

Projeto de Graduação



December 16, 2022

Q-NAS Applied to the Classification of Medical Images

Marina Moraes da Silveira



www.ele.puc-rio.br



Q-NAS Applied to the Classification of Medical Images

Student: Marina Moraes da Silveira

Advisors: Marley Maria Bernardes Rebuszi Vellasco
Karla Tereza Figueiredo Leite

Acknowledgements

First, I would like to thank my advisor Professor Marley Vellasco and co-advisor Professor Karla Figueiredo for their guidance and understanding thought this this project.

I would also like to thank PUC-Rio and École Centrale de Lyon for all the opportunities that I was able to have during my years as a student at both universities.

I wish to thank my parents, who supported me in all my decisions and was always there to offer advice when I needed. I will be forever grateful that they gave me wings to fly and follow my dreams, but also created a nest I know I can always go back to.

I would like to thank Giovanna Montoro and João Victor Mey for their company since the first university semester. They were always there when I needed and never failed to make me laugh.

To the Reptiles Team, that completely changed my perspective on university, made me a better professional and person. I will be forever grateful for the friends I made during those two years and helped me go through university.

To my exchange program friends who were by my side during the most amazing and challenging experience of my life. The two and a half year abroad were truly life changing.

To my school friends, thank you for all the support and for always believing in me, even when I didn't.

Finally, I wish to thank to everyone who contributed to my graduation.

Abstract

This undergraduate thesis consists in classifying images obtained from chest computed tomography (CT) scans from patients who have had COVID-19 before by using Neural Networks.

The Q-NAS model is a quantum inspired algorithm to search for deep networks by assembling substructures. The basic premise of a NAS model (Neural Architecture Search) is the capability of automatically generating and searching the best neural network architectures, without requiring advanced machine learning knowledge from the user. The Q-NAS has the same premise but using quantum physics paradigms which improves the accuracy and the convergence time.

Because of these advantages, the Q-NAS model was applied to the CT images and classified them in six different classes according to the post-covid lung pattern found. The purpose of this undergraduate project is to generate new neural networks capable of classifying the post-covid patterns with a new database and test those models by using new inputs that were obtained from Pedro Ernesto University Hospital's patients.

Keywords: Neural Networks, Post-Covid, Neural Architecture Search, Medical Images Classification, Artificial Intelligence.

Resumo

Este Trabalho de Conclusão de Curso consiste em classificar as imagens obtidas a partir de tomografias computadorizadas do tórax de pacientes que já tiveram COVID-19 usando redes neurais.

O modelo Q-NAS é um algoritmo de inspiração quântica para pesquisar redes neurais através da montagem de subestruturas. A premissa básica de um modelo NAS (*Neural Architecture Search*) é a criação de um método automatizado de criação e pesquisa da arquitetura de redes neuronais, sem a necessidade de conhecimento avançado de *machine learning* por parte do usuário. O Q-NAS tem a mesma premissa, mas utilizando paradigmas de física quântica que melhoram a precisão e o tempo de convergência.

Devido a estas vantagens, o modelo Q-NAS foi aplicado às imagens CT e classificou-as em seis classes diferentes, de acordo com o padrão pulmonar pós-covid encontrado. O objetivo deste projeto de graduação é gerar novas redes neurais capazes de classificar os padrões pós-covid com uma nova base de dados e testar esses modelos, utilizando novos inputs que foram obtidos dos pacientes do Hospital Universitário Pedro Ernesto.

Palavras-chave: Redes Neurais, Pós-covid, Pesquisa de Arquitetura Neural, Classificação de Imagens Médicas, Inteligência Artificial.

Summary

1. Introduction	1
a. Motivation	1
b. Objectives	1
c. Organization	2
2. Theoretical foundations	3
a. Artificial Intelligence (AI) and Deep Learning	3
b. Convolutional neural network (CNN)	4
c. Neural Architecture Search (NAS)	6
d. Q-NAS	7
3. Post Covid patterns.....	11
4. Previously generated neural networks	14
5. New neural networks.....	18
a. Lung Parenchyma dataset	18
b. Expiration dataset	19
6. Neural Networks Tests.....	20
7. Conclusion and next steps	21
8. Bibliography	22

List of Figures

Figure 1: Illustration of a Deep Neural Network.....	3
Figure 2: Convolutional Layer Diagram	5
Figure 3: Q-NAS Network Quantum Individual Representation Scheme [Julia].....	8
Figure 4: Q-NAS flowchart [11].....	9
Figure 5: CT with fibrosis-like lesions.....	11
Figure 6: CT with nonspecific interstitial pneumonia (NSIP) lesions	11
Figure 7: CT where reabsorption tomographic pattern was identified	12
Figure 8: CT where airway disease was identified	12
Figure 9: Layer functions [4]	14
Figure 10: Confusion Matrix for network #1 [4]	14
Figure 11: Expanded function set [4].....	15
Figure 12: Confusion Matrix for network #2 [4]	15
Figure 13: Confusion Matrix for network #3 [4]	16
Figure 14: Confusion Matrix for network #4 [4]	16
Figure 15: Parameter and Hyperparameter configuration of the Q-NAS model [4]	17
Figure 16: Confusion Matrix for new network #1	19
Figure 17: Confusion Matrix for new network #2	20

List of Tables

Table 1: Results for the new network #1.....	18
Table 2: Architecture for new network #1	18
Table 3: Results for the new network #2.....	19
Table 4: Architecture for new network #2	20

1. Introduction

a. Motivation

Coronavirus disease 2019 (COVID-19) is a highly contagious viral illness caused by severe acute respiratory syndrome SARS-CoV-2. It has had a devastating effect on the world's demographics, resulting in more than 5.3 million deaths worldwide. It has emerged as the most consequential global health crisis since the era of the influenza pandemic of 1918 [1].

