

Accelerated edge computing with Neural Networks

Arno PLAETINCK

Promotor: Prof. Dr. Ir. Nobby Stevens

Co-promotoren: Ing. Willem Raes
Ing. Jorik De Bruycker

Masterproef ingediend tot het behalen van
de graad van master of Science in de
industriële wetenschappen: Industriële
Ingenieurswetenschappen Elektronica-ICT
Embedded Systems

©Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologicampus Gent, Gebroeders De Smetstraat 1, B-9000 Gent, +32 92 65 86 10 of via e-mail iiw.gent@kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Dankwoord

Dank aan mezelf

Dank aan een ander

Abstract

Testabstract. Omschrijf hier je thesis.

Trefwoorden: latex, thesis, stijl

Inhoudsopgave

1 Literatuurstudie	1
1.1 Kadering masterproef	2
1.2 Machine Learning (ML)	3
1.2.1 Leermodellen	3
1.2.2 Keuze voor een ML-methode	8
1.2.3 Leertechnieken	11
1.3 Evolutie Single Board Computers (SBC)	12
1.3.1 Geschiedenis	12
1.4 Assortiment aan 'on the shelf' toestellen	14
1.4.1 Beaglebone AI	14
1.4.2 Coral Dev Board	14
1.4.3 Nvidia Jetson Nano	15
1.4.4 Nvidia Jetson TX2	15
1.4.5 Raspberry Pi	16
1.5 Benchmarking van ML algoritmes	17
2 Inleiding	19
3 Bespreking	23
4 This is the another Chapter	25
4.1 This is a Section Heading	25
A Een aanhangsel	29
B Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel	31
C Poster	33

Lijst van figuren

1.1	Structuur van een Neuraal Netwerk	4
1.2	Algemene structuur van een node.	5
1.3	De Sigmoid activatiefunctie[1].	5
1.4	Algemene structuur van een beslissingsboom.[2]	6
1.5	Twee dimensionale Support Vector Machine.[3]	7
1.6	Voorbeeld van Lineaire Regressie.[4]	8
1.7	Routine bij Reinforcement Learning	12
1.8	Eerste SBC: MMD-1 [5]	13
4.1	This is the Caption for Figure 1	25

Lijst van tabellen

Lijst van symbolen en acroniemen

Symbols

λ_{ex}	Excitation wavelength
λ_{em}	Emission wavelength
$N(\dots)$	Noise process
NOG TE DOEN OP HET EINDE VAN DE THESIS	






Acronyms

ADC	Analog to Digital Converter
CCD	Charge-Coupled Device
NOG TE DOEN OP HET EINDE VAN DE THESIS	

Hoofdstuk 1



Literatuurstudie

In dit hoofdstuk zal er eerst een korte schets voorzien worden van het kader waarin de thesis zich voordoet. Hierna zal een kleine iding gegeven worden tot Artificiële Intelligentie: waar bestaat  uit en waar het voor kan dienen. We bekijken de verschillende leermodellen en vermelden hoe  onderdelen een werkend geheel vormen. Vervolgens maken we een overzicht van de voor- en nadelen van de verschillende technieken. We kiezen de best bare methode uit en staven deze keuze. We bespreken ook de geschiedenis van Single Board Computers. Van waar kwamen deze toestellen en hoe zijn ze geëvolueerd. Verder worden ook verschillende voorbeelden van SBCs besproken die in staat zijn om Machine Learning-technieken te draaien. Deze zullen onderworpen worden aan een benchmark die  in paragraaf 1.5 bespreken.

1.1 Kadering masterproef



~~De wereld vandaag is niet meer dezelfde als die van gisteren.~~ De computerwereld maakt de laatste jaren grote stappen op vlak van Machine Learning. Deze vorderingen werden gedreven door onder meer de nieuwste ontwikkelingen op vlak van computerrekenkracht en de vierde industriële revolutie. Door de veelvuldige toepassingsmogelijkheden werd ML een populair en veelbesproken onderwerp. Tegenwoordig kan een bepaalde vorm van Artificiële Intelligentie (AI) in elke sector teruggevonden worden. Van hartritmestoornissen in de medische sector tot commentaarherkenning in de toeristische tak.

