“High Performance Computing: Need for Neural Networks”

Dr. Chhavi Dhiman
Assistant Professor
Department of Electronics & Communication Engineering
Delhi Technological University, Delhi
(formerly known as Delhi College of Engineering)

Email: chhavi.dhiman@dtu.ac.in

Google Scholar, *Homepage*, *LinkedIn*

OUTLINE

- Introduction : What is HPC?
- CPUs vs GPUs
- Insight of NNs
- Computer Vision Applications
- Deep Frameworks
- Conclusion

1

High Performance Computing : Introduction

HPC: INTRODUCTION

High Performance Computing (HPC)

The use of the most efficient algorithms on computers capable of the highest performance to solve the most demanding problems.

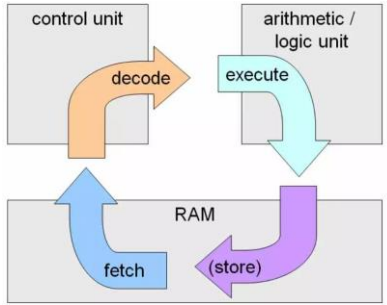
INTRODUCTION

Central Processing Unit (CPUs)

A few standard components of CPUs

1. Cores ➔ The central architecture of the CPU is the “core,” where all computation and logic happens. Initially, all CPUs were single-core, but with the proliferation of multi-core CPUs, we’ve seen an increase in processing power.

Cores support **rapid switching** between hundreds of different tasks per second. That’s why your computer can run **multiple programs**, display a desktop, connect to the internet, and more all at the same time.



Two cores→ Dual Core

Four cores→ Quad Core

Six cores→ Hexa Core

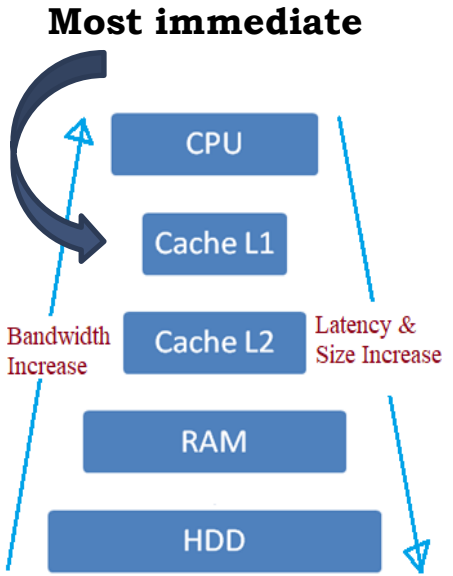
Eight cores→ Octa Core

INTRODUCTION

Central Processing Unit (CPUs)

2. Cache → Cache is super-fast memory built either within the CPU or in CPU-specific motherboards to **facilitate quick access to data** the CPU is currently using.

Since CPUs work so fast to complete **millions of calculations per second**, they supported with **ultra-fast and expensive memory** to do it—memory that is much faster than hard drive storage or even the fastest RAM.



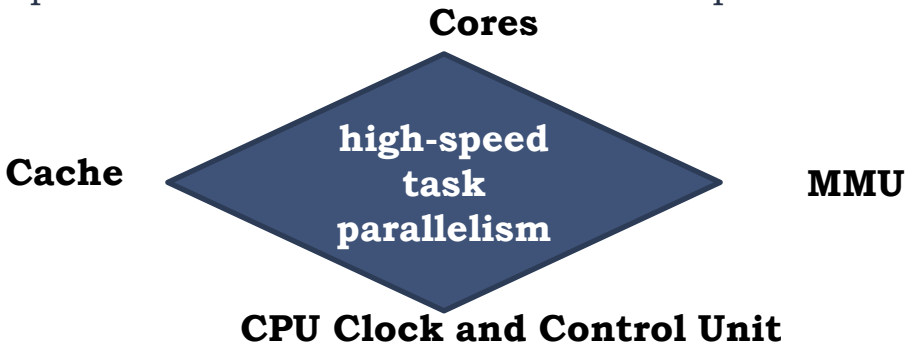
Cache L1→ fastest and L3 → slowest

INTRODUCTION

Central Processing Unit (CPUs)

3. Memory Management Unit (MMU) ➔ It controls data movement between the CPU and RAM during the instruction cycle.

4. CPU Clock and Control Unit: Every CPU works on synchronizing processing tasks through a clock. So, the higher the CPU clock rate, the faster it will run and quicker processor-intensive tasks can be completed.



GPUs

Graphical Processing Unit (GPUs)

GPUs offers a way to continue accelerating applications — such as graphics, supercomputing and AI — by dividing tasks among many processors.

