END OF YEAR PROJECT

SPECIALTY: NETWORKS AND TELECOMMUNICATIONS

CARTHAGE UNIVERSITY

DEEP LEARNING PARALLELIZATION BY NEURAL NETWORKS FOR BIOLOGICAL AND MEDICAL ANALYSIS

PRESENTED BY:

SARAH BEN ABDALLAH

MAHDI BEN ZINOUBA

MOHAMED YASSINE ELKHADIRI

SUPERVISOR:

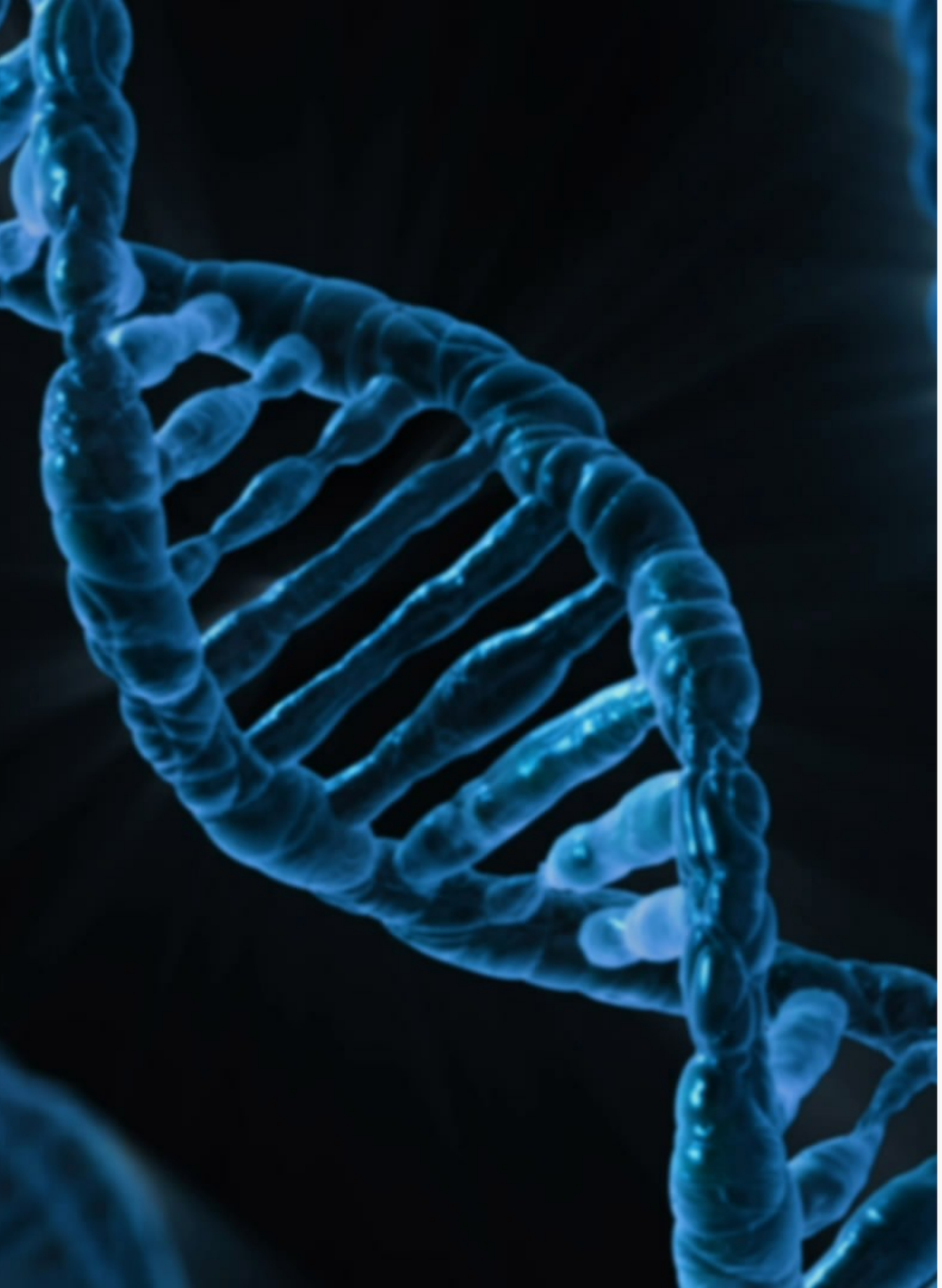
SOFIANE OUNI

Presented On: 06/06/2022

TABLE OF CONTENT

1

PROBLEM STATEMENT



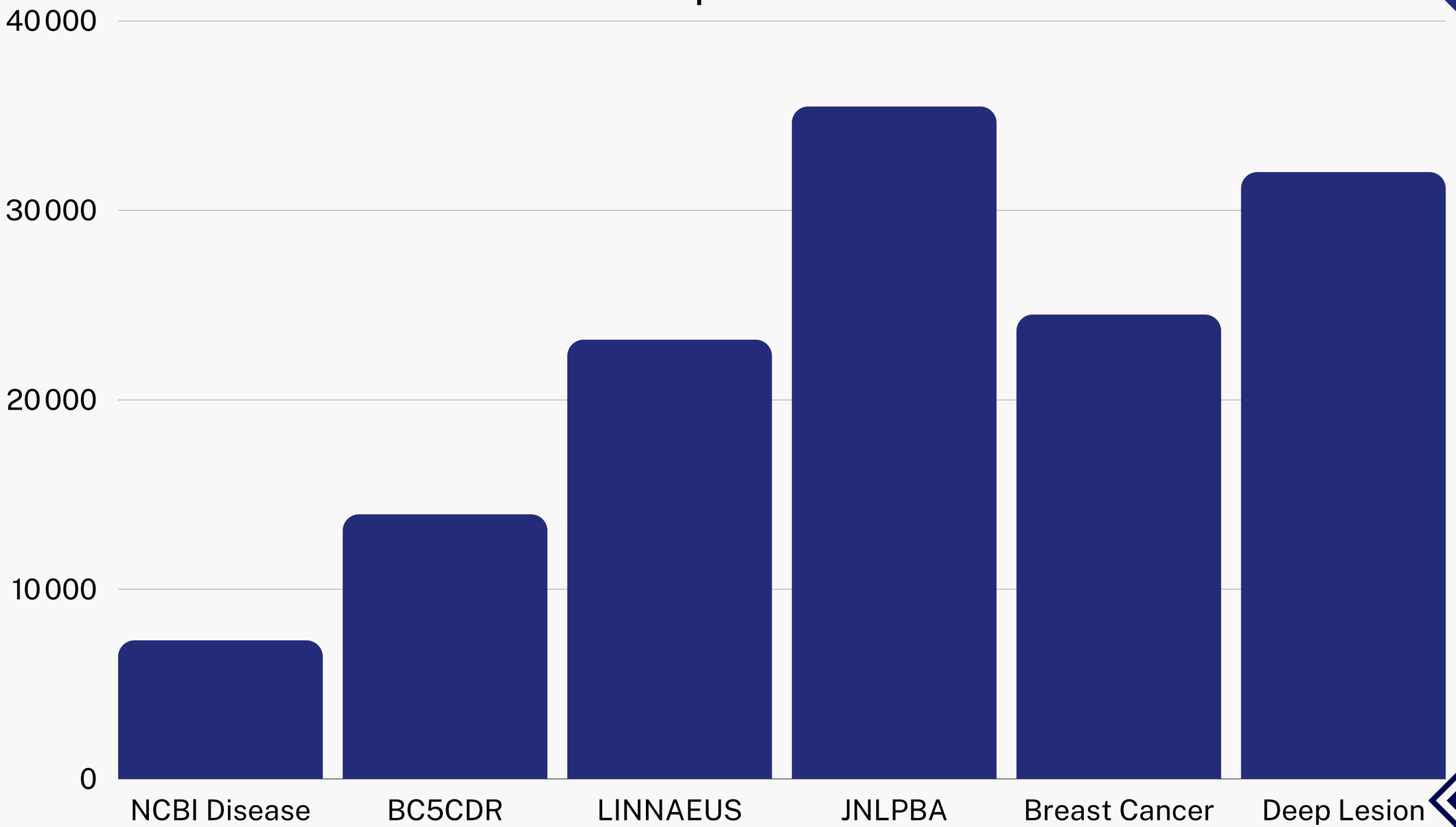
PROBLEM STATEMENT

Recently, the bioinformatics research community has developed and specialized laboratories have been established around the world.

The related data enabled researcher to study biological systems at a higher specificity.

