quantum computers. This will make Artificial Intelligence more and more explainable, to preserve human control over machines in the future of Humanity.

References:

1. Trizstartup. <http://trizstartup.com/index.php/en/home/> Accessed 8 Apr. 2019.
2. Goodfellow, I., Bengio, Y. and Courville, A.: Deep Learning. Cambridge, MA: MIT Press (2017).

3. Pwc.com.: Sizing the prize: PwC's Global Artificial Intelligence Study.
<https://www.pwc.com/gx/en/issues/analytics/assets/pwc-ai-analysis-sizing-the-prize-report.pdf>. Accessed 12 Apr. 2019
4. MarketsandMarkets.: Deep Learning Market: Analysis Size, Growth, Trend and Forecast to 2023. <https://www.marketsandmarkets.com/Market-Reports/deep-learning-market-107369271.html> Accessed 11 Apr. 2019.
5. Marketresearchfuture.: Deep Learning Market Research Report – Global Forecast to 2023. <https://www.marketresearchfuture.com/reports/deep-learning-market-6058> Accessed 11 Apr. 2019.
6. TensorFlow. <https://www.tensorflow.org/> Accessed 14 Apr. 2019.
7. Pytorch. <https://pytorch.org/> Accessed 1 Apr. 2019.
8. Keras Home Documentation. <https://keras.io/> Accessed 10 Apr. 2019.
9. MXNet. <https://mxnet.apache.org/> Accessed 9 Apr. 2019.
10. Caffe.: Deep Learning Framework. <https://caffe.berkeleyvision.org> Accessed 12 Apr. 2019.
11. Caffe2. <https://caffe2.ai/> Accessed 1 Apr. 2019.
12. Deeplearning4j. <https://deeplearning4j.org/> Accessed 11 Apr. 2019.
13. Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., Jiang, Y. and Guo, Y.: Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. <https://ijabe.org/index.php/ijabe/article/view/4475> Accessed 14 Apr. 2019.
14. Marcus, G.: Deep Learning: A Critical Appraisal. solving. <https://arxiv.org/abs/1801.00631> Accessed 11 Apr. 2019.
15. Stanford. <https://engineering.stanford.edu/people/oyekunle-olukotun/> Accessed 11 Apr. 2019.
16. Lynley, M.: The red-hot AI hardware space gets even hotter with \$56M for a startup called SambaNova Systems. <https://techcrunch.com/2018/03/15/the-red-hot-ai-chip-space-gets-even-hotter-with-56m-for-a-startup-called-sambanova/> Accessed 12 Apr. 2019.
17. EETimes.: Baidu Releases. In: Intel, Nvidia post results training neural nets. https://www.eetimes.com/document.asp?doc_id=1330521# Accessed 14 Apr. 2019.

18. Guehika, A.: Cartographie des processus d'innovation.(2014). <https://www.amazon.com/Cartographie-processus-dinnovation-lInnovateur-French-ebook/dp/B00HVIC53Y> Accessed 13 Apr. 2019.
19. The C-K theory. <https://www.ck-theory.org/c-k-theory/?lang=en> Accessed 15 Apr. 2019.
20. SolidCreativity.: ASIT. <http://www.asit.info/> Accessed 16 Apr. 2019.
21. Altshuller, G.: Creativity as an Exact Science. New York, NY: Gordon & Breach (1984).
22. D2L.: Deep recurrent neural network. https://d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html Accessed 12 Apr. 2019.
23. Modern Triz Academy. <http://www.modern-triz-academy.com/> Access 11 Apr. 2019.
24. Terninko, J., Domb, E., Miller, J.: The Seventy-six Standard Solutions, with Examples—Class 2. (2000). <http://www.metodolog.ru/triz-journal/archives/2000/03/d/index.htm> Accessed 19 Apr. 2019.
25. IBM Q Home page. <https://www.research.ibm.com/ibm-q/> Accessed 20 Apr. 2019
26. arXiv, E.: The new benchmark quantum computers must beat to achieve quantum supremacy.In: MIT Technology Review. <https://www.technologyreview.com/s/613507/the-new-benchmark-quantum-computers-must-beat-to-achieve-quantum-supremacy/> Accessed 19 Apr. 2019.
27. Villalonga, B., Boixo, S., Nelson, B., Henze, C., Rieffel, E., Biswas, R., Mandra, S.: A flexible high-performance simulator for the verification and benchmarking of quantum circuits implemented on real hardware. (2000). <https://arxiv.org/pdf/1811.09599.pdf> Accessed 18 Apr. 2019.
28. Frenklach, G.: DTC operators. <https://triz-journal.com/efficient-use-dtc-operator/> Accessed 19 Apr. 2019.
29. Python. <https://www.python.org/> Accessed 22 Apr. 2019.
30. Pytorch. <https://pytorch.org/> Accessed 01 May. 2019.

31. Boyer, J.: Using Quantum Operations to Achieve Computing Objectives. (2018). <https://medium.com/qiskit/using-quantum-operations-to-achieve-computing-objectives-868a7f1cfa09> Accessed 01 May. 2019.
32. QisKit.: An open-source quantum computing framework for leveraging today's quantum processors in research, education, and business. <https://qiskit.org/> Accessed 03 May. 2019.
33. Kyrillidis, A.: Introduction to quantum computing: Bloch sphere. http://akyrillidis.github.io/notes/quant_post_7 Accessed 25 May. 2019.
34. Livotov, P.: Modelling 40 Innovation Principles with "Moving Little People". (2003). <https://triz-journal.com/modelling-40-innovation-principles-moving-little-people/> Accessed 28 May. 2019.
35. Mathworld. <http://mathworld.wolfram.com/FrobeniusNorm.html> Accessed 17 May. 2019.