Key part of modern supercomputing

“Such accelerators are critical to the future of semiconductors”

According to *John Hennessey and David Patterson, winners of the 2017 A.M. Turing Award* and authors of *“Computer Architecture: A Quantitative Approach”* the seminal textbook on microprocessors

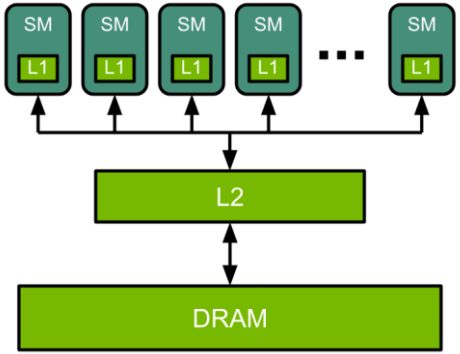
GPUs

Graphical Processing Unit (GPUs)

1. The GPU is a highly parallel processor architecture, composed of processing elements and a memory hierarchy.

2. At a high level, NVIDIA® GPUs consist of a number of **Streaming Multiprocessors (SMs)**, **on-chip L2 cache**, and **high-bandwidth DRAM**.

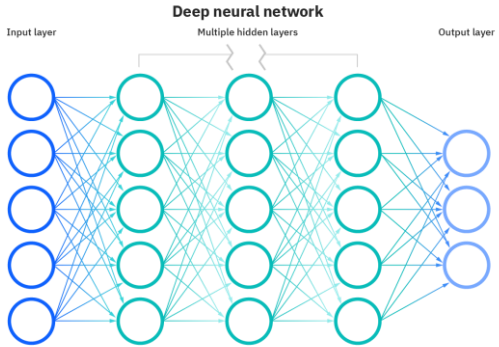
3. Arithmetic and other instructions are executed by the SMs; data and code are accessed from DRAM via the L2 cache



Example: An NVIDIA A100 GPU
108 SMs, a 40 MB L2 cache, and
up to 2039 GB/s bandwidth from 80 GB of HBM2 memory.

Neural Networks

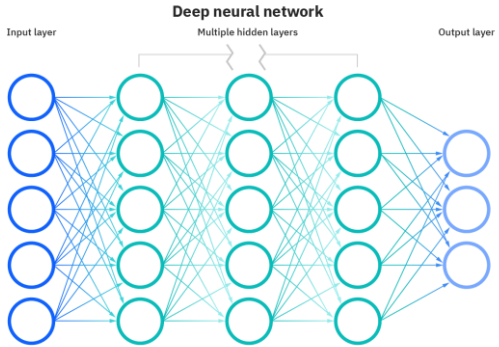
1. The name and structure of NNs are inspired by the human brain, mimicking the way that biological neurons signal to one another.
2. Large amount of data needs to be fed to neural networks, in order to train them to perform the tasks i.e. too complicated for any human coder to describe.
3. It is large data that provides generalisability of the problem No. of parameters.
4. It demands large storage space and also large computation capacity to process the entire bulk of data quickly.



Neural Networks

Important parameters that demand computational power:

1. No. of layers
2. No. of neurons in each layer
3. Batch size



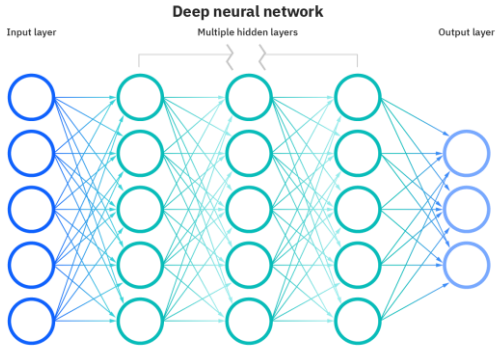
Neural Networks

Important parameters that demand computational power:

1. No. of layers

2. No. of neurons in each layer

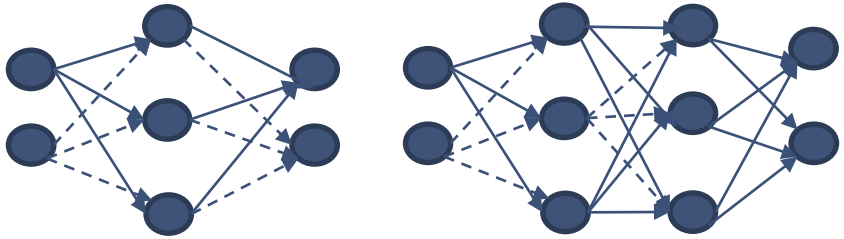
3. Batch size



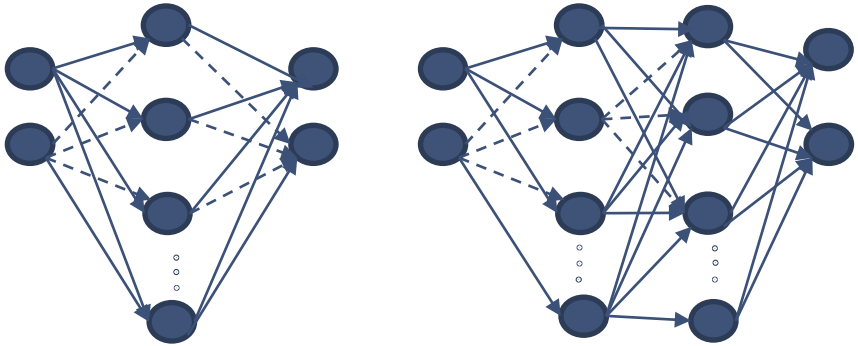
Shallow to Deep NNs

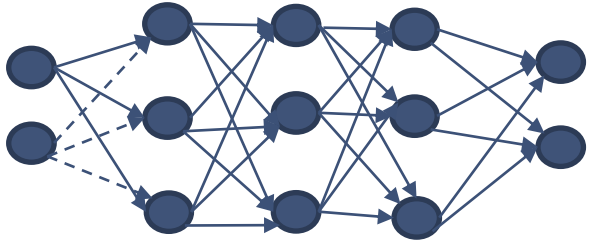
A neural network with two or more hidden layers properly takes the name of a deep neural network, in contrast with shallow neural networks that comprise of only one hidden layer.

Shallow to Deep NNs



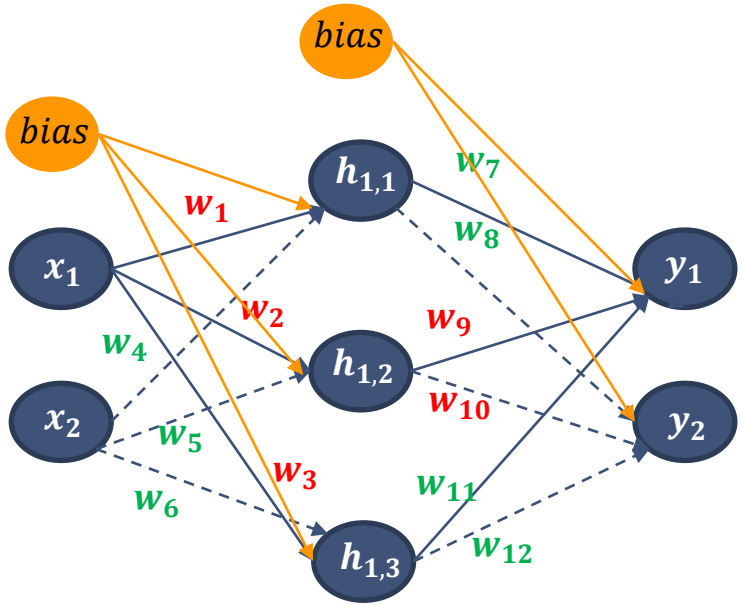
Shallow NNs





Simplest Deep NN

Shallow NN



Assuming:

i = number of neurons in input layer

h = number of neurons in hidden layer

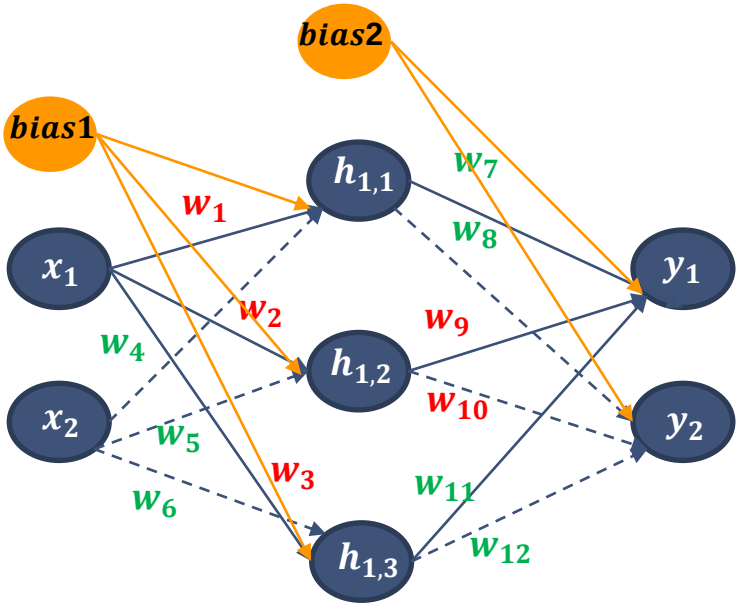
o = number of neurons in output layer

$$i = 2\ ;\ h = 3\ ;\ o = 2$$

Number of connections between the first and second layer: $\mathbf{2 \times 3 = 6}$, $(i \times h)$