Biological and medical databases have considerably grown since many years. A base request that only took few minutes can now take days.

Number of instances per Biomedical Datasets



2

SOLUTION

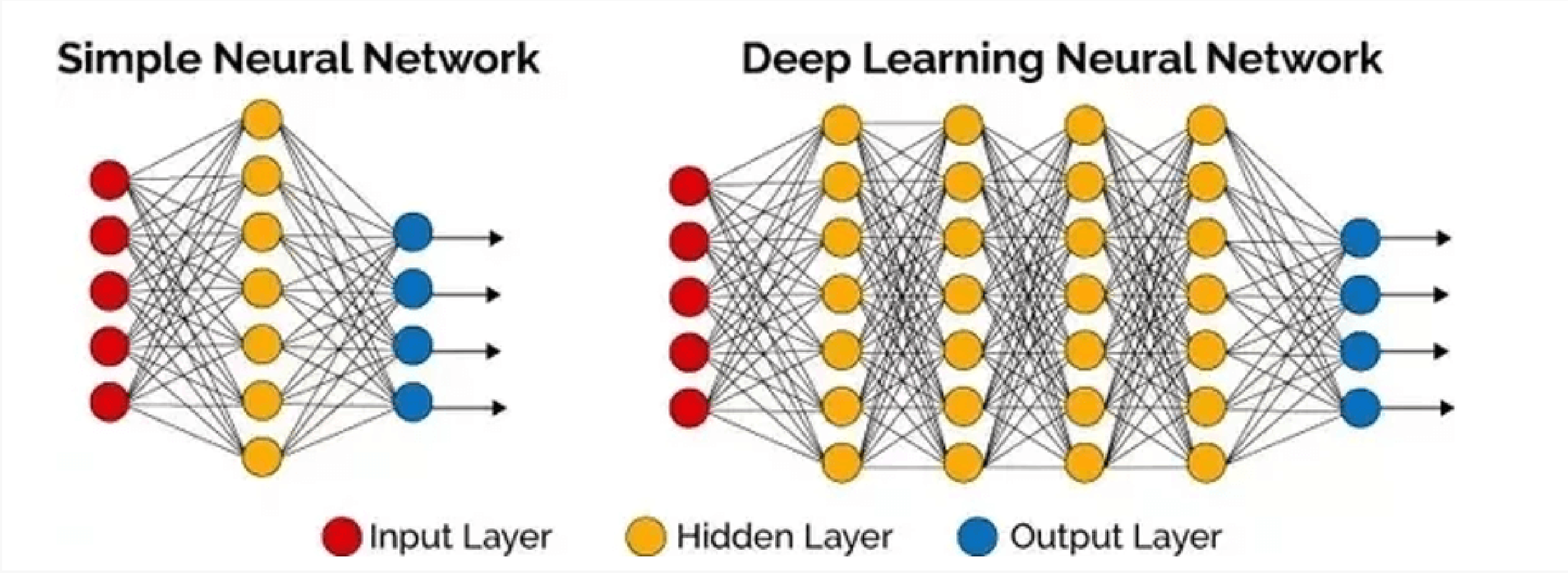
NEURAL NETWORKS PARALLELIZATION

The proposed system provides a deep learning algorithm in a distributed system to effectively predict the prevalence of a disease in the least possible training time.

3

DEEP LEARNING

DEEP LEARNING



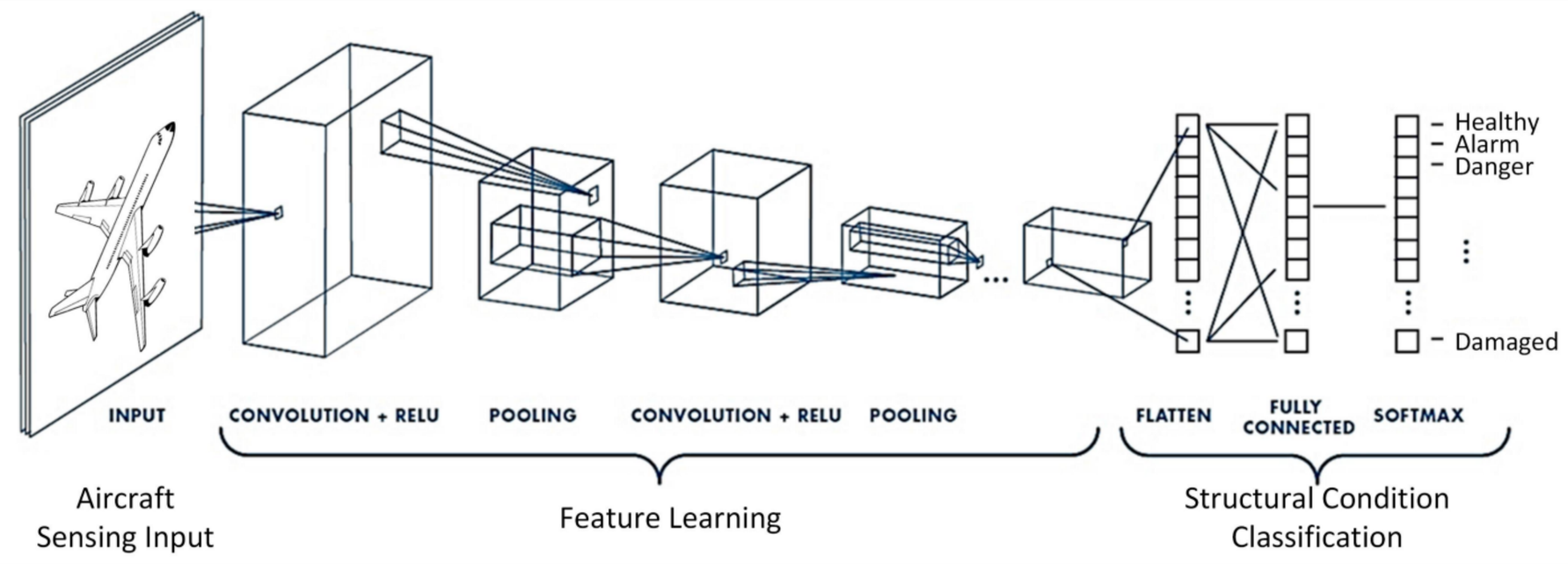
CONVOLUTIONAL NEURAL NETWORKS

The Convolutional Neural Network (CNN) is a type of artificial neural network that includes convolution functions perceived as feature extractors and organized in layers.