After the first cases of this predominantly respiratory viral illness were first reported in Wuhan, Hubei Province, China, in late December 2019, SARS-CoV-2 rapidly disseminated across the world in a short span of time, compelling the World Health Organization (WHO) to declare it as a global pandemic on March 11, 2020. [2]

The outbreak of COVID-19 has proven to be a worldwide unprecedented disaster and can be considered one of the most critical global crises over the last few years. The virus has inflicted billion of lives across physically, psychologically and socially.

Globally, until November 2022, there have been more than 630,000,000 confirmed cases of COVID-19, including over 6,500,000 deaths, reported to WHO [3].

The clinical manifestations of this virus have exhibited deleterious impacts on the respiratory system, that is the primary target organ, but also in the brain, hematological system, liver, kidneys, endocrine system, and others.

The persistence of disabling symptoms long after the acute illness has become known as long COVID 19.

In this study, we aim to use artificial intelligence to analyze new computed tomography chest images taken from patients from Pedro Ernesto University Hospital who had COVID-19 before. By doing this, we hope to aid medical practitioners in identifying the effects that COVID-19 can have on our lungs by improving the detection accuracy of post covid patterns using neural networks.

The model used to develop these networks was the Q-NAS, that is a quantum-inspired technique for finding deep networks by putting together substructures. A NAS is a subarea of AutoML and its model's fundamental tenet is the capacity to automatically generate and seek the finest neural network design. The same basic idea underpins the Q-NAS, but quantum physics paradigms boost accuracy and convergence time.

b. Objectives

The global idea of the project is to identify different types of patterns that can appear on someone's lungs after they have had COVID-19.

This was done by applying the Quantum-Inspired Neural Architecture Search (Q-NAS) model to the CT images dataset. This work is inspired by [4], that already developed neural networks to classify the patterns present on the lungs' images in six different categories: Normal, Fibrosis, Nonspecific Interstitial Pneumonia (NSIP), Reabsorption, Airway, and Other. However, after further analysis of the tomography machine outputs, we realized those six categories could be reorganized. Previously all the images were grouped into one dataset, but as noticed there are four types of scans and two of them can be used to detect those six patterns defined in [4].

So, the project presented in this work has two main objectives.

The first one is to separate the six classifications into two groups, each group using one type of image generated by the tomography machine. These new datasets are used to train new networks and compare them to the previously generated ones.

The second objective is to test all these neural networks using new inputs that were obtained through the Pedro Ernesto University Hospital to simulate a real use of the tool and check its accuracy.

c. Organization

This work has six main chapters as follows:

In Chapter 2 some key concepts necessary to understand how the Q-NAS model work are presented. This includes a brief introduction to deep learning, convoluted networks, neural architecture search and quantum-inspired evolutionary algorithms.

Chapter 3 describes the types of long covid patterns that can be identified by Q-NAS in chest images.

In Chapter 4, the best generated neural networks from [4] are be presented, as well as the respective dataset that was used for training and the final test accuracy. Those networks are used in the tests presented in a further chapter.

In Chapter 5 the neural networks developed in this work are presented with the new classification dataset that was used.

Finally, some new inputs are used to test the previously presented networks. The outcomes of those tests, conclusions and final remarks of this work are presented in the last two chapters.

2. Theoretical foundations

a. Artificial Intelligence (AI) and Deep Learning

"It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."

AI today includes the sub-fields of machine learning and deep learning, which are frequently mentioned in conjunction with artificial intelligence. These disciplines are comprised of AI algorithms that typically make predictions or classifications based on input data. Machine learning has improved the quality of some expert systems and made it easier to create them. [5]

The way in which deep learning and machine learning differ is in how each algorithm learns. "Deep" machine learning can use labeled datasets, also known as supervised learning, to inform its algorithm, but it doesn't necessarily require a labeled dataset. Deep learning can ingest unstructured data in its raw form (e.g. text, images), and it can automatically determine the set of features which distinguish different categories of data from one another. This eliminates some of the human intervention required and enables the use of larger data sets, so we can think of deep learning as "scalable machine learning". Classical, or "non-deep", machine learning is more dependent on human intervention to learn. Human experts determine the set of features to understand the differences between data inputs, usually requiring more structured data to learn. [5]

Deep learning [6] (like some machine learning) uses neural networks. The "deep" in a deep learning algorithm refers to a neural network with more than three layers, including the input and output layers. This is generally represented using the following diagram:

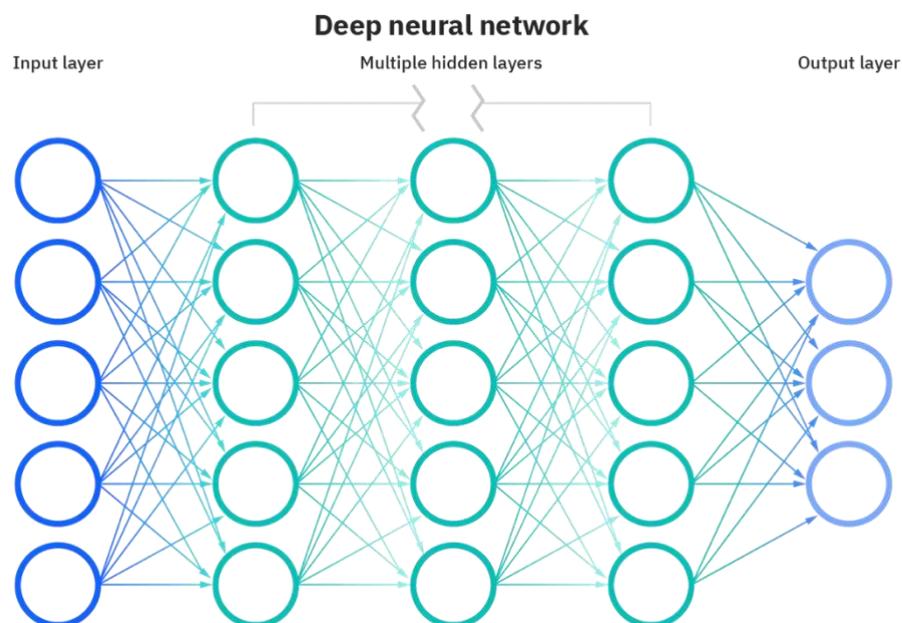


Figure 1: Illustration of a Deep Neural Network

The rise of deep learning has been one of the most significant breakthroughs in AI in recent years, because it has reduced the manual effort involved in building AI systems.

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called *visible* layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. The neural networks this study is interested in are Convolutional neural networks, that are used primarily in computer vision and image classification applications, and can detect features and patterns within an image, enabling tasks, like object detection or recognition.

b. Convolutional neural network (CNN)

Convolutional neural networks outperform other neural networks when given inputs such as images, voice, or audio, for example. Convolutional layer, Pooling layer, and Fully Connected (FC) layer are their three primary types of layers. [7]

A convolutional network's first layer is the convolutional layer. While convolutional layers can be followed by other convolutional layers or pooling layers, the last layer is always the fully connected layer. The CNN's complexity grows with each layer, allowing it to detect larger sections of the image. Early layers emphasize basic elements like colors and borders. The larger features or shapes of the object are first recognized when the visual data moves through the CNN layers, and eventually the intended object is recognized. [7]

Convolutional Layer

The convolutional layer is the core building block of a CNN, and it is where most of the computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

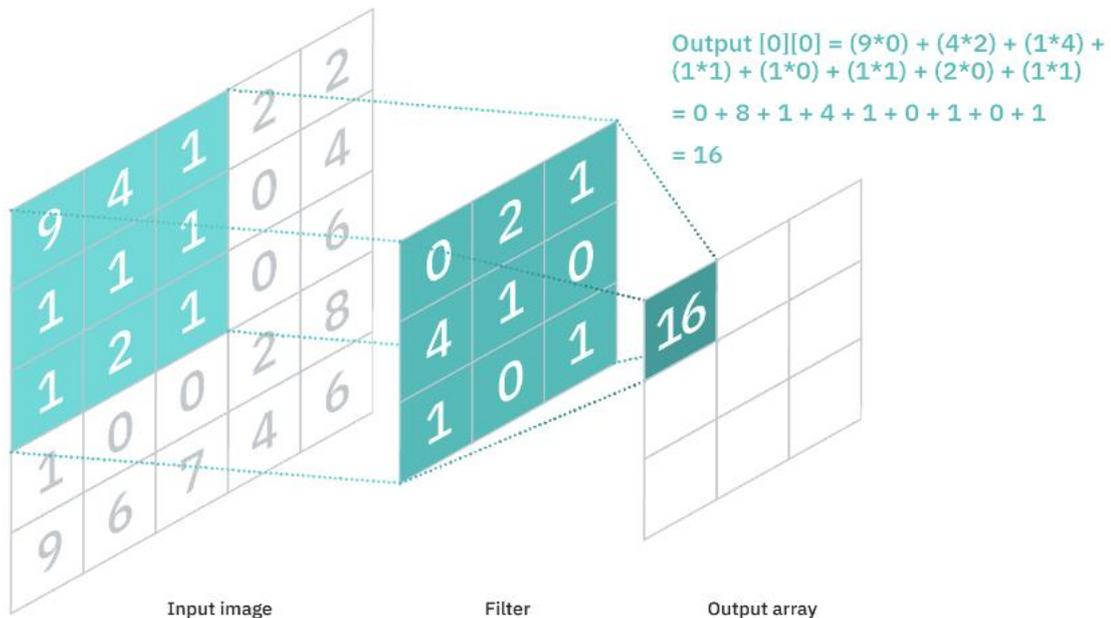


Figure 2: Convolutional Layer Diagram

As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as “partially connected” layers. However, this characteristic can also be described as local connectivity.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The **number of filters** affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.
2. **Stride** is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3. **Zero-padding** is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:
 - **Valid padding:** This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
 - **Same padding:** This padding ensures that the output layer has the same size as the input layer
 - **Full padding:** This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within

the receptive fields of prior layers. As an example, let's assume that we're trying to determine if an image contains a bicycle. You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower-level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN.

Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

Pooling Layer

Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array [8]. There are two main types of pooling:

- **Max pooling:** As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- **Average pooling:** As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

Fully Connected Layer

The name of the fully connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use REL functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

c. Neural Architecture Search (NAS)

Neural Architecture Search (NAS) is a subarea of AutoML and is an essential step toward automating machine learning methods. It is a technique that aims to automate the construction processes of a neural network architecture [9].