Een branche in de ML die steeds meer licht op het podium krijgt, valt in de logistieke sector. Door de steeds verder doorgedreven automatisatie van bedrijven, wordt er ook in het magazijn gekeken voor het optimaliseren van het leveren van de verschillende onderdelen. Gebruik maken van zelfrijdende kheftrucks kan een grote stap vormen in het opknappen van de efficiëntie. Niet alleen in het magazijn maar ook op de weg is er een groeiende belangstelling naar zelfrijdende auto's, die ontwikkeld worden door grote bedrijven zoals Tesla en Uber. In beide cases zal er een zekere vorm van positiebepaling voorvallen. Het is van groot belang dat deze bepaling zo accuraat mogelijk gebeurt met niet alleen een juiste locatie, maar ook op zo kort mogelijke tijdsperiode. Deze nood aan *low latency* kan het verschil betekenen tussen een voertuig die beslist dat hij moet vertragen of beslist dat hij veilig kan doorrijden maar toch een botsing veroorzaakt. De berekening van de locatiebepaling kan zowel in de *cloud*, als in de *edge* gebeuren en gebeurt vooral met behulp van Machine Learning. Door de grote toegevoegde latency bij berekeningen in de cloud zal de keuze voor *edge computing* vallen. Welke hardware men gebruikt kan variëren van applicatie tot applicatie.

In deze thesis zal men trachten de edge-kant te belichten door een benchmark op te stellen. Dit instrument moet meer inzicht verschaffen in welke mate machineleertechnieken toepasbaar zijn. Hoe groot is de latency die op treedt? Welk verbruik en complexiteit van het netwerk gaat er hier mee gepaard? De benchmark zal toegepast worden op verschillende Single Board Computers. Ook tussen deze hardware-opties wordt er een afweging gemaakt. Welk toestel is voordeliger in welke situatie? Is een goedkoper toestel tot evenwaardige resultaten in staat?



1.2 ML



ML is een onderdeel van 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 onder meer op patroonherkenning om de kans op het succesvol uitvoeren van taken te maximaliseren. Hiervoor wordt er een wiskundig model gebouwd op basis van training-data. Dit mathematisch modelleren en *datahandling* kunnen op verscheidene manieren gebeuren. Hieronder bespreken we een aantal gebruikelijke modellen. We maken ook een afweging welk model het meest interessant is voor onze toepassing.

1.2.1 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.

