

## **preface**



# Inhoudsopgave

<b>1 Inleiding</b>	<b>5</b>
<b>2 Literatuurstudie</b>	<b>7</b>
2.1 Machine Learning (ML) . . . . .	7
2.1.1 Leermodellen . . . . .	7
2.1.2 Leertechnieken . . . . .	12
2.2 Evolutie Single Board Computers (SBC) . . . . .	14
2.2.1 Geschiedenis . . . . .	14
2.3 Benchmarking van ML algoritmes . . . . .	16
2.4 Assortiment aan 'on the shelf' toestellen . . . . .	17
2.4.1 Beaglebone AI . . . . .	17
2.4.2 Coral Dev . . . . .	17
2.4.3 Nvidia Jetson Nano . . . . .	17
2.4.4 Nvidia Jetson T2 . . . . .	18
2.4.5 Raspberry Pi . . . . .	18
2.4.6 ... . . . .	18
<b>3 This is the another Chapter</b>	<b>19</b>
3.1 This is a Section Heading . . . . .	19
<b>A Een aanhangsel</b>	<b>23</b>
<b>B Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel</b>	<b>25</b>
<b>C Poster</b>	<b>27</b>



## **Hoofdstuk 1**

### **Inleiding**



## Hoofdstuk 2

# Literatuurstudie

### 2.1 ML

ML is een onderdeel van Artificiële Intelligentie (AI) en staat voor het wetenschappelijke onderzoek naar algoritmes en statistische modellen dat gebruikt kan worden door verschillende computersystemen. Deze systemen zijn hierdoor in staat om specifieke taken te voltooien zonder rechtstreekse instructies of regels mee gekregen te hebben. Ze steunen in de plek op onder meer patroonherkenning om de kans op het succesvol uit te voeren van taken te maximaliseren. Hiervoor wordt er een wiskundig model gebouwd op basis van training-data. Deze mathematische modellen en *datahandling* kan op verscheidene manieren gebeuren. Hieronder bespreken we een aantal gebruikelijke modellen. We maken ook een afweging welk model het interessantst is voor onze toepassing.

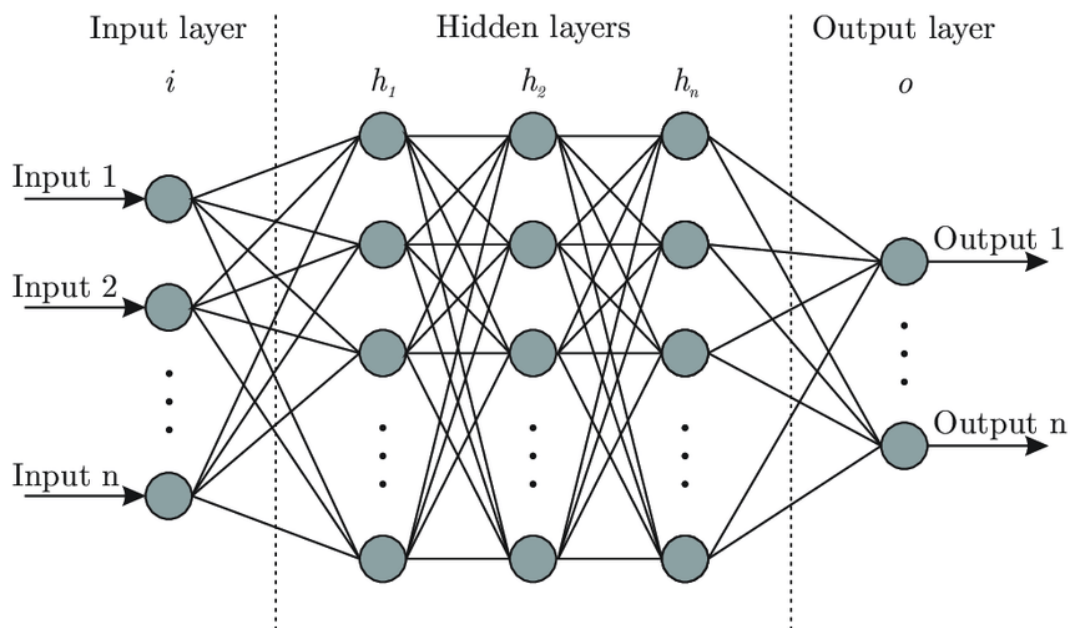
#### 2.1.1 Leermodellen

Om ML technieken toe te passen moet men gebruik maken van een bepaald model. Een model dat is toegepast op trainingsdata en nieuwe data kan verwerken om voorspellingen te maken. Er bestaat een hele waaier aan mogelijke modellen. In de volgende paragrafen bespreken we een aantal opties waarna we de verschillende modellen afwegen tegenover elkaar.