The idea behind using this algorithm is optimizing the process of deciding all the variables necessary when creating a CNN. It can automatize de the decisions regarding the architectures which implies deciding the quantity of layers, the configuration of filter to be used. Or even, decisions regarding the values of the hyperparameters such as the learning rate. As these choices directly influence the performance of the network, if it's fast or has good accuracy, the NAS facilitates the process of optimizing the neural network by automatizing the process.

d. Q-NAS

The Quantum Inspired Neural Architecture Search model is one approach that is based on quantum-inspired evolutionary algorithms. The idea of quantum-inspired computing is to create classical algorithms, which can be executed in classical computers, but that take advantage of the paradigms of quantum physics [10, 11]

Evolutionary Algorithms

Evolutionary algorithms are inspired in the evolution process by using a population of individuals within some environment that has limited resources. These limitations generate competition for those resources which propitiates the natural selection phenomenon. Given a quality function to be maximized, we can randomly create a set of candidate solutions and apply this function to discover the fitness of the population – the higher the better. Based on these fitness values some of the better candidates are chosen to create the next generation. This is done by applying recombination and mutation to them.

Recombination is an operator that is applied to two or more selected candidates (parents), producing one or more new candidates (children). Mutation, on the other hand is applied to one candidate and results in one new candidate. Therefore, executing the operations of recombination and mutation on the parents leads to the creation of a set of new candidates. These have their fitness evaluated and then compete – based on their fitness with the old ones for a place in the next generation. This process can be iterated until a candidate with sufficient quality is found or a computational limit is reached [12].

The general scheme of an evolutionary algorithm in pseudocode is as follows.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Quantum Inspired Evolutionary Algorithms (QIEA)

The QIEAs were created by employing important quantum computing concepts, including the quantum bit, the linear superposition of states, and the quantum rotation gate in the original evolutionary algorithms.

According to empirical findings, QIEAs can solve many optimization problems more effectively and efficiently than comparable algorithms with fewer evaluations. This quality can be extremely important in applications like NAS, where it can be quite expensive to evaluate potential solutions [13].

Analogue to a classical computer, the smallest unit of information stored in a two-state quantum computer is the Q-bit. The main difference is that besides the regular 1 or 0 states, the Q-bit state can be found in a superposition of both states, and it can be described by the following equation:

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

where $|\Psi\rangle$ is the q-bit state, α and β are complex numbers and $|\alpha|^2$ and $|\beta|^2$ gives the probability that the Q-bit will be found in the 0 or 1 states respectively [14].

Initially, a Q-bit individual represents all possible states with the same probability. During evolution, a Q-gate operator can modify the probability of each Q-bit, so it gradually converges to a single state: the

optimal solution. As each quantum individual represents the probability of a state, they cannot be directly evaluated: they must be observed to generate classical individuals represented by one single state [13].

The general scheme of an QIEA in pseudocode is as follows.

```

BEGIN
  t ← 0 ;
  INITIALISE quantum population Q(t);
  GENERATE classical population P(t);
  EVALUATE P(t);
  STORES best solution of P(t) in B(t);
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 t ← t+1;
    2 P(t) ← generated classical population from Q(t);
    3 EVALUATE P(t);
    4 UPDATE Q(t) using Q-gate;
    5 STORE best individuals of B(t-1) ∪ P(t) in P(t);
    6 STORE the best solution b in B(t);
  OD
END
    
```

Quantum Inspired Neural Architecture Search (Q-NAS)

The Q-NAS algorithm was recently developed to address the efficiency issue and was inspired by the performance of QIEAs compared to other evolutionary approaches.

In the Q-NAS model, quantum individuals' chromosomes have two parts: one is responsible for encoding the hyperparameters and the other for encoding the structure design of the network. However, this work will not focus on hyperparameters evolution because its evolution does not provide better results when compared with the Tensorflow's default hyperparameters values [11].

As mentioned before, one quantum individual can generate multiples classical individuals based on observations, that are then evaluated as solutions containing the network architecture. The figure below shows the network generation process for one individual in a specific generation from the user input parameters to the final decoded network.

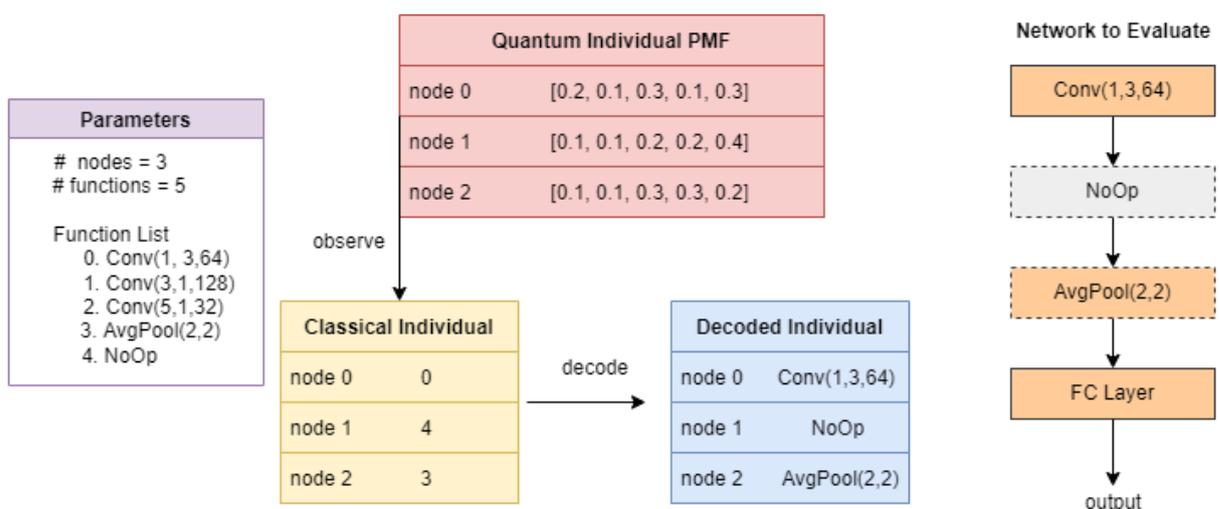


Figure 3: Q-NAS Network Quantum Individual Representation Scheme [Julia]

Each generation development process has three main phases and can be illustrated by the diagram below. This process is repeated over and over until evolution is completed.