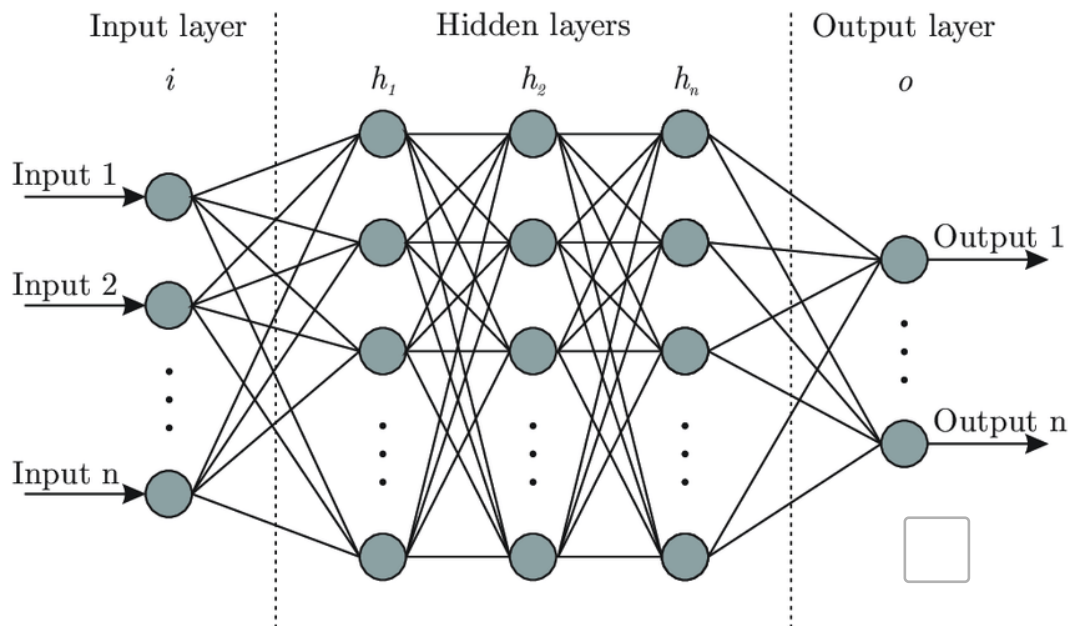
Artificiële Neurale Netwerken

Artificiële Neurale Netwerken (ANN) is een term die gebruikt wordt om een algoritme te omschrijven ~~ijkt op, maar niet hetzelfde is als een biologisch neuraal netwerk in de hersenen van dieren.~~ Deze systemen zijn in staat om een bepaalde taak te leren, en zichzelf te verbeteren. In de meeste gevallen worden er zelfs geen richtlijnen of een 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 Neurale Netwerken (NN) Een ANN is een verzameling van nodes die met elkaar verbonden ~~zijn zoals neuronen in de hersenen van een mens.~~ Hierbij kan elke neuron een signaal doorgeven naar het 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 connecties of *edges* tussen de verschillende nodes. In figuur 1.1 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.



Figuur 1.1: Structuur van een Neuraal Network

- **Output Layer:** Na de Hidden Layers is de laatste laag de Output Layer. Hier worden de laatste operaties uitgevoerd en worden de eindwaarden verkregen.

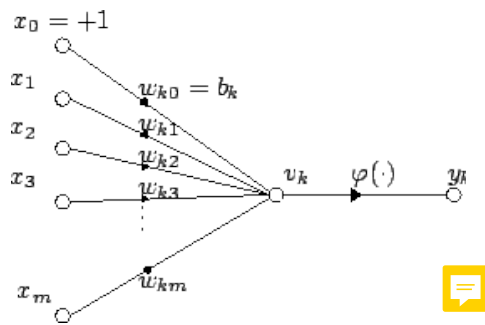
Nodes Een node is gebaseerd op zijn biologisch tegenbeeld. Het krijgt een bepaald aantal inputs, verwerkt deze en geeft een bepaald aantal outputs. Deze inputs en outputs worden van node naar node doorgegeven via verbindingen. Elke node heeft met elke node in de volgende laag een connectie. Deze worden *edges* genoemd. Elke edge draagt een bepaald gewicht. Via dit gewicht kan men 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 1.2 bespreken we de algemene vorm 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 ϕ een van de mogelijke transferfuncties die verder besproken zal worden.

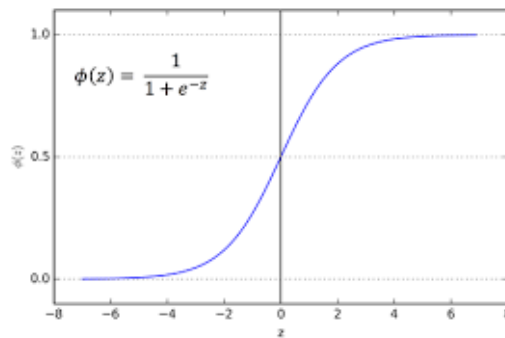
$$y_k = \phi \left(\sum_{j=0}^m w_{kj} x_j \right) \quad (1.1)$$

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

- **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 θ , dan wordt de output



Figuur 1.2: Algemene structuur van een node.



Figuur 1.3: De Sigmoid activatiefunctie[1].

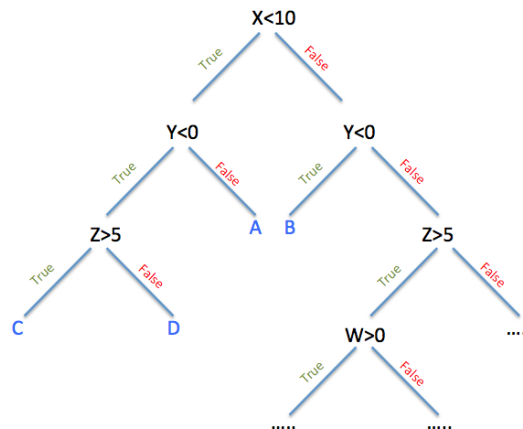
gelijkgesteld aan nul, bij een hogere waarde dan weer aan 1. Dit is te zien in formule 1.3. Dit type wordt vooral gebruikt om binaire inputs te verzorgen bij de volgende laag.