CNN is adapted to image and matrix processing and used to solve several problems of classification.

There are several datasets that have been proposed in the literature and trained on different models such as MNIST and CIFAR datasets.

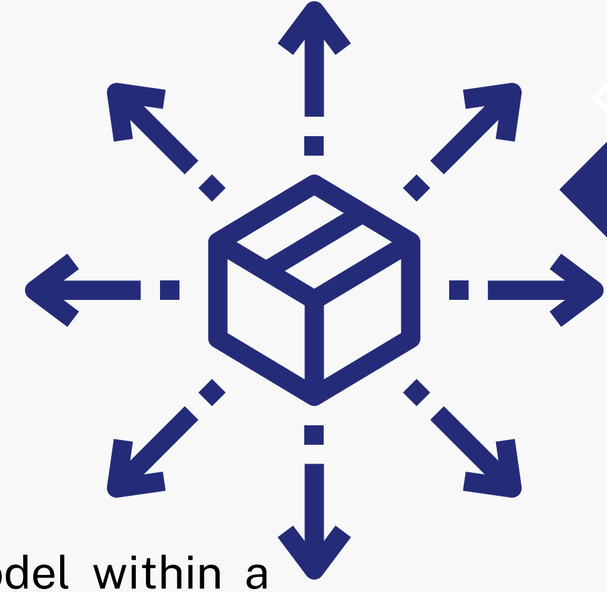
The transfer learning technique allows to transfer the knowledge of these pre-trained models for other databases and other problems.



4

DISTRIBUTED TRAINING

DISTRIBUTED TRAINING

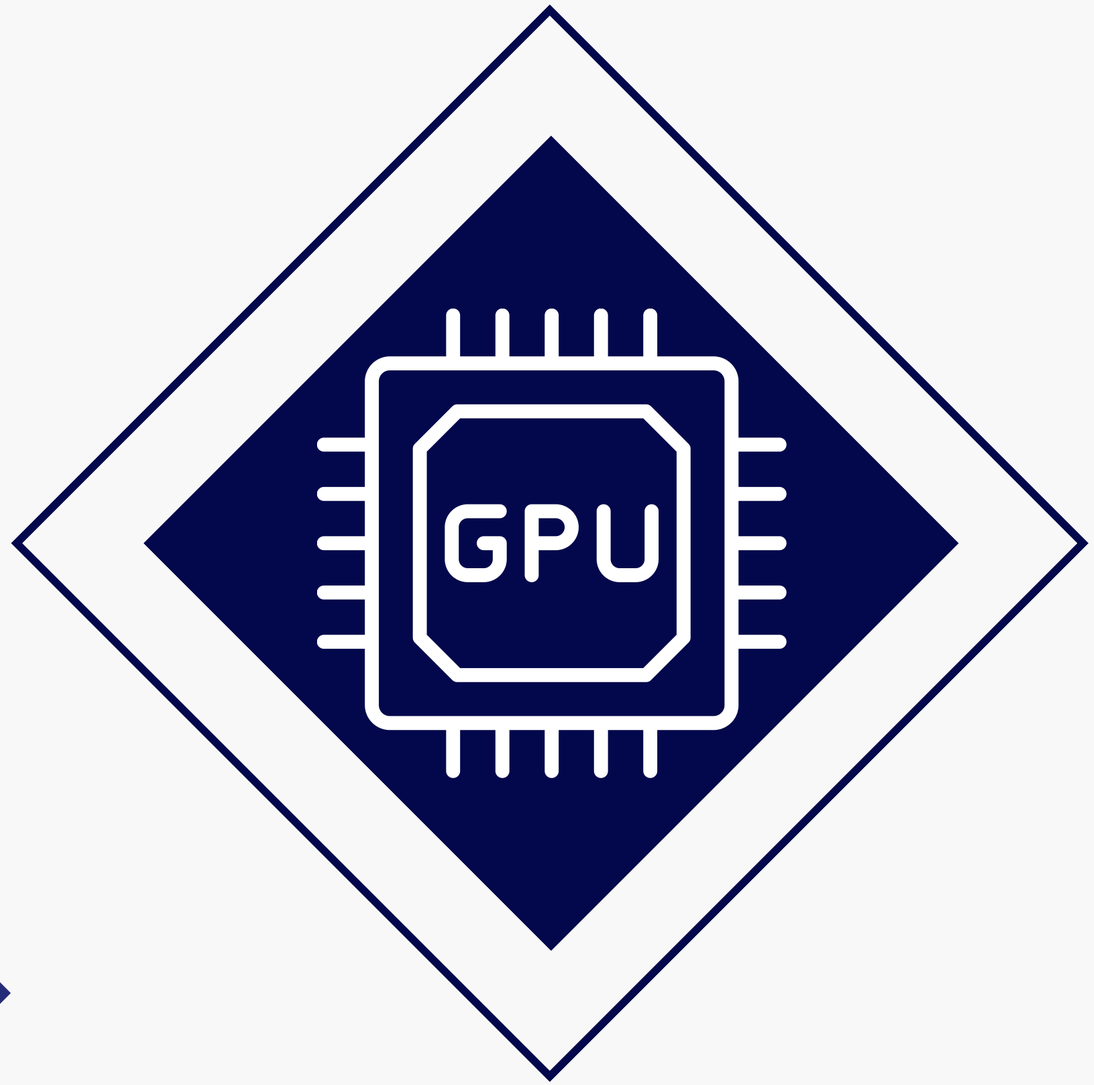


As the dataset size increases, it becomes very difficult to train a model within a limited time frame. To address this type of problem, distributed training methods are used.

Distributed training allows us to train very large models and reduce training time. Parallelization of DNNs is considered a challenge due to the inherent sequential nature of stochastic gradients (SGD).

Distributed training consists on adding multiple GPUs to have multiple machines , all connected over a network.

GPU



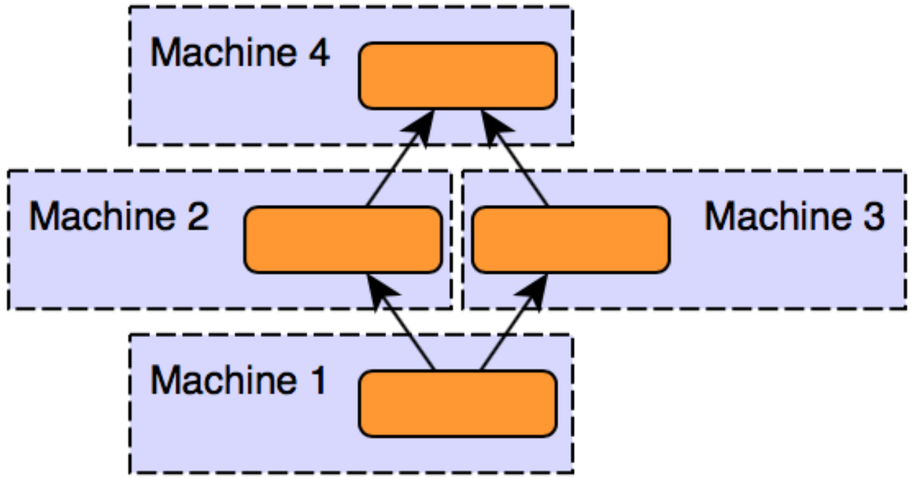
GPU (graphics processing unit)
It is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.