1. Population generation: the classical population is generated based on the observations of the quantum individuals. Based on this observation, the classical individuals are decoded, and the corresponding network is ready to be evaluated.
2. Candidate evaluation: The evaluation procedure involves training the candidate networks for a few epochs with a subset of the training data and using a validation dataset to assign a fitness score to the individual.
3. Ranking and update: after the evaluation, quantum individuals are updated based on the best classical individuals. This update increases the probability of the most promising function, based on its fitness, and consequently reduces the other probabilities proportionally so that it continues to add up to 1.

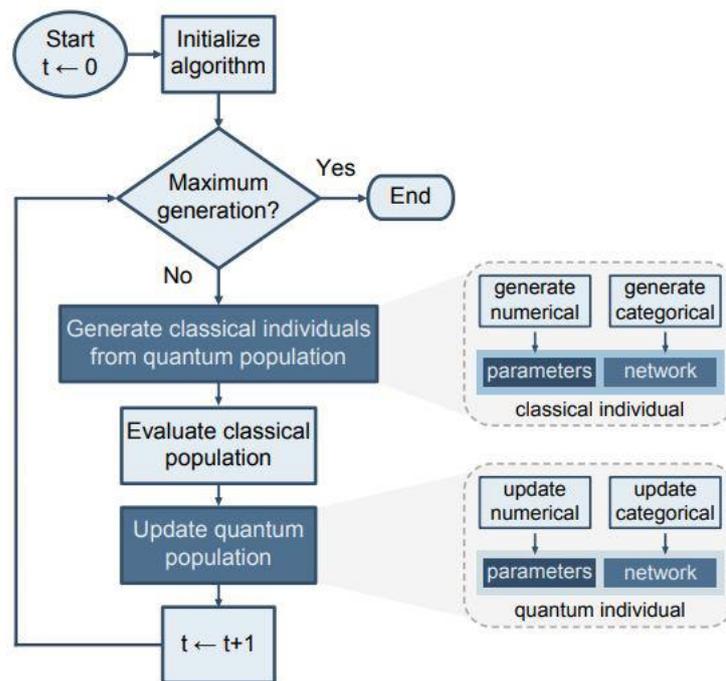


Figure 4: Q-NAS flowchart [11]

Between generation some other processes take place. In the first generation, $C(t)$ individuals are ranked and stored in $P(t)$. Note that classical recombination is only possible after the first generation. From the second generation, since $P(t)$ exists, a new population $C(t)$ must be evaluated and then the best individuals from old ($P(t)$) and new populations ($C(t)$) are selected to be saved. Finally, quantum individuals are updated based on the best classical individuals. The idea is to gradually modify the quantum population so it can generate solutions that are closer to the optimal.

The combination of these processes is repeated for T generations and the general scheme of a Q-NAS pseudocode is as follows, where $Q(t)$ is the quantum population, $C(t)$ is the classical population and $P(t)$ is the saved classical population.

```
BEGIN
  t ← 0 ;
  INITIALISE quantum population Q(t);
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 GENERATE classical population C(t) observing Q(t);
    2 IF t=0 THEN
      EVALUATE C(t);
      P(t) ← C(t);
    3 ELSE
      C(t) ← recombination between C(t) and P(t);
      EVALUATE C(t);
      P(t) ← best individuals from C(t) ∪ P(t);
    4 END IF
    5 Q(t+1) ← UPDATE Q(t) based on P(t) values;
    6 t ← t+1;
  OD
END
```

When the evolution is complete, the final architecture is retrained from scratch for 300 epochs with all the available training data. In addition, the network is evaluated every ten epochs using a validation dataset. The periodic evaluations' accuracy is used to save the best model during the retraining phase. When training is over, the best validation model is applied to the test data to obtain the final accuracy value. The test accuracy is used to compare the models among different experiments and with other works [11].

3. Post Covid patterns

In this section, we present the use of Q-NAS algorithm applied to Post Covid Pattern detection in computed tomography (CT) images. The images used in this work were extracted from patients by the Pedro Ernesto University Hospital and were evaluated and labeled by radiologists after analyzing the numerous images generated for each chest CT [15].

The following tomography images showcase examples of the Post Covid patterns that the network must correctly identify.