#### Artificiële Neurale Netwerken

Artificiële Neurale Netwerken (ANN) is een term dat gebruikt wordt om een algoritme te omschrijven dat lijkt op maar niet hetzelfde is als een biologisch neuraal netwerk in de hersens van dieren. Deze systemen zijn in staat om een bepaalde taak te leren, en zichzelf te verbeteren. In de meeste gevallen worden er zelf geen richtlijnen of omschrijving meegegeven. Het systeem ontdekt zelf hoe deze regels in elkaar zitten. Een bekend voorbeeld is het herkennen van de cijfers 0 tot 9. Hierin wordt er niet aan het systeem verteld dat het cijfer 8 uit twee cirkels bestaat die verticaal tegen elkaar aansluiten. Het algoritme zal dit gaandeweg ontdekken met behulp van de vele voorbeelden waar het gebruik van kan maken. Met behulp van veel data kan een algoritme zichzelf verfijnen en nauwkeuriger bepaalde cijfers herkennen.

**Structuur van een Neurale Netwerken (NN)** Een ANN is verzameling van nodes die met elkaar verbonden zijn zoals neuronen in de hersens van een mens. Hierbij kan elke neuron een signaal



**Figuur 2.1:** Structuur van een Neuraal Netwerk

doorgeven naar de volgende neuron waar het signaal verwerkt kan worden en weer doorgegeven kan worden. Hetzelfde principe geldt ook bij NN met het verschil dat er meerdere lagen van nodes te onderscheiden zijn.

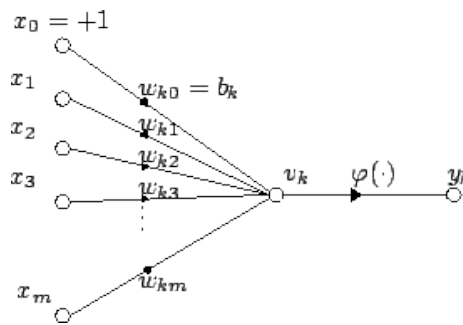
**Lagen** Er zijn drie soorten lagen te onderscheiden: een Input Layer, Hidden Layers en een Output Layer. Elke laag is verbonden met de volgende laag door middel van de connecties of *edges* tussen de verschillende nodes. In figuur 2.1 is de algemene vorm te vinden van een NN.

- **Input Layer:** De eerste laag van elk NN is de Input Layer. Deze bestaat uit een aantal inputnodes. Elke inputnode krijgt de ruwe data binnen waar er een operatie op uitgevoerd wordt en vervolgens bepaalde parameters doorgeeft aan de volgende laag.
- **Hidden Layers:** Na de inputlaag komen een aantal Hidden Layers. Het aantal Hidden Layers en de hoeveelheid nodes binnen één Hidden Layer kan variëren van applicatie tot applicatie en is sterk gerelateerd aan de complexiteit van de toepassing.
- **Output Layer:** Na de Hidden Layers is de laatste laag de Output Layer. Hier worden de laatste operaties uitgevoerd en worden de eindwaarden verkregen.

**Nodes** Een node is op zich gebaseerd op zijn biologisch tegenbeeld. Het krijgt een bepaald aantal inputs, verwerkt deze en geeft een bepaald aantal outputs. Deze inputs en outputs worden van node naar node doorgegeven via verbindingen. Elke node heeft met elke node in de volgende laag een connectie. Deze worden *edges* genoemd. Elke edge draagt een bepaald gewicht. Via dit gewicht kan de invloed van de huidige node versterken of verzwakken in de volgende node.