Number of connections between the second and third layer: $\mathbf{3 \times 2 = 6}$, $(h \times o)$

Shallow NN



$i = 2\ ;\ h = 3\ ;\ o = 2$

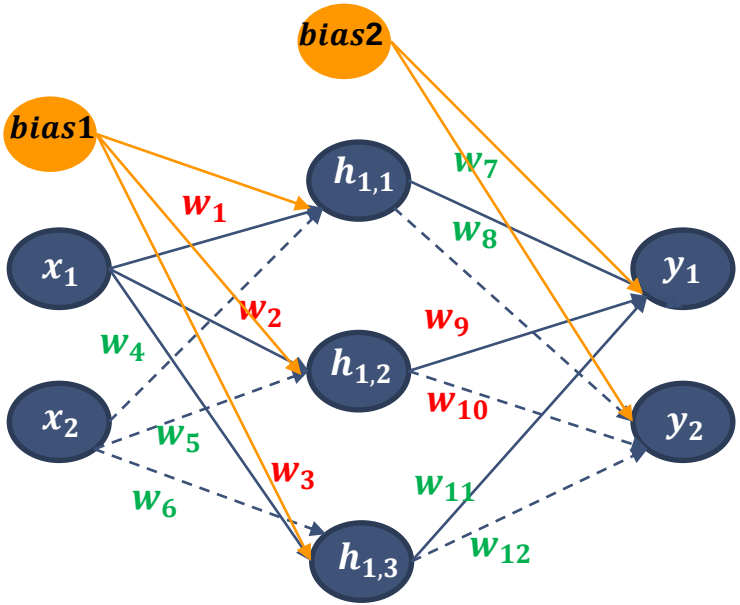
Number of connections between the first and second layer: $2 \times 3 = 6,\ (i \times h)$

Number of connections between the second and third layer: $3 \times 2 = 6,\ (h \times o)$

Number of connections between the **bias** of the **first layer** and the neurons of the **second layer** $= 1 \times 3,\ (bias1 \times h)$

Number of connections between the **bias of the second layer** and the neurons of the **third layer** $= 1 \times 2, (bias \times o)$