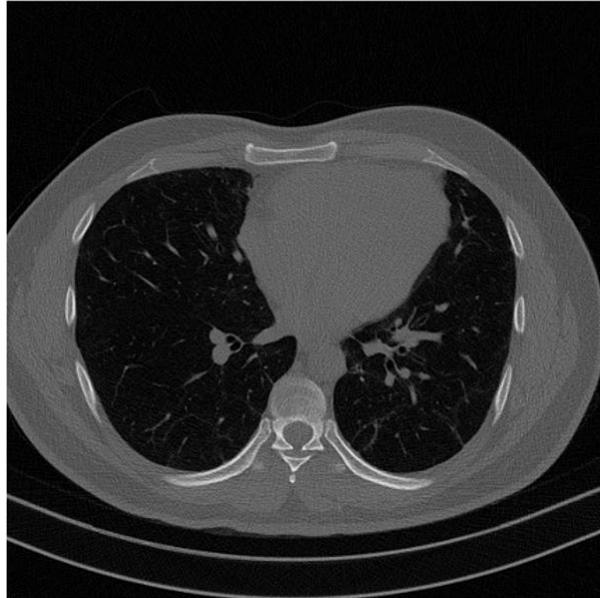


Figure 5: CT with fibrosis-like lesions

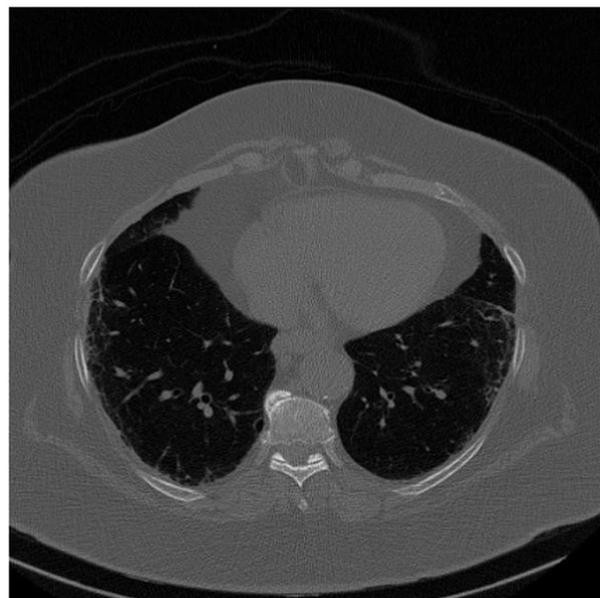


Figure 6: CT with nonspecific interstitial pneumonia (NSIP) lesions



Figure 7: CT where reabsorption tomographic pattern was identified

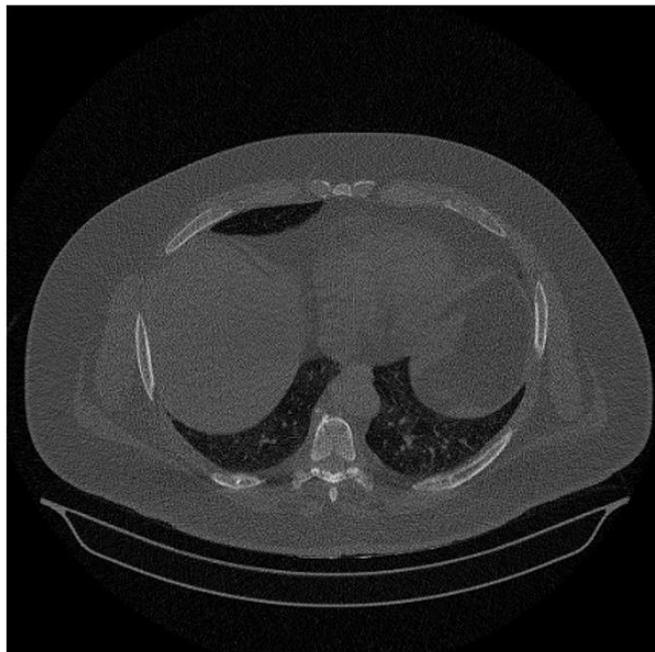


Figure 8: CT where airway disease was identified

As it was mentioned previously, each chest CT generates several images. These images are not necessarily of the lung, as the machine scans the chest, they may produce some images of bones or other organs for example, and these are called "Other". The network classification of these images is worth discussing, as well as the classification of lung images that don't present any of the patterns mentioned above and can be called "Normal".

Currently, there are three possibilities for the classification: (1) classify them as two different groups (Normal and Other); (2) classify them as a single group; (3) remove Other images and leave only the ones where no pattern can be detected in the lung, Normal.

The three possibilities were tested in [4] [15] and dataset without the other class was the most accurate one as the best network reached an accuracy of 98.636%. These generated networks will be discussed in the next session.

4. Previously generated neural networks

The Q-NAS algorithm used to generate the neural networks responsible for detecting the Post Covid Pattern had at its disposal the following function layers. Each classical individual represents a certain combination of these layers to be evaluated. In the quantum spectrum, each individual is formed by chromosomes that instead of representing one single state (layer), they are in a superposition of all states, each with its own probability.

function name	function	kernel size	stride	filters	initial probability
conv_1_1_32	ConvBlock	1	1	32	0.042
conv_1_1_64	ConvBlock	1	1	64	0.042
conv_3_1_32	ConvBlock	3	1	32	0.042
conv_3_1_64	ConvBlock	3	1	64	0.042
conv_3_1_128	ConvBlock	3	1	128	0.042
conv_3_1_256	ConvBlock	3	1	256	0.042
conv_5_1_32	ConvBlock	5	1	32	0.042
conv_5_1_64	ConvBlock	5	1	64	0.042
max_pool_2_2	MaxPool	2	2	-	0.167
avg_pool_2_2	AvgPool	2	2	-	0.167
no_op	NoOp	-	-	-	0.333

Figure 9: Layer functions [4]

The first network that was generated in [4] used six possible classifications: Normal, Fibrosis, NSIP, Reabsorption, Airway and Other. This configuration had some great results and produced a network with a validation accuracy of 0.925% and most importantly, a test accuracy of 89.394%.

		Predicted Class					
		Normal	Fibrosis	NSIP	Reabsorption	Airway	Other
Actual Class	Normal	41	1	0	1	0	1
	Fibrosis	0	44	0	0	0	0
	NSIP	0	0	42	1	0	1
	Reabsorption	0	2	0	38	0	4
	Airway	0	0	0	0	43	1
	Other	2	2	2	7	3	28

Figure 10: Confusion Matrix for network #1 [4]

However, we can see that there are still problems with the Other classification, as mentioned in the previous section.