$$u = \left(\sum_{j=0}^m w_{kj} x_j \right) \quad (1.2)$$

$$y = \begin{cases} 1 & \text{als } u \geq \theta \\ 0 & \text{als } u < \theta \end{cases} \quad (1.3)$$

- **Lineaire Combinaties:** In dit geval is de output niets minder dan de gewogen som vermenigvuldigd met een constante waarbij een tweede constante wordt opgeteld.
- **Sigmoid:** De Sigmoid functie is een mathematische functie die lineaire inputs omzet naar niet-lineaire outputs. Het heeft de karakteristieke 'S'-vorm zoals te zien is in 1.3. Door deze activatiefunctie worden inputs omgezet in een waarde tussen 0 en 1 (of -1 en 1, afhankelijk van de conventie).
- **Rectifier:** De rectifier als activatiefunctie is een functie die enkel het positieve deel van zijn argument doorlaat. In vergelijking 1.4 vind je de functie weer waar x de input is van de neuron. Deze is een vector aan waarden die zowel positief als negatief kunnen zijn. Deze functie is ook gekend onder de naam *rectified linear unit (ReLU)*.

$$f(x) = x^+ = \max(0, x) \quad (1.4)$$



Figuur 1.4: Algemene structuur van een beslissingsboom.[2]

Beslissingsboom

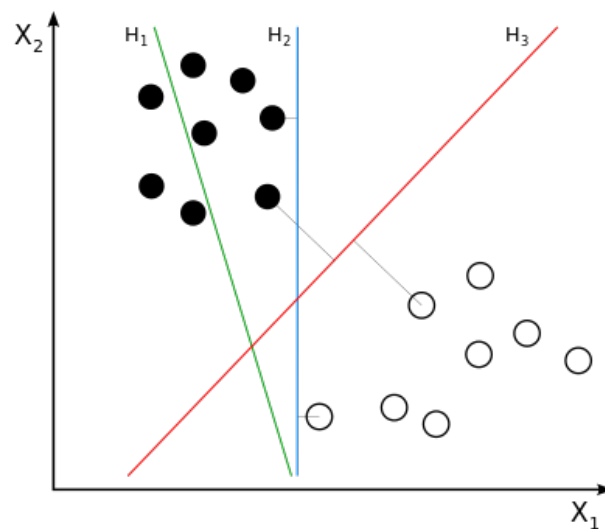
Het gebruik van een beslissingsboom is een leermethode die met regelmaat terugkomt in de statistiek als een voorspellend model. Men maakt gebruik van observaties rond een bepaald onderwerp om een beslissing te nemen rond de aard van het onderwerp. Deze beslissing leidt naar een variabele outputwaarde. 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. Eerst en vooral worden lagen horizontaal weergegeven t.o.v. verticaal bij Neurale Netwerken. Zoals te zien is in figuur 1.4 wordt 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 dit statement. Bij een eindnode is het niet meer mogelijk om door te gaan naar een volgende laag, maar wordt er een outputwaarde gegeven.

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 classificator. Om aan de hand van kenmerken een onderverdeling te maken moeten deze kenmerken eerst omgezet worden in een numeriek model. De data worden hervormd tot een vectorruimte. In de trainingsfase wordt er getracht een zo optimaal mogelijke scheiding tussen beide klassen te vinden. Deze optimale scheiding wordt ook wel een *hypervlak* genoemd en ligt op een zo groot mogelijke afstand tussen de dichtstbijgelegen objecten van beide klassen of support



Figuur 1.5: Twee dimensionale Support Vector Machine.[3]

vectors. in figuur 1.5 kan u een twee dimensionaal voorbeeld vinden. Hierin is scheidingslijn H_1 geen acceptabele scheiding omdat 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 omdat de foutenmarge tussen de twee klassen zo groot mogelijk is. Deze methode is niet alleen bruikbaar in toepassingen met een lineaire scheiding. Ook in niet-lineaire gevallen kan men een transformatie uitvoeren om toch een lineaire scheiding te bekommen. Deze hervorming wordt ook wel de *kernel trick* genoemd.

