Computing Infrastructures
Theory

Christian Rossi

Academic Year 2023-2024

Abstract

The course topics are:

- Hardware infrastructure of datacenters:
  - Basic components, rack structure, cooling.
  - Hard Disk Drive and Solid State Disks.
  - RAID architectures.
  - Hardware accelerators.
- Software infrastructure of datacenters:
  - Virtualization: basic concepts, technologies, hypervisors and containers.
  - Computing Architecture: Cloud, Edge and Fog Computing.
  - Infrastructure, platform and software-as-a-service.
- Methods:
  - Scalability and performance of datacenters: definitions, fundamental laws, queuing network theory basics.
  - Reliability and availability of datacenters: definitions, fundamental laws, reliability block diagrams.

Contents

1 Introduction **1**
- 1.1 Computing infrastructures . . . 1
  - 1.1.1 Virtual machines . . . 1
  - 1.1.2 Containers . . . 2
  - 1.1.3 Summary . . . 2
- 1.2 Edge computing systems . . . 2
  - 1.2.1 Embedded PC . . . 3
  - 1.2.2 Internet of things . . . 3

2 Data centers **4**
- 2.1 Introduction . . . 4
  - 2.1.1 Warehouse-scale computers . . . 4
  - 2.1.2 Geographical distribution of data centers . . . 5
- 2.2 Warehouse-scale computers . . . 6
  - 2.2.1 Architecture . . . 6
- 2.3 Servers . . . 7
  - 2.3.1 Racks . . . 8
  - 2.3.2 Towers . . . 9
  - 2.3.3 Blade servers . . . 9
  - 2.3.4 Building structure . . . 10
- 2.4 Hardware accelerators . . . 10
  - 2.4.1 Graphical processing unit . . . 10
  - 2.4.2 Neural networks . . . 10
  - 2.4.3 Tensor processing unit . . . 13
  - 2.4.4 Field programmable gate array . . . 14

CHAPTER 1

Introduction

1.1 Computing infrastructures

Definition (*Computing infrastructure*)**.** Computing infrastructure refers to the technological framework comprising hardware and software components designed to facilitate computation for other systems and services.

Data centers encompass servers tailored for diverse functions:

- *Processing Servers.*
- *Storage Servers.*
- *Communication Servers.*

1.1.1 Virtual machines

Virtual machines offer a comprehensive stack comprising an operating system, libraries, and applications. Applications rely on a guest operating system.

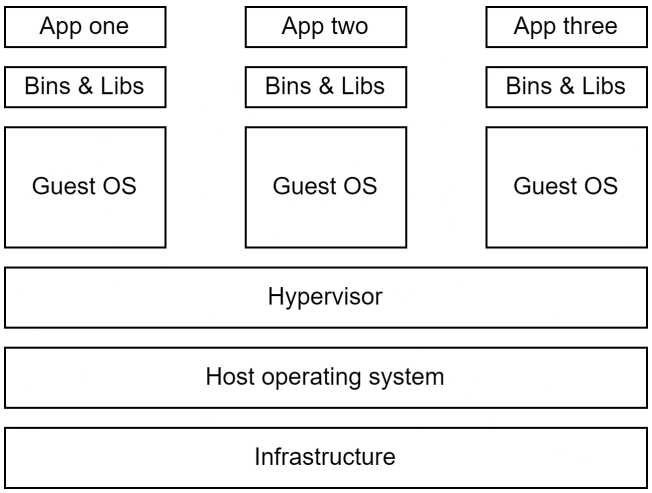


Figure 1.1: Virtual machine's structure

In this configuration, each operating system perceives the hardware as exclusively dedicated to itself. Virtualization results in significant power efficiency gains by consolidating the power consumption of individual machines, typically saving around 60%. Virtualization enables hot swaps, delivering two key advantages:

1. *Maintainability.*
2. *Availability*: overloaded machines can be supplemented by others.

1.1.2 Containers

Containers encapsulate applications along with their dependencies into a uniform unit for streamlined software development and deployment.