But why are we using GPU over CPU ?

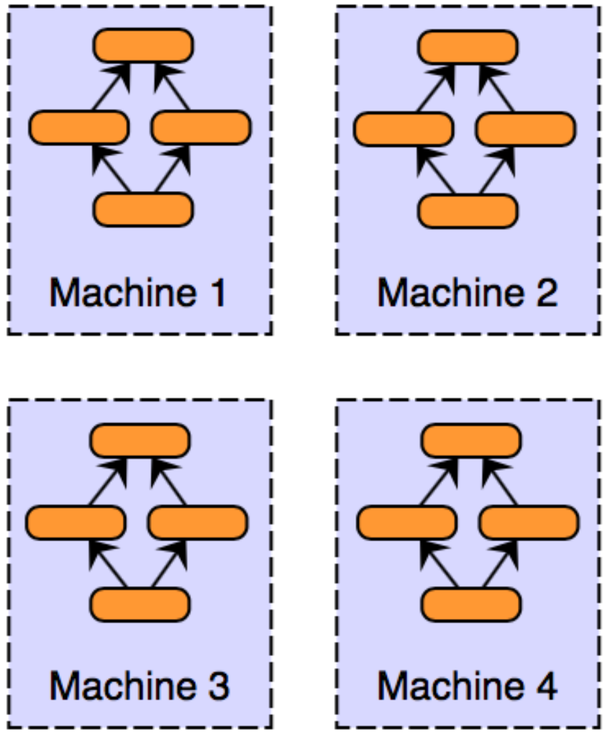
A GPU is a processor that is great at handling specialized computations. We can contrast this to the Central Processing Unit(CPU), which is great at handling general computations.

TYPES OF DISTRIBUTED TRAINING

Model Parallelism



Data Parallelism



DISTRIBUTED TRAINING STRATEGIES

Synchronous Training

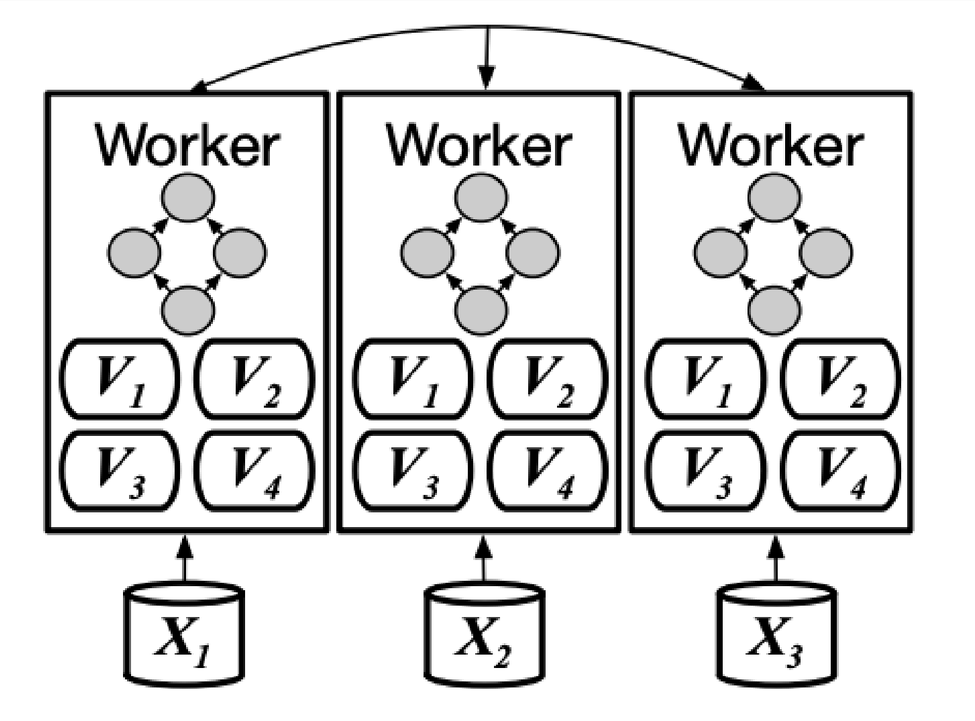
In sync training, all workers/accelerators train over different slices of input data and aggregate the gradients in each step.

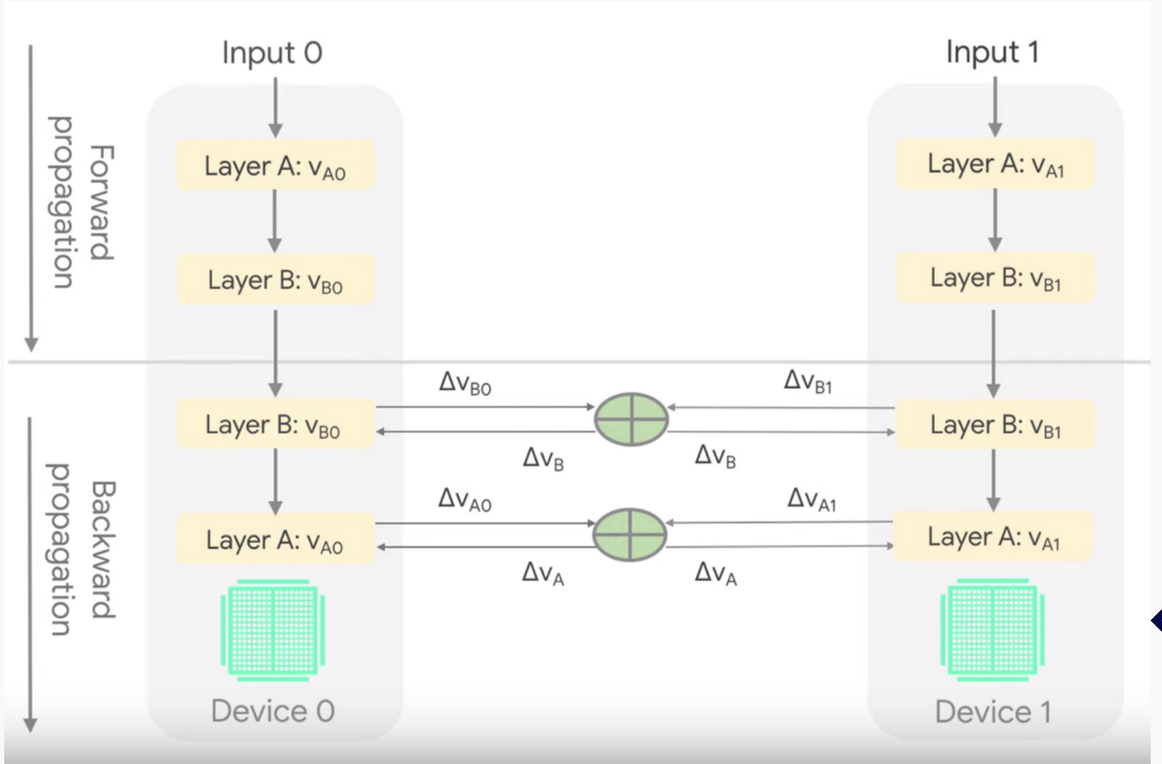
Asynchronous training

In async training, all workers/accelerators are independently trained over the input data and update variables in an asynchronous manner.