For the second network, the experiment the function set was expanded to the one specified below. In this new function set there were more convolutional functions which included different filter options.

function name	function	kernel size	stride	filters	initial probability
conv_1_1_64	ConvBlock	1	1	64	0.028
conv_1_1_128	ConvBlock	1	1	128	0.028
conv_1_1_256	ConvBlock	1	1	256	0.028
conv_1_1_512	ConvBlock	1	1	512	0.028
conv_3_1_64	ConvBlock	3	1	64	0.028
conv_3_1_128	ConvBlock	3	1	128	0.028
conv_3_1_256	ConvBlock	3	1	256	0.028
conv_3_1_512	ConvBlock	3	1	512	0.028
conv_5_1_64	ConvBlock	5	1	64	0.028
conv_5_1_128	ConvBlock	5	1	128	0.028
conv_5_1_256	ConvBlock	5	1	256	0.028
conv_5_1_512	ConvBlock	5	1	512	0.028
max_pool_2_2	MaxPool	2	2	-	0.167
avg_pool_2_2	AvgPool	2	2	-	0.167
no_op	NoOp	-	-	-	0.333

Figure 11: Expanded function set [4]

The second network was generated using same possible classifications and this time, the algorithm created a network with a test accuracy of 90.909% which is better than the previous one. It also has less layers and a validation accuracy of 92.500%.

Predicted Class

Actual Class	Normal	42	0	0	1	0	1
	Fibrosis	0	42	0	1	0	1
	NSIP	0	0	44	0	0	0
	Reabsorption	0	2	0	38	0	4
	Airway	0	0	0	0	44	0
	Other	4	2	1	5	2	30
		Normal	Fibrosis	NSIP	Reabsorption	Airway	Other

Figure 12: Confusion Matrix for network #2 [4]

The same problem observed in the first network with the Other classification can be found in this one.

To see if the accuracy would increase without the class Other, two new datasets with only 5 classes were created. In the first one, the classes Fibrosis, NSIP, Reabsorption and Airway were kept the same, the difference is that in the first dataset we combined the classes Normal with Other.

This new dataset generated a network with a 94.091% test accuracy and a 93.500% validation accuracy, which is better than the previous two networks.

		Predicted Class				
		Normal + Other	Fibrosis	NSIP	Reabsorption	Airway
Actual Class	Normal+Other	34	0	0	8	2
	Fibrosis	0	44	0	0	0
	NSIP	1	0	43	0	0
	Reabsorption	0	1	1	42	0
	Airway	0	0	0	0	44

Figure 13: Confusion Matrix for network #3 [4]

The second new dataset was created by excluding the class Other, while all the other classes remained the same.

As expected, the results found with this dataset are much better than the ones found in all the other experiments. As it can be observed in the confusion matrix below, this is because we don't have the class Other, which was the reason for most of the errors in the previous tests. The generated network has a test accuracy of 98.636% and a validation accuracy of 99.500%.

		Predicted Class				
		Normal	Fibrosis	NSIP	Reabsorption	Airway
Actual Class	Normal	44	0	0	1	0
	Fibrosis	0	42	0	2	0
	NSIP	0	0	44	0	0
	Reabsorption	0	1	0	43	0
	Airway	0	0	0	0	44

Figure 14: Confusion Matrix for network #4 [4]

The hyperparameters and evolution parameters configuration shared among all the experiments are the following.

Parameter	Value
crossover_rate	0.5
num_quantum_ind	2
repetition	2
update_quantum_gen	5
update_quantum_rate	0.1
penalize_number	8
max_num_nodes	20
decay	0.9
learning_rate	1.0e-3
momentum	0.9
weight decay	1.0e-4

Figure 15: Parameter and Hyperparameter configuration of the Q-NAS model [4]

More details on the network's generation, parameters and architecture can be found in [4].

5. New neural networks

The images used on the previous datasets corresponded to the complete output of a CT scan of a patient. However, after analyzing more carefully the images, we realized they are not all the same medical exam. The machine itself divides the output into four groups, and each group is used to monitor certain patterns in the patients' lungs. The two we are interested in are the ones automatically classified as lung parenchyma and expiration. The first one is used by the doctors to detect the airways disease, and the second one helps the detection of fibrosis, NSIP and reabsorption patterns. After this discovery it became clear that two models should be generated instead of one, as there is no need to force the machine to detect all the patterns among all the images if they are already generated in a way that makes this process easier. Therefore, two new neural networks were created and trained using this new classification.

a. Lung Parenchyma dataset

The first one used the lung parenchyma images and was generated using 5 possible classifications: Normal, Fibrosis, NSIP, Reabsorption, and Other. The parameters were kept the same as the previously mentioned networks for comparison purposes. The results for the new dataset are listed in table below.

Table 1: Results for the new network #1

#	# of layers	# parameters	accuracy		
			fitness	retrain validation	retrain test
1	15	2.93M	0.97000	0.96500	0.95455
2	13	10.05M	0.96250	0.9500	0.92636
3	13	1.25M	0.97000	0.96500	0.95000

As we are using the Normal and Other classes, this network can be compared to network #2 previously presented. This new network has a better test and validation accuracies: 95,455% and 96,500% compared to 90.909% and 92.500%, respectively. This result was already expected because we're excluding one of the patterns (Airway) as it cannot be detected in these images. So, there are no useless inputs, and the classification process becomes easier for the machine. The architecture and confusion matrix for new network #1 can be found below.