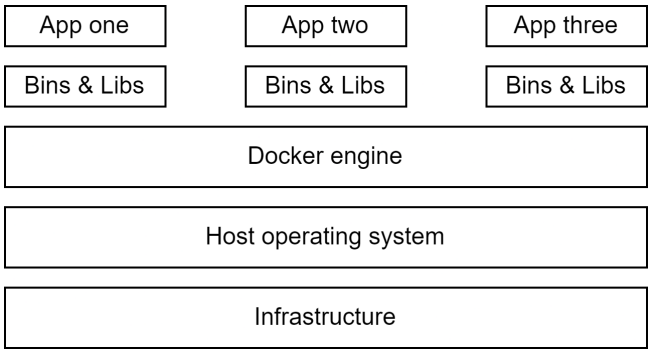


Figure 1.2: Container's structure

Containers are employed to execute specific services and offer a lighter alternative to virtual machines.

1.1.3 Summary

Data centers offer several advantages, including reduced IT costs, enhanced performance, automatic software updates, seemingly limitless storage capacity, improved data reliability, universal document accessibility, and freedom from device constraints. However, it also presents challenges such as the need for a stable internet connection, poor compatibility with slow connections, limited hardware capabilities, privacy and security concerns, increased power consumption, and delays in decision-making.

1.2 Edge computing systems

Edge computing is a distributed computing model in which data processing occurs as close as possible to where the data is generated, improving response times and saving on bandwidth. Processing data near the location where it is generated brings significant advantages in terms of processing latency, reduced data traffic, and increased resilience in case of data connection interruptions. Edge computing systems can be categorized as follows:

- *Cloud*: providing virtualized computing, storage, and network resources with highly elastic capacity.

- *Edge servers*: utilizing on-premises hardware resources for more computationally intensive data processing.
- *IoT and AI-enabled edge sensors*: enabling data acquisition and partial processing at the edge of the network.

Edge computing offers several advantages, including high computational capacity, distributed computing capabilities, enhanced privacy and security, and reduced latency in decision-making. However, it comes with drawbacks such as the requirement for a power connection and dependence on a connection with the Cloud.

1.2.1 Embedded PC

An embedded system refers to a computer system that comprises a computer processor, computer memory, and input/output peripheral devices, all serving a specific function within a larger mechanical or electronic system. Advantages of this approach include its pervasiveness in computing, high-performance units, availability of development boards, ease of programming similar to personal computers, and the support of a large community. On the other hand, it has disadvantages such as relatively high power consumption and the necessity for some hardware design work to be done.

1.2.2 Internet of things

The internet of things (IoT) encompasses devices equipped with sensors, processing capabilities, software, and other technologies. These devices are designed to connect and exchange data with other devices and systems over the internet or other communication networks. Advantages of IoT devices include their high pervasiveness, wireless connectivity, battery-powered operation, low costs, and their ability to sense and actuate. However, these devices also come with several disadvantages, such as their low computing ability, constraints on energy usage, limitations in memory (RAM/FLASH), and difficulties in programming them.

CHAPTER 2

Data centers

2.1 Introduction

Over the past few decades, there has been a significant shift in computing and storage, transitioning from PC-like clients to smaller, often mobile devices, coupled with expansive internet services. Concurrently, traditional enterprises are increasingly embracing cloud computing.

This shift offers several user experience improvements, including ease of management and ubiquitous access.

For vendors, Software-as-a-Service (SaaS) facilitates faster application development, making changes and improvements easier. Moreover, software fixes and enhancements are streamlined within data centers, rather than needing updates across millions of clients with diverse hardware and software configurations. Hardware deployment is simplified to a few well-tested configurations.

Server-side computing enables the swift introduction of new hardware devices, such as hardware accelerators or platforms, and supports many application services running at a low cost per user. Certain workloads demand substantial computing capability, making data centers a more natural fit compared to client-side computing.