SYNCHRONOUS TRAINING

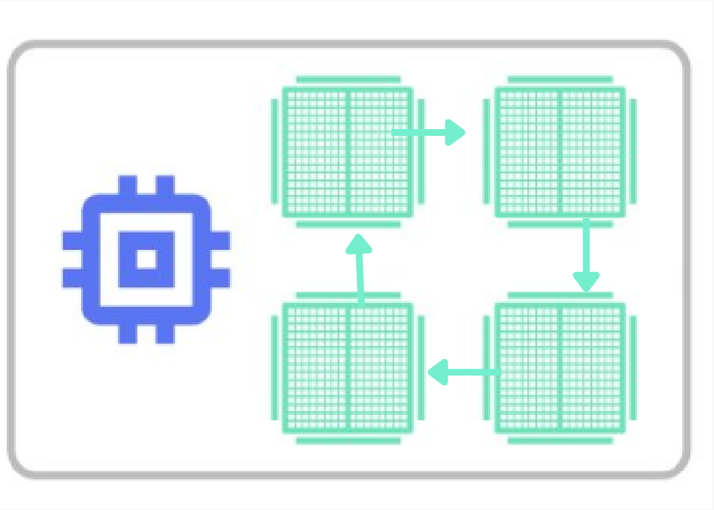
All-reduce Algorithm :
It is a distributed algorithm that aggregates all the trainable parameters from different workers or accelerators.
It receives the weights from all workers and performs aggregation
on them to compute the final weights.



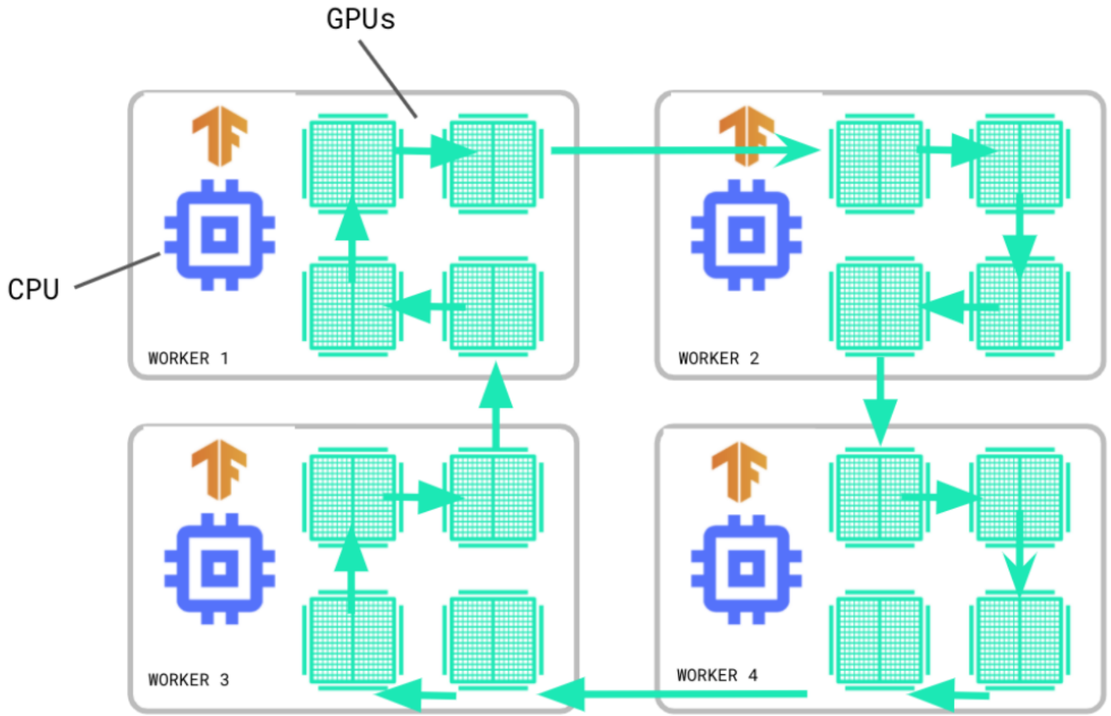


SYNCHRONOUS TRAINING

Mirrored Strategy



Multiworker Mirrored Strategy



ASYNCHRONOUS TRAINING

Parameter Server Algorithm :