Shallow NN



Input layer
2 units

hidden layer
3 units

output layer
2 units

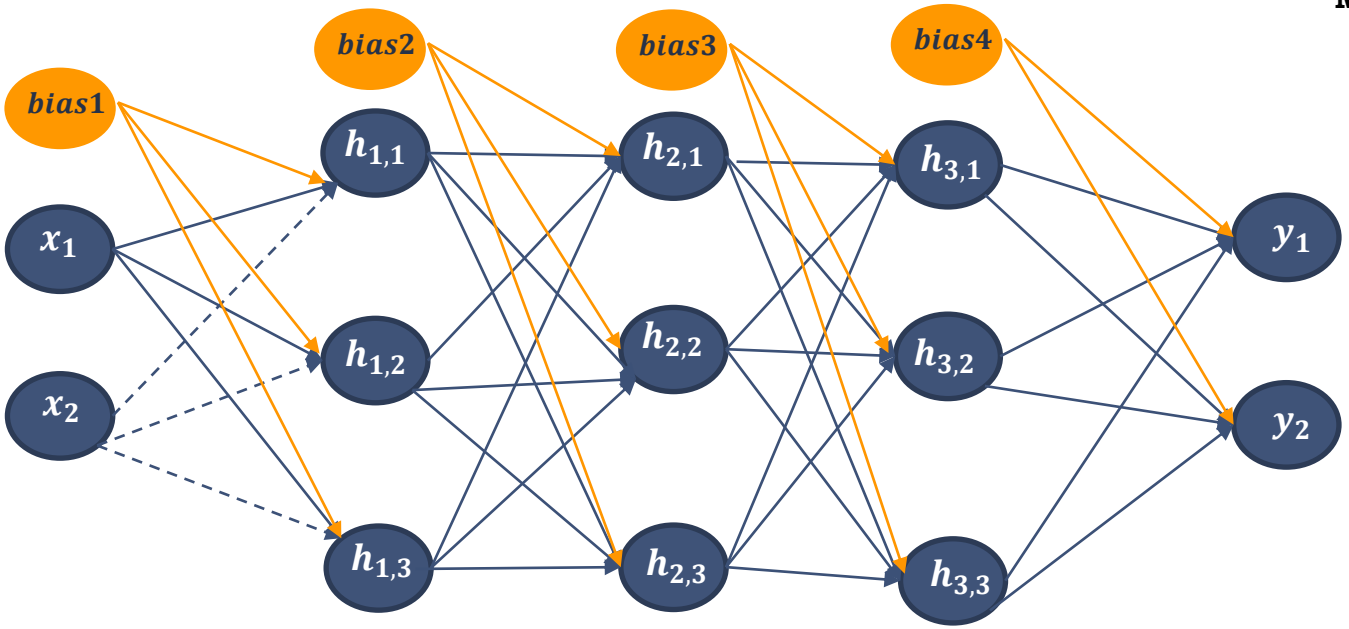
MODEL 1

Number of trainiable parameters in the shallow network

$$= 2 \times 3 + 3 \times 2 + 1 \times 3 + 1 \times 2 = 17$$

Deep NN

MODEL 2



Input layer
2 units

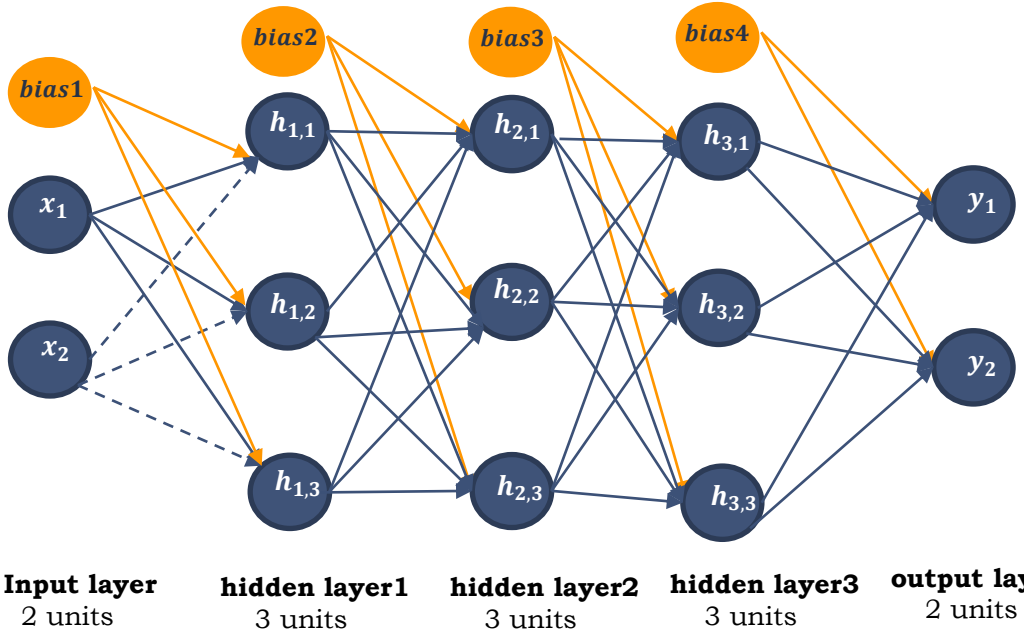
hidden layer1
3 units

hidden layer2
3 units

hidden layer3
3 units

output layer
2 units

Deep NN



Input layer
2 units

hidden layer1
3 units

hidden layer2
3 units

hidden layer3
3 units

output layer
2 units

MODEL 2

The total number of parameters in a deep NN with 3 hidden layers is given by:

$$= (i \times h1 + h1 \times h2 + h2 \times h3 + h3 \times o) + 1 \times h1 + 1 \times h2 + 1 \times h3 + 1 \times o$$

$$= (2 \times 3 + 3 \times 3 + 3 \times 3 + 3 \times 3) + (1 \times 3) + (1 \times 3) + (1 \times 3) + (1 \times 3) = 45$$

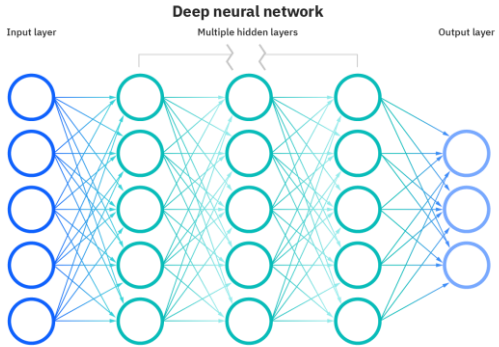
Neural Networks

Important parameters that demand computational power:

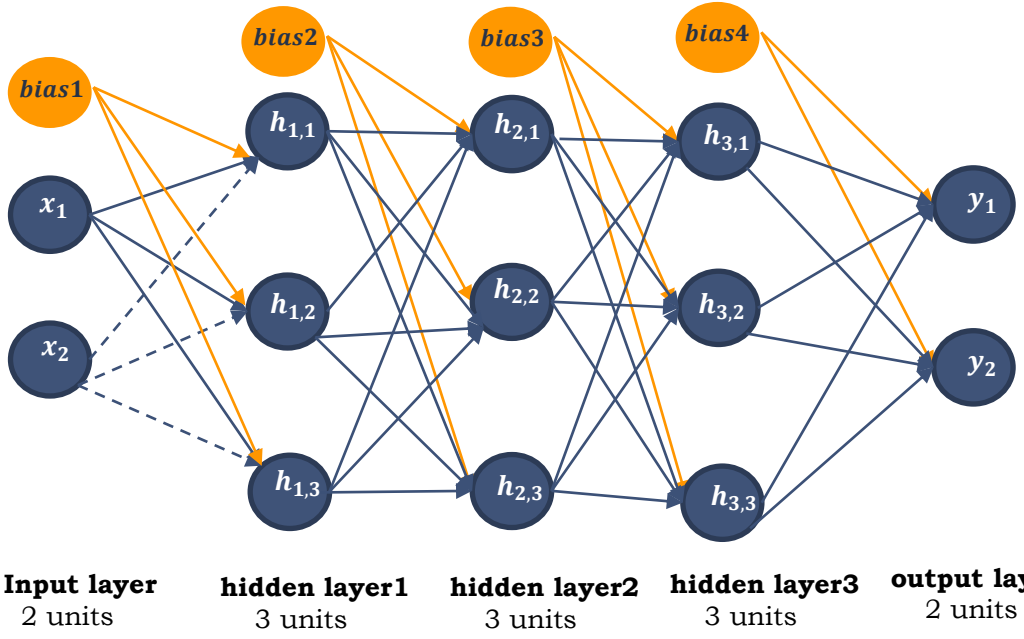
1. No. of layers

2. **No. of neurons in each layer**

3. Batch size



Deep NN



Input layer
2 units

hidden layer1
3 units

hidden layer2
3 units

hidden layer3
3 units

output layer
2 units

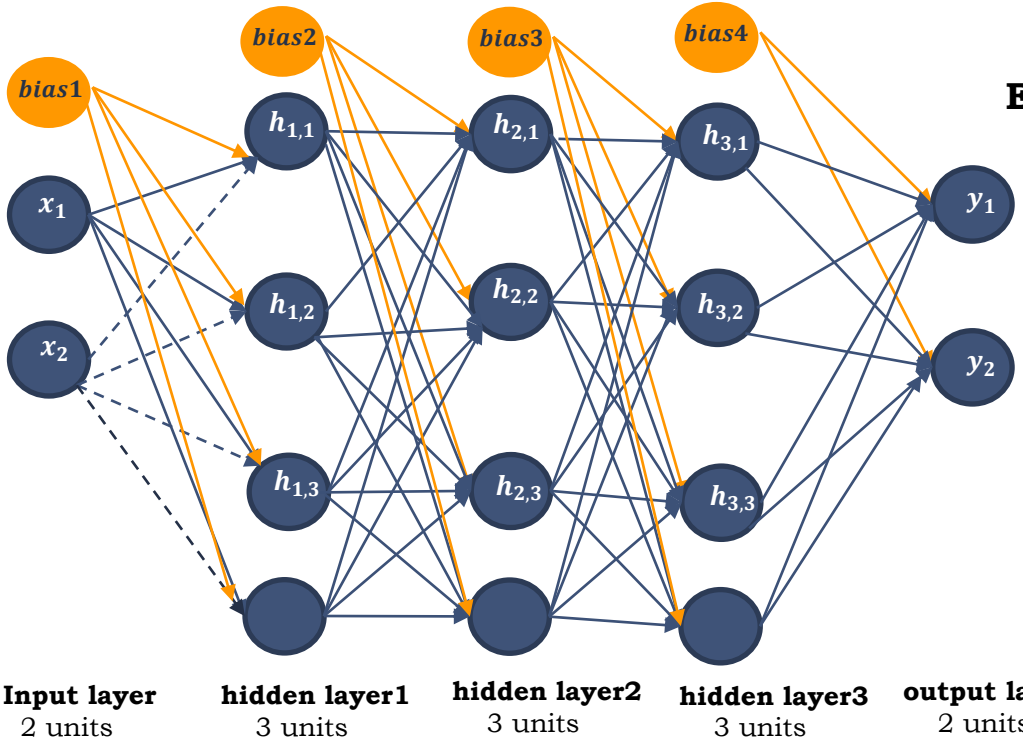
Model 3
Effect of no. of neurons in each hidden layer

$$= i \times h1 + \sum_{k=1}^{n-1}(h_k \times h_{k+1}) + h_n \times o + \sum_{k=1}^{n} h_k + o$$

i=512
$h_k = 20, k \in (1,3)$
$o = 2$

No. of parameters
$= 512 \times 20 + (20 \times 20 \times 2) + \mathbf{20 \times 2} = \mathbf{10240} + \mathbf{800} + \mathbf{40} = \mathbf{11,080}$