2.1.1 Warehouse-scale computers

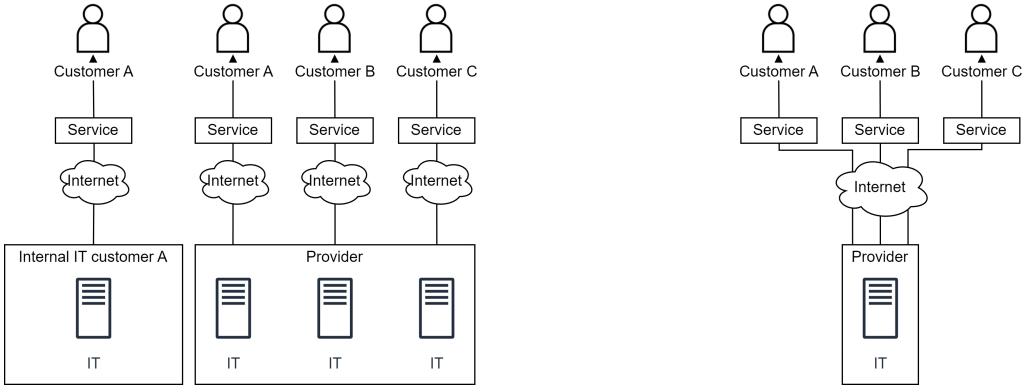
The rise of server-side computing and the widespread adoption of internet services have given rise to a new class of computing systems known as warehouse-scale computers (WSCs). In warehouse-scale computing, the program:

- Operates as an internet service.
- Can comprise tens or more individual programs.
- These programs interact to deliver complex end-user services like email, search, maps, or machine learning.

Data centers are facilities where numerous servers and communication units are housed together due to their shared environmental needs, physical security requirements, and for the sake of streamlined maintenance. Traditional data centers typically accommodate a considerable number of relatively small- or medium-sized applications. Each application operates on a

dedicated hardware infrastructure, isolated and safe guarded against other systems within the same facility. These applications typically do not communicate with one another. Moreover, these data centers host hardware and software for multiple organizational units or even different companies.

In contrast, warehouse-scale computers are owned by a single organization, employ a relatively uniform hardware and system software platform, and share a unified systems' management layer.



(a) Data center (b) Data warehouse

Figure 2.1: Structures of data centers and data warehouses

Warehouse-scale computers operate a reduced quantity of highly expansive applications, often internet services. Their shared resource management infrastructure affords considerable deployment flexibility. Designers are driven by the imperatives of homogeneity, single-organization control, and cost efficiency, prompting them to adopt innovative approaches in crafting WSCs.

Originally conceived for online data-intensive web workloads, warehouse-scale computers have expanded their capabilities to drive public cloud computing systems, such as those operated by Amazon, Google, and Microsoft. These public clouds do accommodate numerous small applications, resembling a traditional data center setup. However, all these applications leverage virtual machines or containers and access shared, large-scale services for functionalities like block or database storage and load balancing, aligning seamlessly with the WSC model.

The software operating on these systems is designed to run on clusters comprising hundreds to thousands of individual servers, far surpassing the scale of a single machine or a single rack. The machine itself constitutes this extensive cluster or aggregation of servers, necessitating its consideration as a single computing unit.

2.1.2 Geographical distribution of data centers

Frequently, multiple data centers serve as replicas of the same service, aiming to reduce user latency and enhance serving throughput. Requests are typically processed entirely within one data center.

Definition (*Geographic area*)**.** Geographic areas partition the world into sectors, each defined by geopolitical boundaries.

Within each geographic area, there are at least two computing regions. Customers perceive regions as a more detailed breakdown of the infrastructure. Notably, multiple data centers within the same region are not externally visible. The perimeter of each computing region

is defined by latency (with a round trip latency of two milliseconds), which is too far for synchronous replication but sufficient for disaster recovery.

Definition (*Availability zone*)**.** Availability zones represent more granular locations within a single computing region.

They enable customers to operate mission-critical applications with high availability and fault tolerance to datacenter failures by providing fault-isolated locations with redundant power, cooling, and networking. Application-level synchronous replication among availability zones is implemented, with a minimum of three zones being adequate for ensuring quorum.