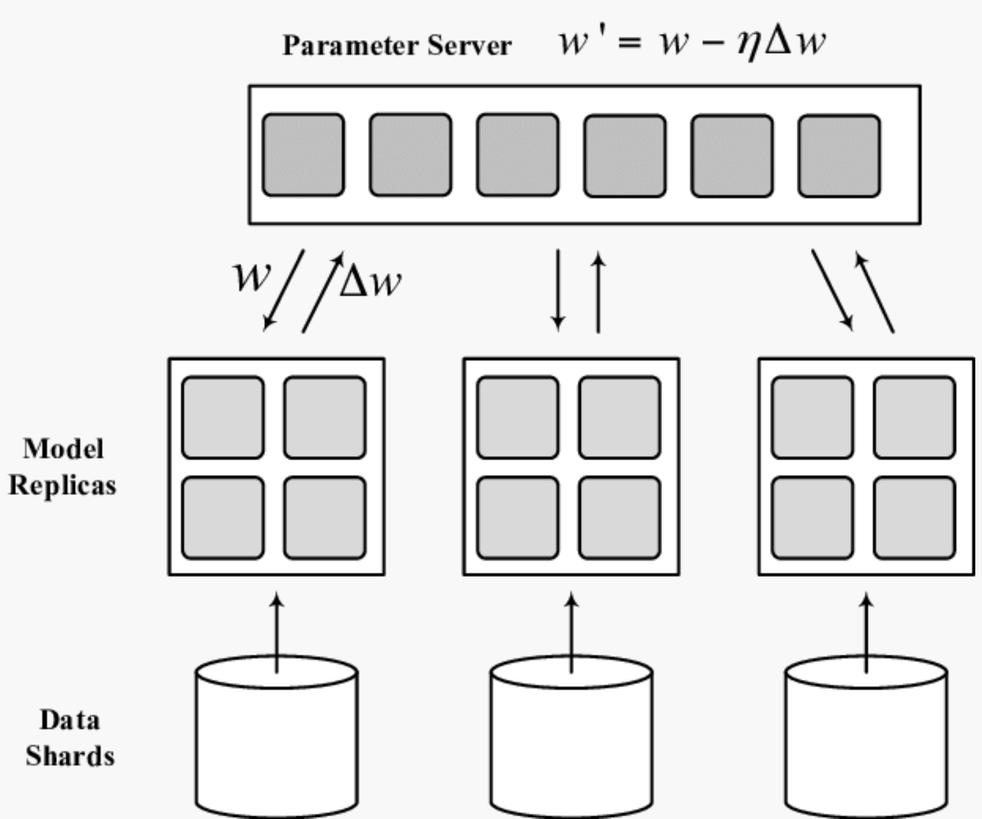
When having a cluster of workers,

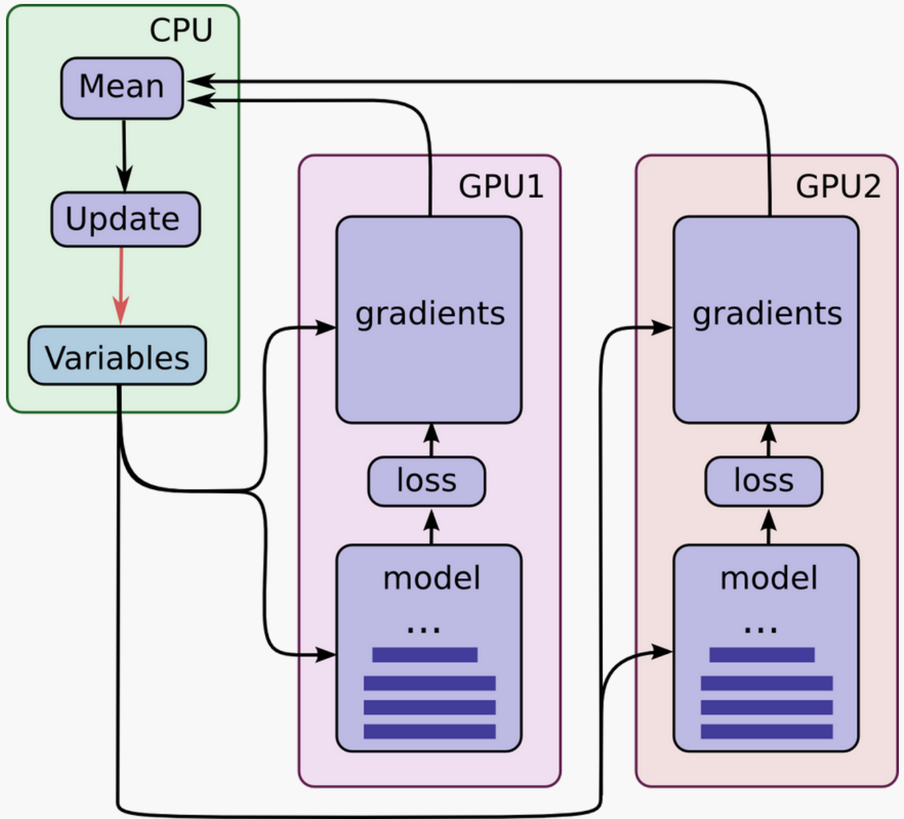
we can assign a different role to each one.

We designate some devices to act as parameter servers and the rest as training workers.

The servers : hold the parameters of our model and are responsible for updating them.

Training workers: run the actual training loop and produce the gradients and the loss from the data.





5

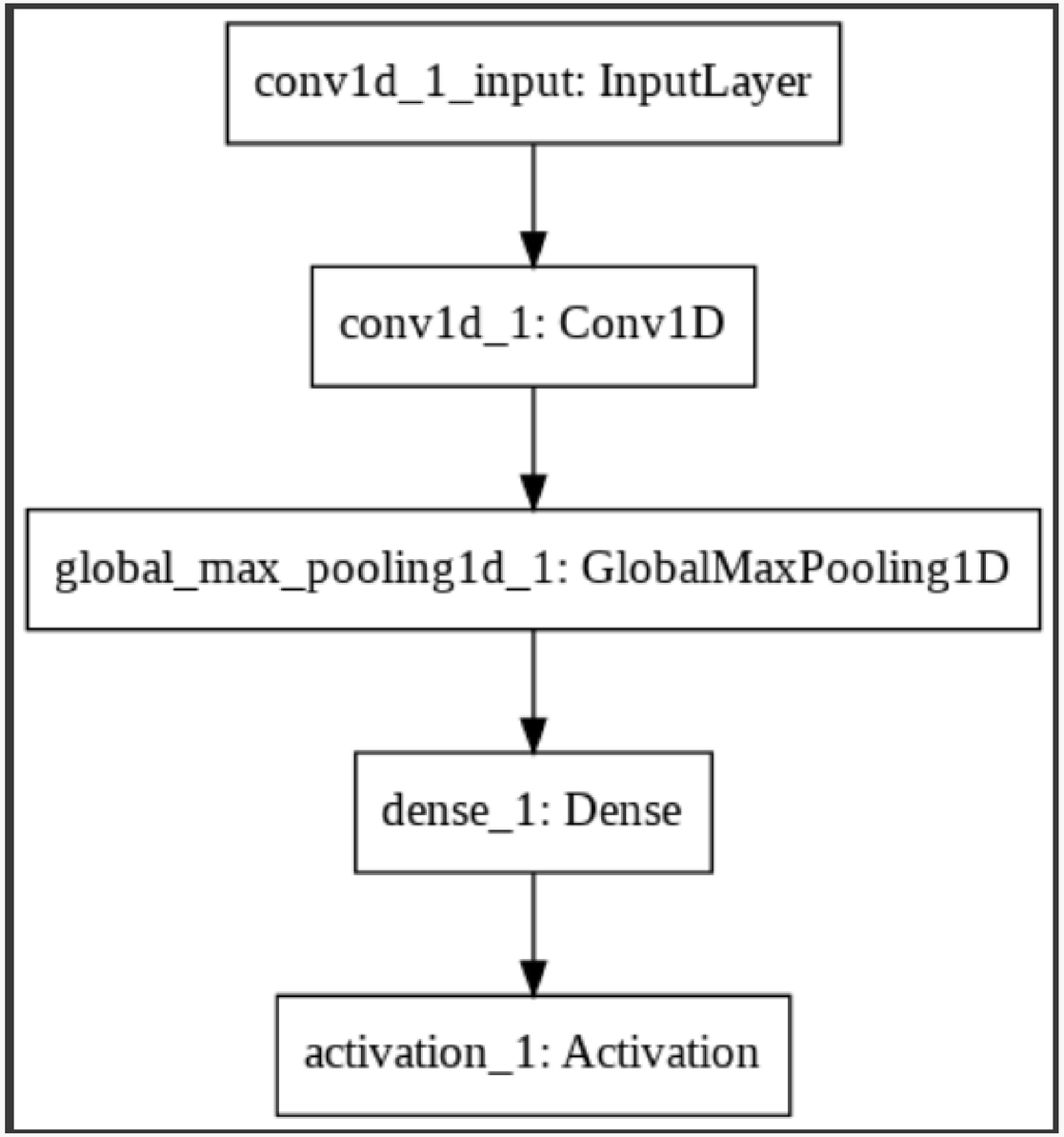
IMPLEMENTATION AND RESULTS

OUR DATASET

The dataset used in this project is simulated and generated by **SIMDNA** (simulated datasets of dna).

The simulated data contains positive instances of the TAL1 motif (is a T-cell acute lymphocytic leukemia protein) and the negative set will be random sequences.