**Propagation function** De mathematische functie die een node gebruikt voor het verwerken van inputs naar outputs heet de propagatie functie. In figuur 2.2 bespreken we de algemene vorm





**Figuur 2.2:** Algemene structuur van een node.

van een neuron. Voor een bepaalde neuron met  $m+1$  inputs ( $x_0$  t.e.m.  $x_m$ ) en bijhorende gewichten ( $w_0$  t.e.m.  $w_m$ ). Gebruikelijk wordt  $x_0 = +1$  genomen. Hierdoor blijven er maar  $m$  echte inputs over waardoor er voor een bepaalde output volgende functie opgesteld kan worden. Hierbij is  $\phi$  een van de mogelijke transfer functies die verder besproken zal worden.

$$y_k = \phi \left( \sum_{j=0}^m w_{kj} x_j \right) \quad (2.1)$$

**types transfer functies** De transfer functie of activatiefunctie van een neuron bevat bepaalde eigenschappen die het volledige netwerk kan verbeteren of versimpelen. Hieronder bespreken we enkele transfer functies.

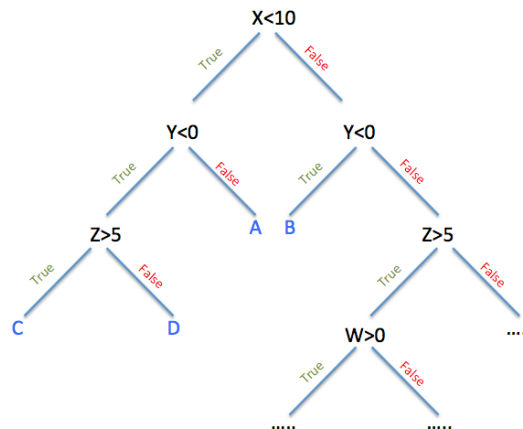
- **Stapfunctie:** Hier wordt er gekeken naar de verkregen waarde van de gewogen som  $u$  van  $m+1$  inputs. Bedraagt deze waarde minder dan een bepaalde drempel  $\theta$ , dan wordt de output gelijkgesteld aan nul, bij een hogere waarde dan weer aan 1. Dit is te zien in formule 2.3. Dit type wordt vooral gebruikt om binaire inputs te verzorgen bij de volgende laag.

$$u = \left( \sum_{j=0}^m w_{kj} x_j \right) \quad (2.2)$$

$$y = \begin{cases} 1 & \text{als } u \geq \theta \\ 0 & \text{als } u < \theta \end{cases} \quad (2.3)$$

- **Lineaire Combinaties:** In dit geval is de output niets minder dan de gewogen som plus  $e$
- **Sigmoid:**
- **Rectifier:**

**Voor- en nadelen**



**Figuur 2.3:** Algemene structuur van een beslissingsboom.

## Beslissingsboom

Het gebruik van een beslissingsboom is een leermethode die met regelmaat terugkomt in de statistiek als een voorspellend model. Men maakt gebruik van observaties rond een bepaald onderwerp om een beslissing te nemen rond de aard van het onderwerp. Deze beslissing leidt naar een variabele outputwaarde. Indien deze waarde valt onder te verdelen in discrete klassen spreekt men over een *classificatie boom*. Neemt de gezochte variabele eerder een continue vorm aan, dan maakt men gebruik van *regressie bomen*.

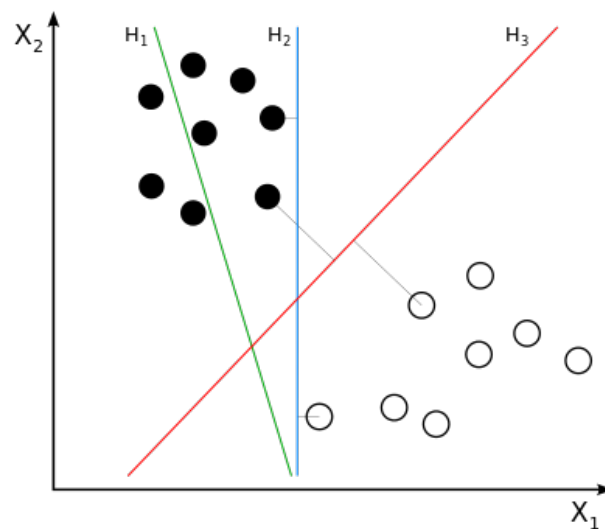
**Structuur van een beslissingsboom** Net zoals bij de NN bestaat een beslissingsboom uit verschillende lagen en nodes. Er zijn echter wel belangrijke verschillen in het gebruik bij NN en een beslissingsboom. Om te beginnen worden lagen horizontaal weergegeven t.o.v. verticaal bij Neurale Netwerken. Zoals te zien is in figuur 2.3 word er in elke laag een onderscheid gemaakt op basis van een bepaalde statement of parameter. Deze parameter kan een enkele inputwaarde zijn of een lineaire combinatie van meerdere inputwaarden.