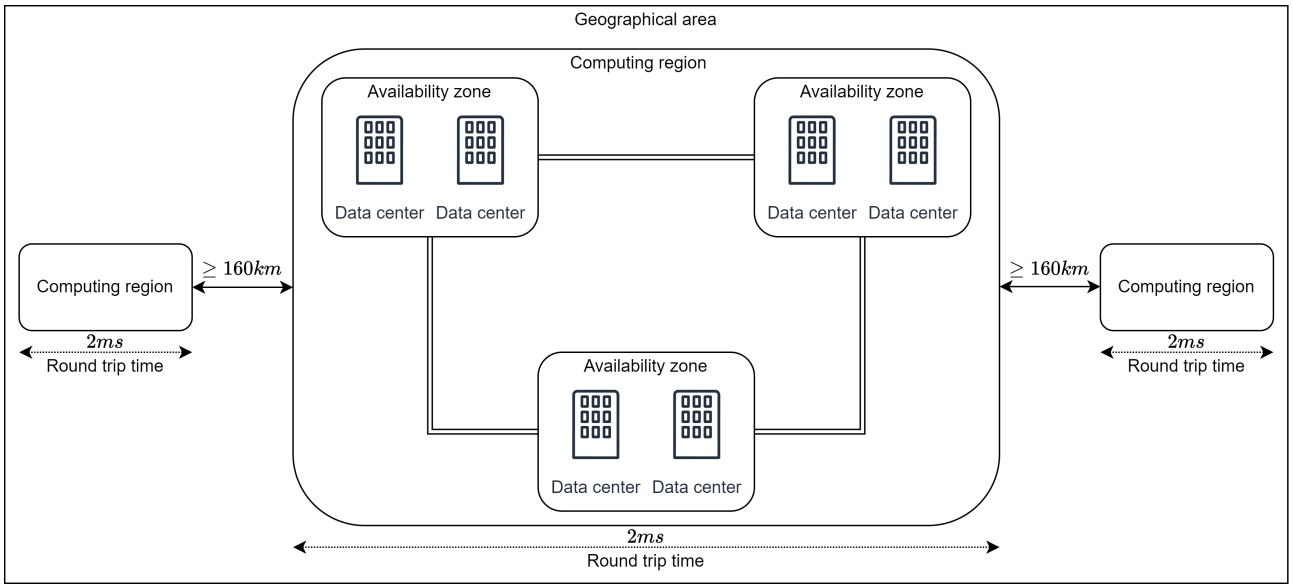


Figure 2.2: Geographical area structure

2.2 Warehouse-scale computers

Services offered by warehouse-scale computers need to ensure high availability, usually targeting a minimum uptime of 99.99%, equating to one hour of downtime per year. Maintaining flawless operation is challenging when managing a vast array of hardware and system software components. Therefore, WSC workloads must be crafted to gracefully handle numerous component faults, minimizing or eliminating any adverse effects on service performance and availability.

2.2.1 Architecture

While the hardware implementation of WSCs may vary considerably, the architectural organization of these systems remains relatively consistent.

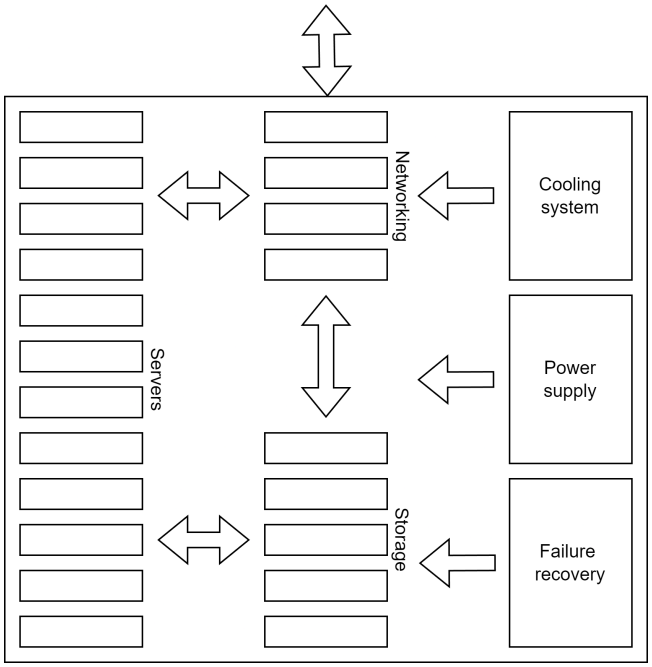


Figure 2.3: Architecture of warehouse-scale computers

Servers Servers resemble standard PCs but are designed with form factors that enable them to fit into racks, such as rack (one or more), blade enclosure format, or tower configurations. They can vary in terms of the number and type of CPUs, available RAM, locally attached disks (HDD, SSD, or none), as well as the inclusion of other specialized devices like GPUs, DSPs, and coprocessors.

Storage Disks and Flash SSDs serve as the fundamental components of contemporary WSC storage systems. These devices are integrated into the data-center network and overseen by advanced distributed systems. Examples of storage configurations include Direct Attached Storage (DAS), Network Attached Storage (NAS), Storage Area Networks (SAN), and RAID controllers.