Dataset Parts	Shape
X train	(12800, 500, 4)
Y train	(12800, 1)
X test	(3200, 500, 4)
Y test	(3200, 1)

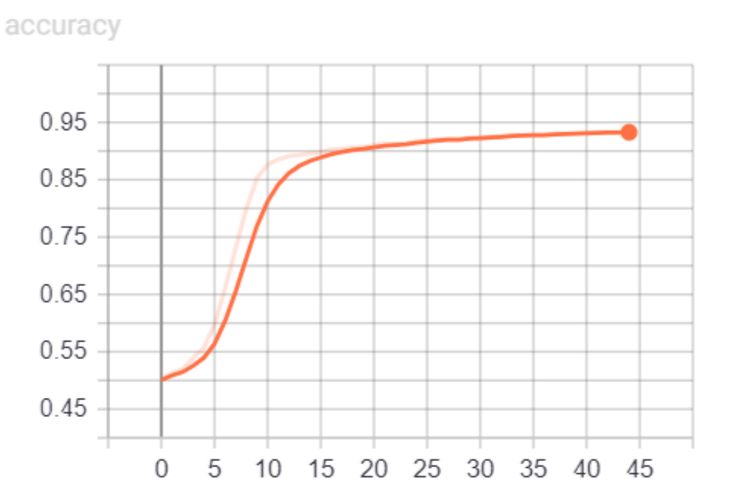


OUR MODEL

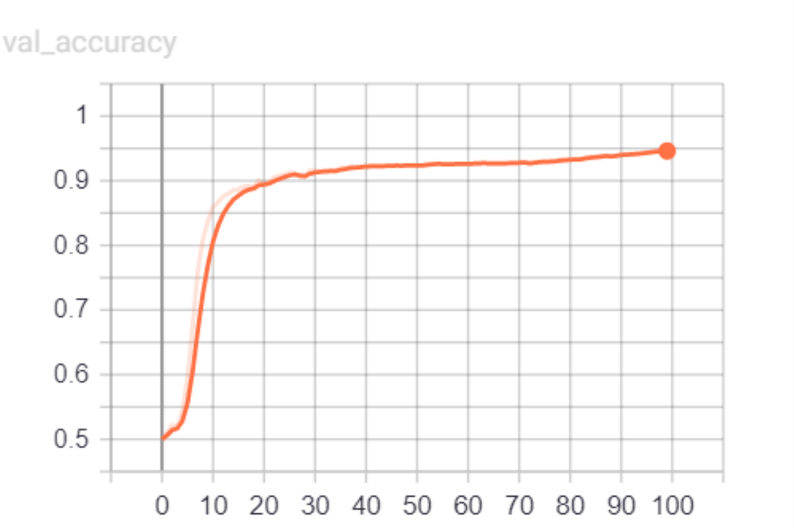
We used adam optimizer since it have faster computation time, and require fewer parameters for tuning.

And since our model is a binary classification model, we use binary crossentropy as a loss function and finally early stopping to prevent over fitting

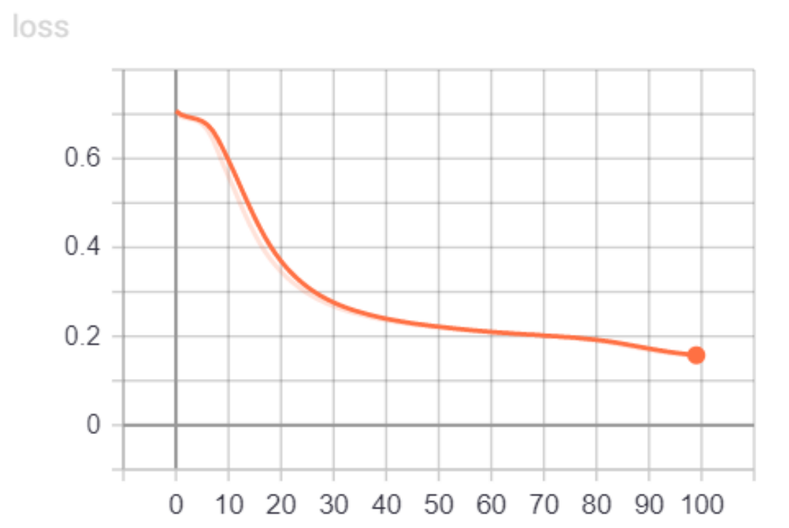
TRAINING AND TEST RESULTS



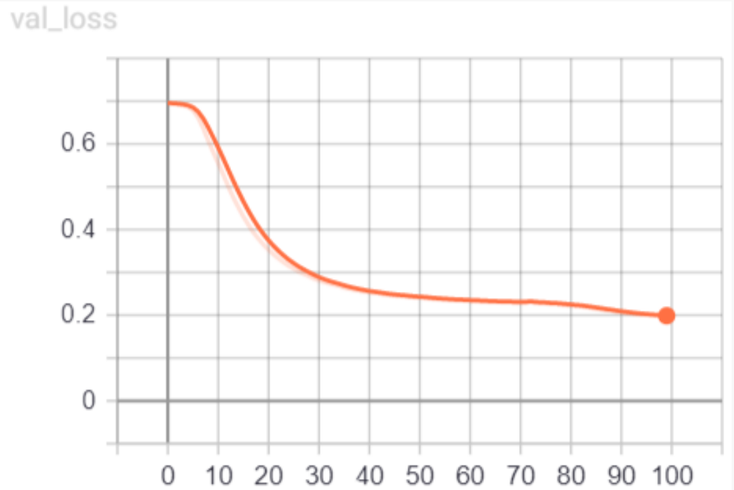
Train Accuracy



Test Accuracy



Train Loss

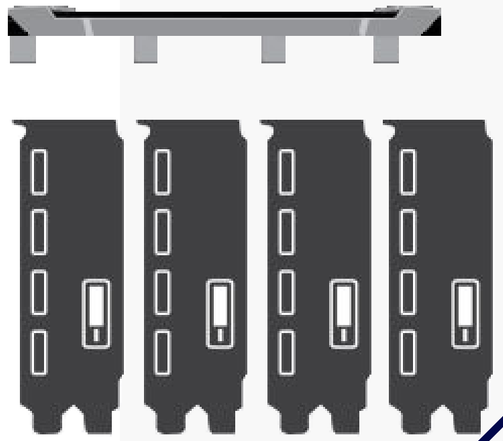


Test Loss

WORKBENCH FOR PARALLELIZATION

For scalability, we used a virtual machine running on **Lambda GPU Cloud** equipped with 4 NVIDIA RTX A6000 GRAPHICS CARD.

Lambda



4 WAY SLI

RESULTS

We trained our model using the **Mirrored Strategy** method provided by **Tensorflow** for synchronous data parallelism.