Deep NN



Model 4
Effect of no. larger no. of hidden layer

$$No.\,of\ parameters$$

$$= i \times h1 + \sum_{k=1}^{n-1}(h_k \times h_{k+1}) + h_n \times o + \sum_{k=1}^{n} h_k + o$$

i=512
$h_k = 20, k \in (1,10)$
$o = 2$

No. of parameters= $512 \times 20 + (20 \times 20 \times 9) + 20 \times 2 = 10240 + \mathbf{3600} + 40 = \mathbf{13,880}$

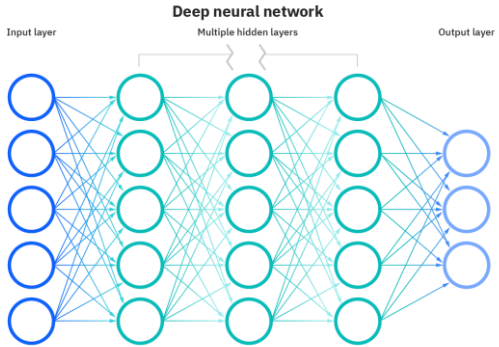
Neural Networks

Important parameters that demand computational power:

1. No. of layers

2. No. of neurons in each layer

3. **Batch size**

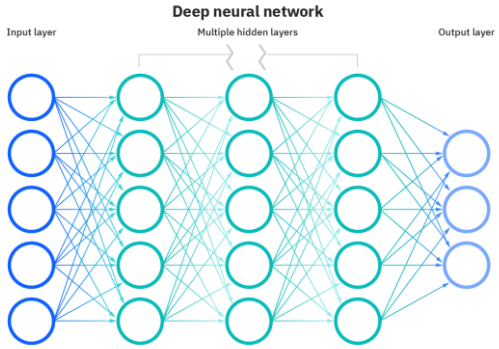


Neural Networks

Impact of Batch Size:

1. Using a **batch size** equal to the **entire dataset** guarantees **convergence** to the **global optima** of the objective function. However, this is at the cost of **slower, empirical convergence** to that optima.
2. Using **smaller batch** sizes have been empirically shown to have **faster convergence** to "good" solutions.

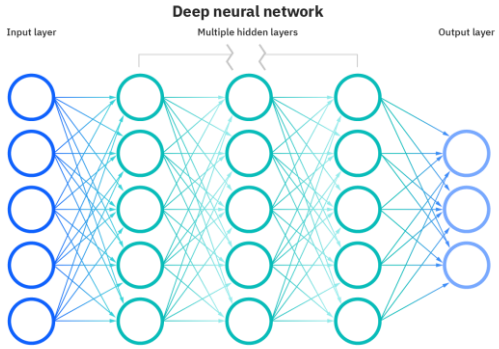
"start learning before having to see all the data."



Neural Networks

Impact of Batch Size:

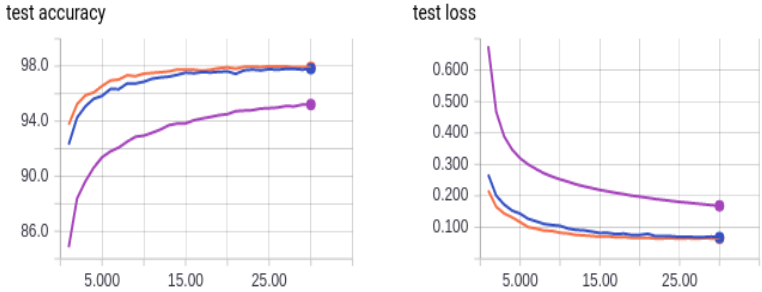
1. With a smaller batch size is that the model is **not** guaranteed to **converge** to the **global optima**.
2. It will **bounce around the global optima.**
3. Therefore, under no computational constraints, it is often advised that
   a) one starts at a small batch size, reaping the benefits of faster training dynamics,
   b) steadily grows the batch size through training, also reaping the benefits of guaranteed convergence.



Neural Networks

Impact of Batch Size:

1. Total no. of samples: 1024
2. Batch size=1024, 64, 16, 4
3. Steps per epochs=1024/1024=1 steps
   = 1024/64=16 steps
   = 1024/16= 64 steps
   = 1024/4= 256 steps



Orange: batch size=64
Purple: batch size =1024

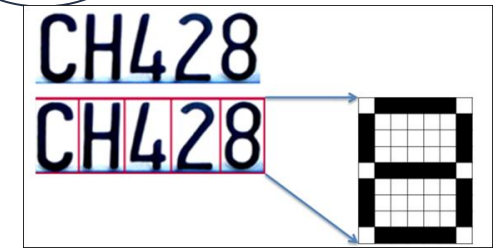
Comparison Chart for CNNs

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

Reference

Computer Vision Applications

How many person?
Who are in the scene?
What they are doing?