Networking Communication equipment facilitates network interconnections among devices within the system. These include hubs, routers, DNS or DHCP servers, load balancers, switches, firewalls, and various other types of devices.

2.3 Servers

Servers housed within individual shelves form the foundational components of warehouse-scale computers. These servers are interconnected through hierarchical networks and are sustained by a shared power and cooling infrastructure.

They resemble standard PCs but are designed with form factors that enable them to fit into shelves, such as rack (one or more), blade enclosure format, or tower configurations. Servers are typically constructed in a tray or blade enclosure format, housing the motherboard, chipset, and additional plug-in components.

Motherboard The motherboard offers sockets and plug-in slots for installing CPUs, memory modules (DIMMs), local storage (such as Flash SSDs or HDDs), and network interface cards (NICs) to meet various resource requirements.

The chipset and additional components of these systems include:

- *Number and type of CPUs*: these systems support a varying number of CPU sockets, typically ranging from 1 to 8, and can accommodate processors such as Intel Xeon Family, AMD EPYC, etc.
- *Available RAM*: they offer a range of DIMM slots, typically from 2 to 192, for memory modules.
- *Locally attached disks*: these systems come with between 1 and 24 drive bays for local storage. They support both HDD and SSD options, with specific configurations detailed in lectures. Users can choose between SAS for higher performance (albeit at a higher cost) or SATA for entry-level servers.
- *Other special purpose devices*: these systems can integrate specialized components like GPUs or TPUs, with support for various models including NVIDIA Pascal, Volta, A100, etc.
- *Form factor*: they come in different sizes, ranging from 1U to 10U, and can also be configured as tower systems.

2.3.1 Racks

Racks serve as specialized shelves designed to house and interconnect all IT equipment. They are utilized for storing rack servers and are measured in rack units, denoted as U, with one rack unit (1U) equivalent to 44.45 mm (1.75 inches). One of the advantages of using racks is that they allow designers to stack additional electronic devices alongside the servers. IT equipment must adhere to specific sizes to fit into the rack shelves.

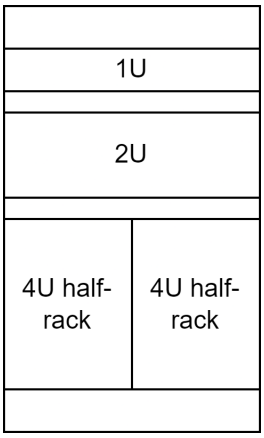


Figure 2.4: Rack elements dimensions

The rack serves as the shelf that securely holds together tens of servers. It manages the shared power infrastructure, including power delivery, battery backup, and power conversion. The width and depth of racks vary across WSCs, with some adhering to classic 19-inch width and 48-inch depth, while others may be wider or shallower. It is often convenient to connect

network cables at the top of the rack, leading to the adoption of rack-level switches appropriately called Top of Rack (TOR) switches.

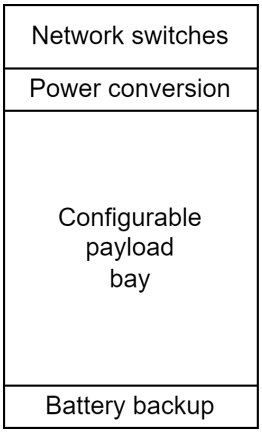


Figure 2.5: Rack structure

Advantages of using racks include the ease of failure containment, as identifying and replacing malfunctioning servers is a straightforward process. Additionally, racks offer simplified cable management, making it efficient to organize cables. Moreover, they provide cost-effectiveness by offering computing power and efficiency at relatively lower costs.

However, there are also drawbacks to consider. The high overall component density within racks leads to increased power usage, requiring additional cooling systems. Consequently, this results in higher power consumption. Furthermore, maintaining multiple devices within racks becomes considerably challenging as the number of racks increases, making maintenance a complex task.

2.3.2 Towers

A tower server closely resembles a traditional tower PC in appearance and functionality. Its advantages include scalability and ease of upgrade, allowing for customization and upgrades as needed. Tower servers are also cost-effective, often being the cheapest option among server types. Additionally, due to their low overall component density, they cool down easily.