Bij de nodes kan er onderscheid gemaakt worden tussen een gewone node en een eindnode. Bij elke gewone node wordt een bepaalde statement geverifiëerd en wordt er naar een node overgegaan in de volgende laag op basis van die statement. Bij een eindnode is het niet meer mogelijk om door te gaan naar een volgende laag, maar wordt er een outputwaarde gegeven.

## Voor- en nadelen

### Support Vector Machines

Support Vector Machines (SVMs) of ook wel gekend als support vector networks is een model dat veelvuldig gebruikt wordt in classificatie en regressie-analyse. SVMs verdeelt objecten onder in twee verschillende klassen aan de hand van een aantal kenmerken. Het is dus een binaire classificeerder. Om aan de hand van kenmerken een onderverdeling te maken moeten deze kenmerken eerst omgezet worden in een numeriek model. De data wordt hervormd tot een vectorruimte. In de trainingsfase wordt er getracht een zo optimaal mogelijke scheiding tussen



**Figuur 2.4:** Twee dimensionale Support Vector Machine.

beide klassen te vinden. Deze optimale scheiding wordt ook wel een *hypervlak* genoemd en ligt op zo groot mogelijke afstand tussen de dichtstbijgelegen objecten van beide klassen of support vectors. In figuur 2.4 kan u een twee dimensionaal voorbeeld vinden. Hierin is scheidingslijn  $H_1$  geen acceptabele scheiding doordat er objecten van de zwarte klasse fout geclassificeerd worden.  $H_2$  is acceptabel maar is nog niet optimaal aangezien er weinig foutmarge is voor een nieuw object.  $H_3$  is het hypervlak doordat de foutmarge tussen de twee klassen zo groot mogelijk is. Deze methode is niet alleen mogelijk in toepassingen met een lineaire scheiding. Ook in niet-lineaire gevallen kan men een transformatie uitvoeren om toch een lineaire scheiding te bekomen. Deze hervorming wordt ook wel de *kernel trick* genoemd.

### Voor- en nadelen

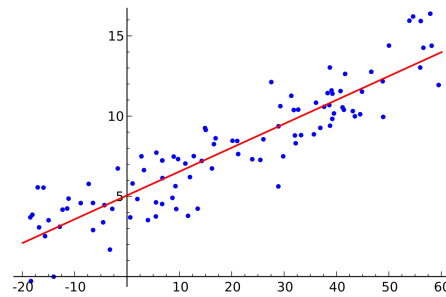
### Regressie Analyse

Regressie Analyse is een techniek uit de statistiek, die gebruikt wordt om gegevens te analyseren met een specifieke verband. Er bestaat vaak een relatie tussen een afhankelijke variabele en een (of meerdere) onafhankelijke variabelen. De meest voorkomende vorm van regressie analyse is de lineaire regressie, waar men op zoek gaat naar de functie die het dichtst aanleunt bij de data en dit volgens specifieke criteria zoals het voldoen aan een bepaalde graad. Regressie analyse wordt vooral gebruikt voor het voorspellen van nieuwe data of gebeurtenissen. In figuur 2.5 kan je een voorbeeld van lineaire regressie vinden.

### Voor- en nadelen

### Bayesian Netwerken

Bayesian Netwerken, ook wel probabilistische netwerken, is een structuur waarin data op probabilistische wijze geanalyseerd kan worden. Men maakt gebruik van gerichte grafen. Hierin bestaan de knopen



**Figuur 2.5:** Voorbeeld van Lineaire Regressie.

uit variabelen en de arcs beschrijven de conditionele afhankelijkheden tussen de verschillende knopen. Bayesian Netwerken worden vooral gebruikt om te analyseren wat de bepalende oorzaak is voor een zekere gebeurtenis.

### Genetisch Algoritmes

Genetische Algoritmes zijn een heuristisch geïnspireerd op het principe van natuurlijke selectie en zijn een klasse binnen evolutionaire algoritmes. Dit type algoritme kan gebruikt worden om oplossingen te vinden in optimalisatie- en zoekproblemen. Door te steunen op biologische principes zoals mutatie, selectie en kruisbestuiving worden er nieuwe *chromosomen* gegenereerd die mogelijks een betere oplossing geven voor een bepaald probleem.