Table 2: Architecture for new network #1

Node	Function Name
1	avg_pool_2_2
2	max_pool_2_2
3	conv_5_1_256
4	conv_5_1_128
5	max_pool_2_2
6	conv_1_1_128
7	max_pool_2_2
8	conv_1_1_64
9	avg_pool_2_2
10	conv_5_1_128
11	conv_3_1_64

12	conv_5_1_128
13	conv_5_1_256
14	conv_1_1_512
15	conv_3_1_128

Predicted Class

Actual Class	Normal	44	0	0	2	0
	Fibrosis	0	42	0	2	0
	NSIP	0	0	44	0	0
	Reabsorption	4	1	0	36	3
	Other	0	0	0	0	44
		Normal	Fibrosis	NSIP	Reabsorption	Other

Figure 16: Confusion Matrix for new network #1

b. Expiration dataset

The second one used the expiration images and was generated using 2 possible classifications: Normal + Other and Airway. The results for the new dataset are listed in table below.

Table 3: Results for the new network #2

#	[8]# of layers	# parameters	fitness	accuracy	
				retrain validation	retrain test
1	14	2.93M	0.95000	0.96250	0.94318
2	14	10.05M	0.96250	0.96250	0.92045
3	14	1.25M	0.96250	0.95000	0.93182

As we are using the Normal + Other class, this network can be compared to network #3 previously presented. This new network has a better test and validation accuracies: 94,318% and 96,250% compared to 94.091% and 93.500%, respectively. This result was already expected because we're excluding most patterns as they cannot be detected in these images. So, there are no useless inputs, and the classification process becomes easier for the machine. The architecture and confusion matrix for new network #2 can be found below.

Table 4: Architecture for new network #2

Node	Function Name
1	avg_pool_2_2
2	conv_1_1_64
3	conv_1_1_128
4	conv_1_1_256
5	conv_1_1_512
6	conv_3_1_64
7	conv_3_1_128
8	conv_3_1_256
9	conv_3_1_512
10	conv_5_1_64
11	conv_5_1_128
12	conv_5_1_256
13	conv_5_1_512
14	max_pool_2_2

		Predicted Class	
Actual Class	Normal+Other	42	2
	Airway	3	41
		Normal+Other	Airway

Figure 17: Confusion Matrix for new network #2

6. Neural Networks Tests

In order to make some tests and simulate the real use of the networks, the files inside the testing directory were replaced by other images. With these new images, the retrain process was run again to test the accuracy of the network for new images.

However, the results obtained were not satisfactory. None of the networks were able to correctly classify the new images of the patients from Pedro Ernesto University Hospital and we were not able to determine if the problem is with the training or the new inputs.

7. Conclusion and next steps

In this work we were able to develop two new networks that have better accuracies than the one previously generated in [4]. This was achieved by using two new datasets that were generated based on the output of the CT machine. It's now more adapted to the real use of the tool as in reality the images of used to detect the Airway pattern would not be mixed with the other ones as they correspond to two different exams. The one with only two classes has an accuracy of 0.94318, and the one with five of 0.95455. These values show that this approach is a better one to be followed that the one used before to train the previous networks. For future works these new division (lung parenchyma and expiration images) can be tested with new datasets that presented promising results in [4], as for example using the strategy of removing the Other class.

However, we were not able to prove the accuracy of neither the neural networks developed in [4], nor the ones developed in this work. When new images were used in the train process there were no satisfactory results. This need to be further investigated to determine if the problem is in the new inputs or the ones used for training the networks.

8. Bibliography

- [1] PhysioPedia, "Coronavirus Disease (COVID-19)," [Online]. Available: [https://www.physio-pedia.com/Coronavirus_Disease_\(COVID-19\)](https://www.physio-pedia.com/Coronavirus_Disease_(COVID-19)).
- [2] C. Sohrabi, Z. Alsafi, N. O'Neill, M. Khan, A. Kerwan, A. Al-Jabir, C. Iosifidis and R. Agha, "World health organization declares global emergency: A review of the 2019 novel coronavirus (covid-19)," *International Journal of Surgery*, vol. 76, 2020.
- [3] W. H. O. (WHO). [Online]. Available: <https://covid19.who.int/>. [Accessed 1 November 2022].
- [4] G. Espozel, "Q-NAS Applied to the Classification of Medical Images," 2022.
- [5] IBM, "Artificial Intelligence (AI)," [Online]. Available: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.
- [6] IBM, "Deep Learning," [Online]. Available: <https://www.ibm.com/cloud/learn/deep-learning>.
- [7] IBM, "Convolutional Neural Networks," [Online]. Available: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>.
- [8] F. Chollet, "Deep Learning with Python," 2018.
- [9] F. Hutter, L. Kotthoff and J. Vanschoren, *Automated machine learning: methods, systems, challenges*, 2019.
- [10] M. Cardoso, M. Silva, M. Vellasco and E. Cataldo, "Quantum-inspired features and parameter optimization of spiking neural networks for a case study from Atmospheric," *Procedia Computer Science*, vol. 53, 2015.
- [11] D. Szwarcman, D. Civitarese and M. Vellasco, "Quantum-inspired neural architecture search," *IEEE*, 2019.
- [12] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2015.
- [13] J. D. Noce, "Enhanced q-nas for image classification," 2022.
- [14] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE*, 2002.
- [15] C. G. C. Chantong, "Caracterização dos padrões pulmonares da covid-19 pós-aguda na tomografia computadorizada e testagem de um modelo correspondente de inteligência artificial," 2022.