However, tower servers have their drawbacks. They consume a significant amount of physical space and can be difficult to manage. Furthermore, while they provide a basic level of performance suitable for small businesses with a limited number of clients, they may not meet the performance needs of larger enterprises. Additionally, complicated cable management is a challenge, as devices are not easily routed together within the tower server setup.

2.3.3 Blade servers

Blade servers represent the latest and most advanced type of servers available in the market today. They can be described as hybrid rack servers, with servers housed inside blade enclosures forming a blade system. The primary advantage of blade servers lies in their compact size, making them ideal for conserving space in data centers.

Advantages of blade servers include their small size and form-factor, requiring minimal physical space. They also simplify cabling tasks compared to tower and rack servers, as the cabling involved is significantly reduced. Additionally, blade servers offer centralized management, allowing all blades to be connected through a single interface, which facilitates easier

maintenance and monitoring. Furthermore, blade servers support load balancing, failover, and scalability through a uniform system with shared components, enabling simple addition or removal of servers.

However, blade servers do have drawbacks. Their initial configuration or setup can be expensive and may require significant effort, particularly in complex environments. Blade servers often entail vendor lock-in, as they typically require the use of manufacturer-specific blades and enclosures, limiting flexibility and potentially increasing long-term costs. Moreover, due to their high component density and powerful nature, blade servers require special accommodations to prevent overheating, necessitating well-managed heating, ventilation, and air conditioning systems in data centers hosting blade servers.

2.3.4 Building structure

The IT equipment is housed in corridors and arranged within racks. It's important to note that server racks are never positioned back-to-back. The corridors where servers are situated are divided into cold aisles, providing access to the front panels of the equipment, and warm aisles, where the back connections are located. Cold air flows from the front (cool aisle), cooling down the equipment, and exits the room through the back (warm aisle).

2.4 Hardware accelerators

The rate of complexity in technology doubles approximately every 3.5 months, contrasting with Moore's Law, which historically predicted a doubling of computing capacity every 18-24 months. However, in the present era, Moore's Law has slowed down, with computing capacity doubling every 4 years or longer.

The emergence and widespread adoption of deep learning models have ushered in a new era where specialized hardware plays a crucial role in powering a wide range of machine learning solutions. Since 2013, the computational requirements for AI training have been doubling every 3.5 months, significantly outpacing the traditional Moore's Law projection of 18-24 months. To meet the escalating computational demands of deep learning tasks, WSCs are deploying specialized accelerator hardware such as GPUs, TPUs, and FPGAs. These accelerators are optimized to handle the intensive processing required for training and inference tasks in deep learning applications.

2.4.1 Graphical processing unit

GPUs have revolutionized data processing by enabling data-parallel computations, where the same program can be executed simultaneously on numerous data elements in parallel. This parallelization technique is particularly effective for scientific codes, which are often structured around matrix operations.

Harnessing the power of GPUs typically involves using high-level languages like CUDA and OpenCL. Compared to traditional CPU-based processing, GPUs can deliver remarkable speed boosts, sometimes performing computations up to 1000 times faster.

2.4.2 Neural networks

Neural networks are a computational model inspired by the human brain, specifically the perceptron. They comprise interconnected nodes, or neurons, organized in layers to process and

analyze data. Neural networks are utilized to learn data representation, enabling them to learn features and function as classifiers or regressors.

Neural networks have a rich history dating back to the 1940s, with notable developments occurring in the 1980s. In recent years, there has been a resurgence of interest in neural networks due to factors such as increased data availability and computational power, with some regarding them as among the top breakthroughs in 2013.

In natural neurons, information is transmitted through chemical mechanisms. Dendrites gather charges from synapses, which can be either inhibitory or excitatory. Once a threshold is reached, the accumulated charge is released, causing the neuron to fire. Like its biological counterpart, an artificial neuron receives input signals, which are weighted and summed. This sum undergoes an activation function, determining the output signal of the neuron:

$$h_j(x|w,b) = h_j\left(\sum_{i=1}^{I} w_j x_i - b\right) = h_j\left(\sum_{i=0}^{I} w_i x_i\right) = h_j\left(w^T x\right)$$