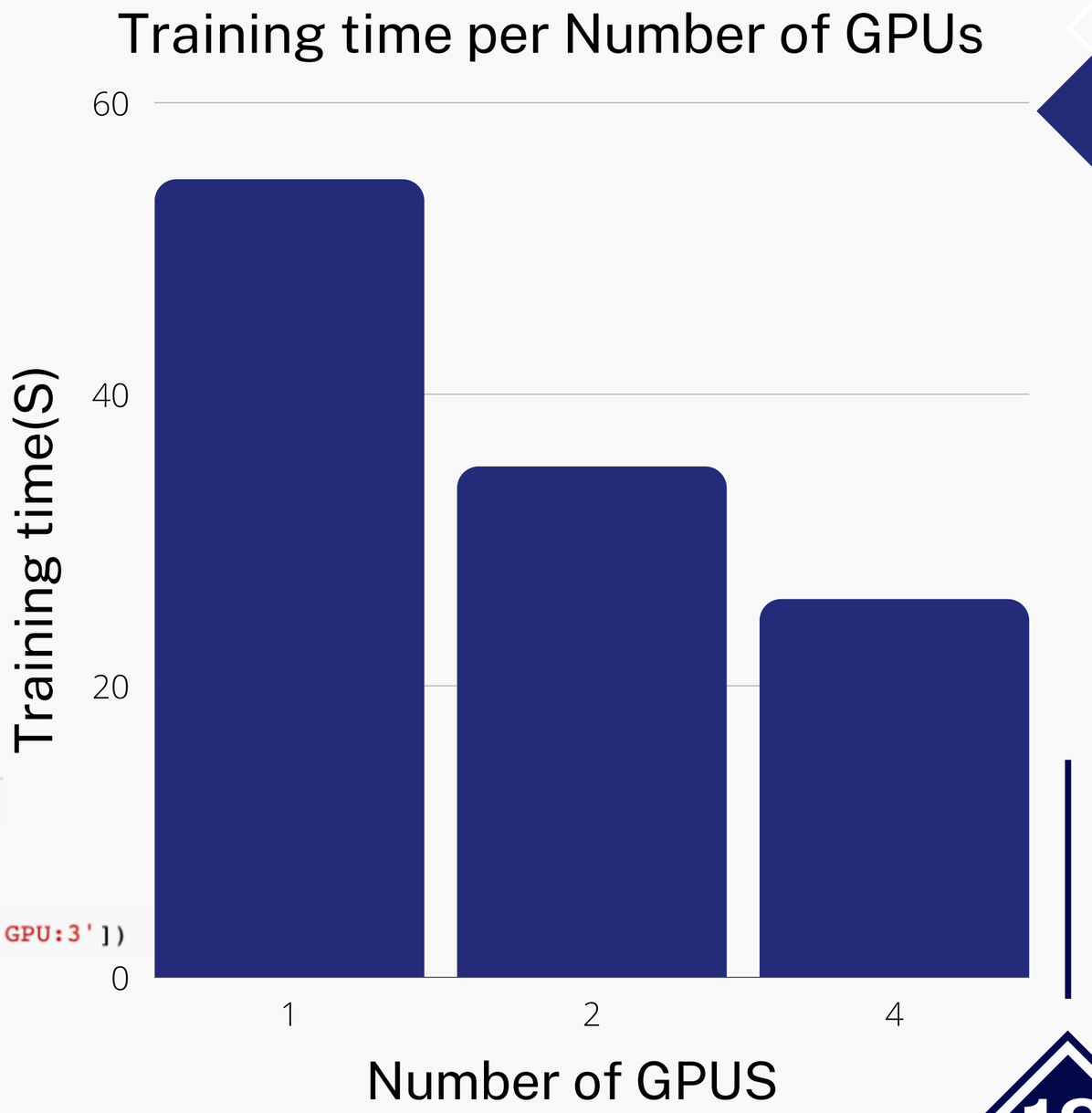
We used for this training, 1GPU, 2 GPUs and finally 4GPUs.

All the GPUs were provided by **Lambda GPU Cloud**.

```
strategy = tf.distribute.MirroredStrategy(['GPU:0'])
```

```
strategy = tf.distribute.MirroredStrategy(['GPU:0', 'GPU:1'])
```

```
strategy = tf.distribute.MirroredStrategy(['GPU:0', 'GPU:1', 'GPU:2', 'GPU:3'])
```



6

CONCLUSION AND PERSPECTIVES

CONCLUSION

Artificial Intelligence is not limited to a single domain.

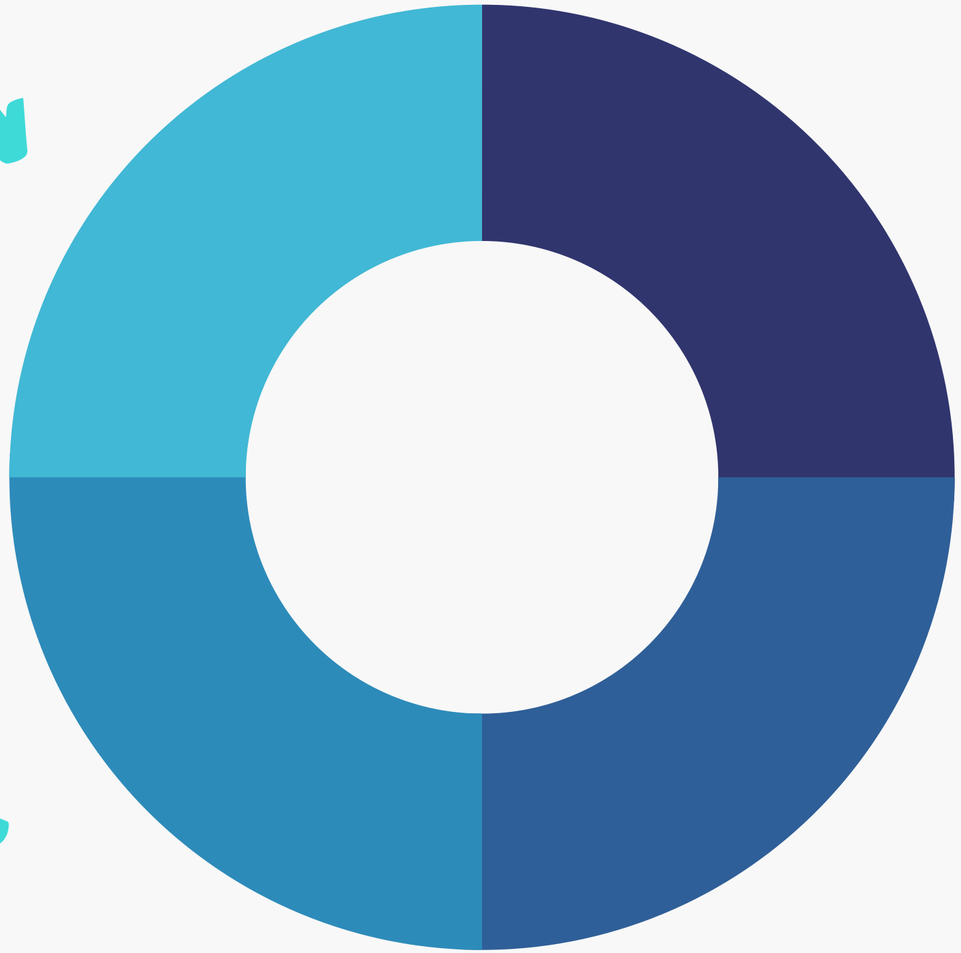
Convolutional neural networks have proven their performance for the diagnosis of anomalies from genes and DNA sequences.

To improve the training time of complex models and large datasets, we can distribute our data through parallelization strategies.

PERSPECTIVES

Training on TPUs

Try distributed learning on more complexe models



Using parallelization methods on Blockchain databases.

Try distributed training methods on RNN architecture.

THANK YOU

FOR YOUR ATTENTION

Diamonds are made under pressure