#### 2.1.2 Leertechnieken

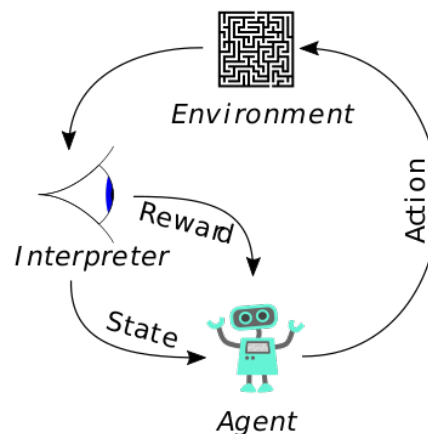
Om een neurale netwerk te laten leren zijn er verschillende parcours om te bewandelen de drie belangrijkste methodes om een mathematische functie te verkrijgen worden hieronder opgesomd.

##### Supervised Learning:

Deze techniek maakt gebruik van gepaarde datasets van inputobjecten en de te verwachten outputobjecten. Het doel is om een mathematische functie te creëren waarbij de gegenereerde outputs zo nauw mogelijk overeenkomen met de gelabelde outputs uit de datasets. Men optimaliseert deze mathematische functie door iteratief te trainen. De bijgeschaafde functie kan dan ook gebruikt worden voor nieuwe datasets zonder gelabelde output. Een toepassing van Supervised Learning is bijvoorbeeld het detecteren van spam met een trainingset van al gelabelde e-mails.

##### Unsupervised Learning:

Unsupervised learning is een techniek dat gebruik maakt van Hebbian Learning om onbekende patronen te herkennen in datasets. De twee meestgebruikte methodes onder Unsupervised Learning zijn principal component en cluster analysis. Principal component maakt gebruik van orthogonale transformaties om een set van mogelijke afhankelijke variabelen om te zetten in een set van lineaire onafhankelijke variabelen. In cluster analyse wordt er getracht om een groep objecten te identificeren en verdelen in een cluster van gelijkaardige objecten. Een belangrijke toepassing

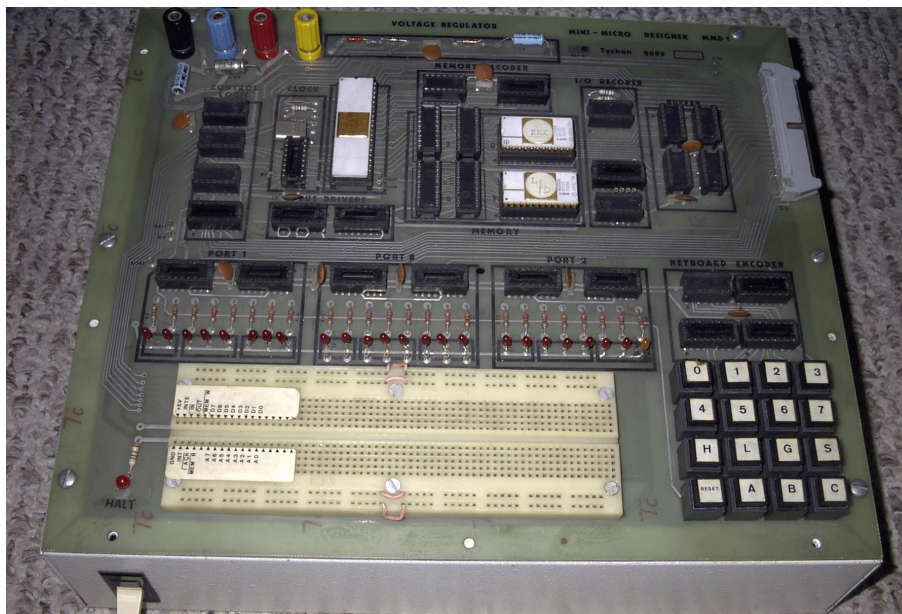


**Figuur 2.6:** Routine bij Reinforcement Learning

van Unsupervised Learning is het clusteren van gelijkaardige documenten op basis van de inhoud van de tekst.