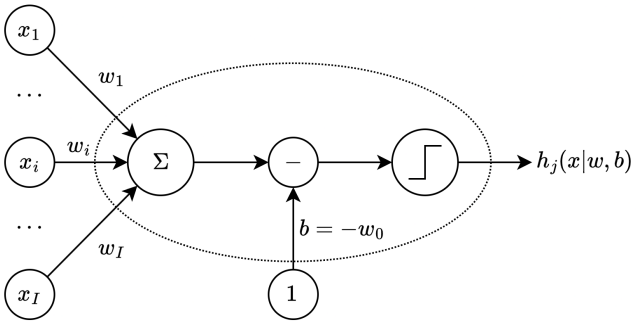


Figure 2.6: Artificial neuron

Neural networks are structured into at least three layers:

- *Input layer*: this is where data is initially introduced into the network.
- *Hidden layers*: intermediate layers that process and transform the input data.
- *Output layers*: the final layer that provides the network's ultimate results or predictions.

These layers are interconnected through weighted connections. Activation functions, which must be differentiable, determine the output of each neuron. Importantly, the output of a neuron is influenced solely by the inputs from the preceding layer.

Neural networks are inherently non-linear models. Their behavior is characterized by various factors, including the number of neurons, the choice of activation functions, and the specific values of the weights assigned to connections within the network. These factors collectively determine the network's ability to learn and make accurate predictions from the input data.

Learning in neural networks involves several key processes:

- *Activation functions*: neurons within the network make decisions based on input data through activation functions.
- *Weights*: connections between neurons are adjusted during training, either strengthened or weakened, to optimize performance.

Neural networks learn from historical data and examples provided during training. This process often involves backpropagation, which utilizes techniques like gradient descent and the chain rule to calculate errors and adjust the model accordingly. Errors are evaluated and used to refine the network's parameters, improving its ability to make accurate predictions or classifications.

Neural networks come in various types, each tailored for specific tasks:

- *Feed forward neural network*: standard neural network where information flows in one direction, from input to output.
- *Convolutional neural networks* (CNNs): designed for processing grid-like data such as images, with specialized layers for feature extraction and pattern recognition.
- *Recurrent neural networks* (RNNs): suitable for sequential data where past information influences the present, often used in natural language processing and time series analysis.

These types of neural networks find applications across diverse domains:

- *Image recognition*: utilized in technologies like FaceNet and YOLO for tasks such as facial recognition, object detection, and instance segmentation.
- *Natural language processing*: leveraged by models like BERT and GPT for applications like chatbots, sentiment analysis, and speech-to-text translation.

The potential for innovation and future development in neural networks is extensive. With the rapid growth of neural network applications, they are continuously expanding into new and diverse areas such as social media, aerospace, e-commerce, finances, and beyond. Additionally, advancements in generative AI are driving innovation in fields like art, music, and content generation, allowing for the creation of novel content through sophisticated generative models.

Neural networks and GPUs GPUs are commonly employed for training neural networks. However, the performance of such a synchronous system is constrained by the slowest learner and the slowest messages transmitted through the network. As the communication phase is a critical component, a high-performance network is essential for expediting parameter reconciliation across learners.

In configurations involving GPUs, a CPU host is typically connected to a PCIe-attached accelerator tray housing multiple GPUs. Within this tray, GPUs are interconnected using high-bandwidth interfaces such as NVlink, facilitating efficient communication and data exchange between the GPUs.

In the A100 GPU, each NVLink lane supports a data rate of $50 \times 4\, Gb/s$ in each direction. The total number of NVLink lanes increases from six lanes in the V100 GPU to 12 lanes in the A100 GPU, resulting in a total bandwidth of $600\, GB/s$. With the H100 GPU, each GPU can have up to 18 lanes, leading to a total bandwidth of $900\, GB/s$.

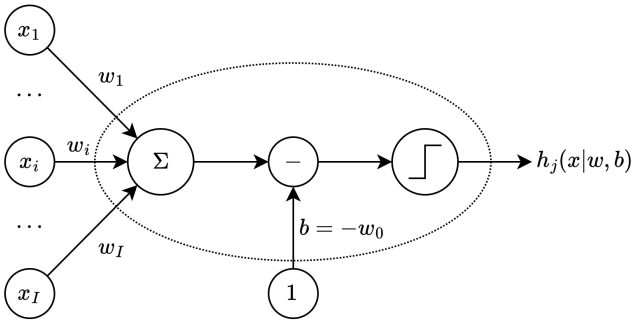


Figure 2.7: Neural network training

2.4.3 Tensor processing unit

Although GPUs are well-suited to machine learning (ML), they are still considered relatively general-purpose devices. However, in recent years, designers have increasingly specialized them into ML-specific hardware. These custom-built integrated circuits are developed specifically for machine learning tasks and are tailored to frameworks like TensorFlow.