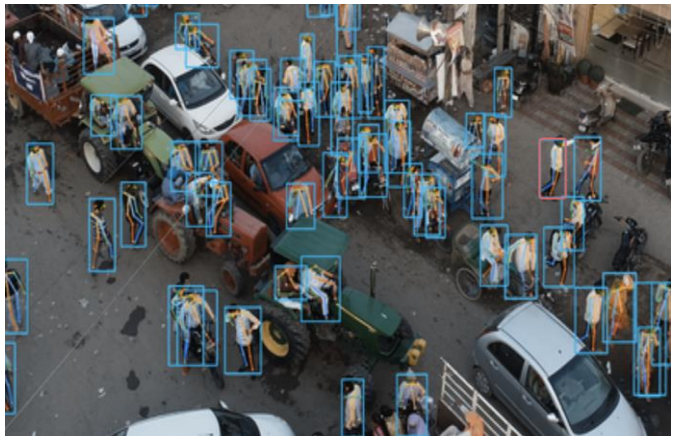
What are characters?

Who are these?

unmatched image recognition

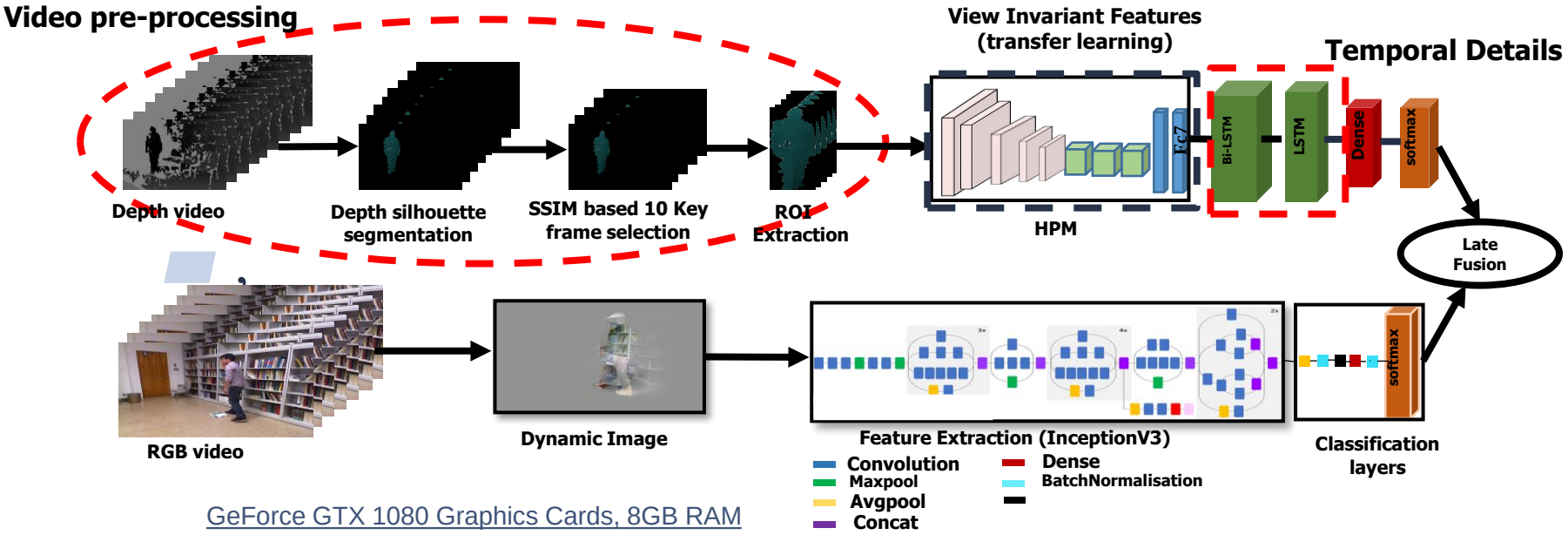
Sharbat Gula

Computer Vision Applications



Video feed: 60 mins = 60x 60= 3600 sec
25fps
No. of frames= 25x3600=90,000 frames to be monitored

Deep Framework Insight



GeForce GTX 1080 Graphics Cards, 8GB RAM

Chhavi Dhiman, *D. K. Vishwakarma, "View-invariant Deep Architecture for Human Action Recognition using Two-stream Motion and Shape Temporal Dynamics", I**EEE Transactions on Image Processing (TIP)**, Vol. 29, pp. 3835-3844, 2020.* ***Impact Factor: 9.340***

Conclusion

- HPC is the demand for today.
- Computer vision based applications demands good performance provided large computation power is supported.
- The architecture must be developed while taking care of the fact that how we can constraint on the expanding demands of high computation.