### **Reinforcement Learning:**

Deze leertechniek heeft betrekking tot hoe agents acties moeten ondernemen in een omgeving om een bepaald attribuut te maximaliseren. Het verschilt van Supervised en Unsupervised Learning door de onafhankelijkheid van gelabelde outputdatasets en dat minder optimale acties niet manueel gecorrigeerd worden. De techniek heeft als doel om een evenwicht te vinden tussen exploratie van ongekend gebied en exploitatie van de huidige kennis. In figuur 2.6 kan je een eenvoudige routine vinden van Reinforcement Learning-algoritme. Hierbij maakt een agent een bepaalde actie gebaseerd op de staat waar hij in is. Deze actie heeft in een omgeving een zekere invloed die door een Interpreter beoordeeld wordt en een score toekent. De agent kan deze verandering daarna gebruiken om zichzelf te verbeteren en zijn acties aanpassen. Een van de vele mogelijke toepassingen van Reinforcement Learning is het aanleren van schaken door enkel mee te geven of het algoritme gewonnen of verloren heeft.



**Figuur 2.7:** Eerste SBC: MMD-1

## 2.2 Evolutie SBC

Een SBC is een volledige computer gemaakt op 1 enkele printplaat. Het bevat onderdelen zoals een microprocessor, geheugen, inputs en outputs. De SBC werd ontwikkeld als een voorstel-hulpmiddel bij educatieve doelstellingen of het gebruik als een embedded computer controller. Tegenwoordig zijn ook vele (draagbare) computers geïntegreerd op één printplaat. Het grote verschil met (draagbare) computers is dat er geen nood is aan expansion slots zoals bijvoorbeeld voor RAM-geheugen of een Graphics Processing Unit (GPU).

### 2.2.1 Geschiedenis

De eerste echte SBC was de zogenaamde "dyna-micro" uit figuur 2.7 die later de naam "MMD-1" (Mini-Micro Designer 1) kreeg. Dit toestel werd uitgegeven in 1976 en werd populair doordat het werd gepresenteerd in het destijds 'BugBook' als het voorbeeld microprocessor. Een andere vroege SBC was de KIM-1 (Keyboard Input Monitor 1) uit hetzelfde jaar. Beide machines werden voor ingenieurs geproduceerd en ontworpen maar vonden een breed publiek onder de hobbyisten waar het heel populair was. Later kwamen nog andere namen zoals de Ferguson Big Board en de Nascom.

Naarmate dat de markt voor desktops en PC's groeide, nam de belangstelling voor SBC in computers meer en meer af. Er werd verschoven naar een moederbord met de belangrijkste componenten en dochterborden voor periferiecomponenten zoals seriële poorten. De voornaamste reden hiervoor was dat de componenten groot waren. Alle onderdelen op dezelfde printplaat zou zorgen voor een onpraktisch ontwerp met grote afmetingen. Deze beweging was echter tijdelijk en naarmate de vorderende technologie kleinere componenten kon leveren, werden onderdelen terug naar het mainframe verschoven. Tegenwoordig kunnen de meeste moederborden terug als SBC beschouwd worden.

In het jaar 2004 werd er in Italië een nieuwe microcontroller uitgebracht onder de naam "Arduino". Dit ontwerp had naast het voordeel van compact en goedkoop te zijn, ook nog eenvoudigheid mee.

Door de eenvoud werd het Arduino-platform snel populair onder techneuten van alle soorten. Twee jaar later kwam er uit de Universiteit van Cambridge ook het nieuws van een nieuwe goedkope SBC uit. De bekende Raspberry Pi werd gelanceerd voor de prijs van \$35. Het hoofddoel van dit project was een nieuw leermiddel om te programmeren maar werd door het grote aantal applicaties ook zeer populair.

De laatste jaren kende een grote explosie aan nieuwe SBC. Een hele reeks nieuwe namen verscheen. Banana Pi, Beaglebone, Intel Galileo, Google Coral Dev en Asus Tinker Board zijn maar enkele van de vele voorbeelden. Deze toestellen hebben vaak een processor gebaseerd op de x86- of ARM-series en maken gebruik van een Linux besturingssysteem zoals Debian.