These ML-specific hardware units have been powering Google data centers since 2015, alongside CPUs and GPUs. In TensorFlow, the basic unit of operation is a Tensor, which is an n-dimensional matrix.

Tensor Processing Units (TPUs) are extensively used for both training and inference tasks. The main versions of Tensor Processing Units are:

- *TPUv1*: inference-focused accelerator connected to the host CPU through PCIe links.
- *TPUv2*: supports both training and inference operations, providing enhanced flexibility and performance. In TPUv2, each Tensor Processing Unit (TPU) core consists of an array for matrix computations unit (MXU) and a connection to high bandwidth memory (HBM), which is used to store parameters and intermediate values during computation. Specifically, TPUv2 is equipped with 8 GiB of HBM for each TPU core, ensuring ample storage capacity. Additionally, there is one MXU allocated for each TPU core, facilitating efficient matrix computations. The hardware layout of TPUv2 comprises 4 chips, with each chip housing 2 cores. This configuration optimizes performance and scalability for machine learning tasks. Within a rack, numerous TPUv2 accelerator boards are interconnected via a custom high-bandwidth network, facilitating a collective ML compute power of 11.5 petaflops. This network's high bandwidth capabilities ensure swift parameter reconciliation with precisely controlled tail latencies. A TPU Pod, consisting of 64 units, can accommodate up to 512 total TPU cores and $4\,TB$ of total memory. This configuration maximizes both processing power and memory capacity for demanding machine learning tasks.
- *TPUv3*: Google's initial venture into liquid-cooled accelerators within their data centers. With a performance boost of 2.5 times compared to its predecessor, TPUv2, this supercomputing-class computational power enables various new ML capabilities such as AutoML and rapid neural architecture search. The TPUv3 Pod offers an impressive maximum configuration, boasting 256 devices, totaling 2048 TPU v3 cores. This setup delivers an astounding 100 petaflops of computing power and accommodates 32 TB of TPU memory, ensuring enhanced performance and scalability for demanding machine learning tasks.

- *TPUv4*: announced in June 2021, TPUv4 marks the latest advancement in Google's Tensor Processing Unit lineup. Initially deployed to bolster Google's in-house services, TPUv4 is not yet available as a cloud service. A single TPUv4 pod comprises 4096 devices, showcasing a remarkable performance increase of approximately 2.7 times compared to its predecessor, TPUv3. With computing capabilities equivalent to around 10 million laptops, TPUv4 sets a new standard for high-performance machine learning hardware.
- *TPUv5*: it has two versions (v5e and v5p). The latter one was announced in December 2023 and has been available since early this year. Compared to TPUv4, v5p can train large language models like GPT3-175B 2.8 times faster. On the other hand, although v5e is slower, it offers more relative performance per dollar compared to v5p.

On the other hand, AWS offers Trainium2, specifically optimized for Large Language Models (LLMs). Additionally, they provide Graviton4, based on ARM architecture, offering 30% greater energy efficiency for general AI tasks. For AI inference, AWS offers Inferentia2, which boasts 2.3 times higher throughput and up to 70% lower cost per inference compared to comparable EC2 instances.

2.4.4 Field programmable gate array

FPGAs (Field-Programmable Gate Arrays) are arrays of logic gates that users can program or configure in the field, without relying on the original designers. These devices consist of carefully designed and interconnected digital subcircuits that efficiently implement common functions, providing extremely high levels of flexibility. The individual digital subcircuits within FPGAs are known as configurable logic blocks (CLBs).

VHDL and Verilog are hardware description languages used to describe hardware. They enable users to create textual representations of hardware components and their interconnections. HDL code resembles a schematic, using text to define components and establish connections between them.

	Advantages	**Disadvantages**
CPU	Easily programmable and compatible Rapid design space Efficiency	Simple models Small training sets
GPU	Parallel execution	Limited flexibility
TPU	Fast for ML	Limited flexibility
FPGA	High performance Low cost Low power consumption	Limited flexibility High-level synthesis