## 2.3 Benchmarking van ML algoritmes



## 2.4 Assortiment aan 'on the shelf' toestellen

### 2.4.1 Beaglebone AI

BeagleBone Ai (BB AI) is een SBC dat verder bouwt op de succesvolle BeagleBoard-series. Het is een open source project met een op Linux gebaseerd aanpak. De BB AI probeert het gat tussen kleinere SBC en krachtigere industriële computers te overbruggen. Met dank aan de krachtige Texas Instruments AM5729 CPU kunnen ontwikkelaars de krachtige System On Chip (SoC) gebruiken van een hele brede waaier aan toepassingen op het AI terrein. De BB AI maakt het makkelijk om dat terrein te ontdekken en verkennen. Door gebruik te maken van onder andere embedded-vision-engine (EVE) cores die steunen op een geoptimaliseerde TIDL machine learning OpenCL API met gebruiksklare hulpmiddelen kan je terecht in alledaagse automatisatie in industriële, commerciële en thuisapplicaties.

#### Specifications

- **GPU:**
- **CPU:** Texas Instruments AM5729
- **Memory:** 16 GB on-board eMMC flash
- **Storage:** 1GB RAM + micro SD-slot
- **Power:**

### 2.4.2 Coral Dev

#### Specifications

- **GPU:** Integrated GC7000 Lite Graphics
- **CPU:** NXP i.MX 8M SOC (quad Cortex-A53, Cortex-M4F) + coprocessor Google Edge TPU
- **Memory:** 8 GB on-board eMMC flash
- **Storage:** 1GB RAM LPDDR4 + micro SD-slot
- **Power:**

### 2.4.3 Nvidia Jetson Nano

#### Specifications

- **GPU:** 128-core Maxwell
- **CPU:** Quad-core ARM A57 @ 1.43 GHz
- **Memory:** 4 GB 64-bit LPDDR4 25.6 GB/s
- **Storage:** micro SD-slot
- **Power:** 5 W

### 2.4.4 Nvidia Jetson T2

#### Specifications

- **GPU:** 256-core NVIDIA Pascal GPU architecture with 256 NVIDIA CUDA cores
- **CPU:** Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
- **Memory:** 8GB 128-bit LPDDR4 Memory 1866 MHz - 59.7 GB/s
- **Storage:** 32GB eMMC 5.1
- **Power:** 7,5 - 15 W

### 2.4.5 Raspberry Pi

#### Specifications

- **GPU:**
- **CPU:** Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- **Memory:** 1GB, 2GB or 4GB LPDDR4-3200 SDRAM
- **Storage:** micro SD-kaart
- **Power:** 2,8 - 5,2 W

### 2.4.6 ...

## Hoofdstuk 3

# This is the another Chapter

You can say great work has been done about something [? ? ] or say that ? ] did something really great. xxxx xxxxx xxxx xxxxxxxxxxx xxx xxxxx xxxxx xxx xxxx xxxx xxxxx xxxxx xxxx xxxxx xxxxx xxxxxxxxxxx

```

XXX XXXXX XXXXX XXX XXXX XXXX XXXXX XXXXX XXXX XXXXX XXXX XXXXXXXXXX XXX XXXXX XXXXX XXX XXXX
XXXX XXXXX XXXXX XXXX XXXXX XXXX XXXXXXXXXX XXX XXXXX XXXXX XXX XXXX XXXX XXXXX XXXXX XXXX XXXXX
XXXX XXXXXXXXXX

```

XXX XXXXX XXXXX XXX XXXX XXXX XXXXX XXXX XXXXX XXXX XXXXXXXXXXX XXX XXXXX XXXXX XXX XXXX  
 XXXX XXXXX XXXXX XXXX XXXXX XXXX XXXXXXXXXXXX

### 3.1 This is a Section Heading

**Figuur 3.1:** This is the Caption for Figure 1

Here's an example  
of a table  
floated with the  
table environment command.

**Tabel 3.1:** This is the Caption for Table 1

## **Bibliografie**



## **Bijlage A**

# **Een aanhangsel**

sdfsffqsfsf





## **Bijlage B**

# **Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel**



**Bijlage C**

**Poster**

FACULTEIT INDUSTRIELE INGENIEURSWETENSCHAPPEN  
TECHNOLOGIECAMPUS GENT  
Gebroeders De Smetstraat 1  
9000 GENT, België  
tel. + 32 92 65 86 10  
fax + 32 92 25 62 69  
iiw.gent@kuleuven.be  
www.iw.kuleuven.be



LID VAN **ASSOCIATIE  
KU LEUVEN**