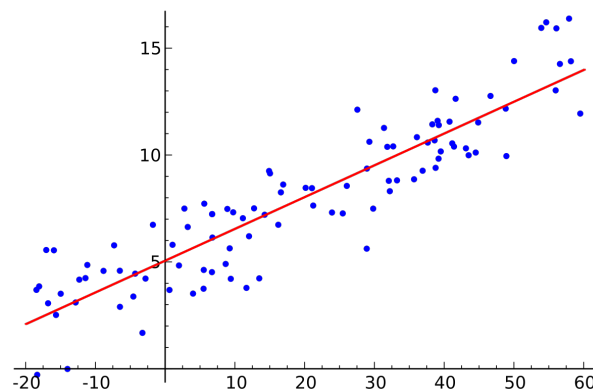
Regressie Analyse

Regressie Analyse is techniek uit de statistiek, die gebruikt wordt om gegevens te analyseren met een specifiek verband. Er bestaat vaak een relatie tussen een afhankelijke variabele en één (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 recente data of gebeurtenissen. In figuur 1.6 kan je een voorbeeld van lineaire regressie vinden.

Bayesian Netwerken



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



Figuur 1.6: Voorbeeld van Lineaire Regressie.[4]

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 mogelijk een betere oplossing geven voor een bepaald probleem.

1.2.2 Keuze voor een ML-methode

Om de keuze voor een bepaalde ML-techniek te verantwoorden bespreken we in dit hoofdstuk de voor- en nadelen van de voornaamste technieken die via regressie en classificatie toegepast kunnen worden.[6]

Regressie

• Neurale Netwerken:

Voordelen: Neurale netwerken zijn de **huidige state-of-the-art** in verschillende domeinen. NN kunnen uitstekend omgaan met onder meer beeld-, audio- en tekstdata en deze verwerken. Verder kan de architectuur ook nog gemakkelijk aangepast worden aan de toepassing door te variëren in het aantal lagen of nodes. Het gebruik van de hidden layers vermindert ook het hanteren van feature engineering.

Nadelen: Neurale netwerken zijn minder bruikbaar voor *general-purpose* algoritmes door de grote hoeveelheid data die er voor nodig zijn. In dat geval is het beter om voor beslissingsbomen te kiezen. Bovendien vragen ze veel **vermogen** voor het trainen van het netwerk en vragen veel expertise voor kleine aanpassingen **zoals architectuur en hyperparameters**.

• Regression Trees:

Voordelen: Beslissingsbomen met nadruk op regressie zijn in staat om niet-lineaire relaties te leren en zijn robuust voor uitschieters in de te verwerken dataset.

Nadelen: Regression trees zijn vatbaar voor **overfitting** indien er te veel gebruik gemaakt wordt van branches. **Bomen hebben vaak de neiging om branches aan te maken** tot het een

exacte kopie voorstelt van de trainingsdata.

- **Lineaire Regressie:**

Voordelen: Dit is een eenvoudige methode om zowel te verstaan als uit te leggen. Daarnaast kan er bescherming tegen overfitting geïmplementeerd worden.

Nadelen: Niet-lineaire relaties zijn een zwak punt voor lineaire regressie. Het is moeilijk om een correcte fitting te vinden voor een gegeven netelige relatie. Bovendien is het onvoldoende flexibel om complexe patronen op te vangen.

Classificatie

- **Neurale Netwerken:**

Voordelen: Neurale werken blijven uitstekend presteren bij het classificeren van audio-, tekst- en beeldherkenning.

Nadelen: Er is nood aan grote hoeveelheden data om het model te trainen en minder geschikt als general-purpose algoritmes.

- **Classification Trees:**

Voordelen: Verrichten zeer goed werk in praktijk. Ze zijn robuust voor uitschieters, schaalbaar voor meerdere klassen en kunnen niet-lineaire grenzen op natuurlijke wijze modelleren dankzij de hiërarchische structuur.

Nadelen: Classification trees zijn vatbaar voor overfitting indien er te veel gebruik gemaakt wordt van branches. Bomen hebben vaak de neiging om branches aan te maken tot het een exacte kopie voorstelt van de trainingsdata.

- **Support Vector Machines:**

Voordelen: SVMs zijn in staat om niet-lineaire beslissingsgrenzen te modelleren en hebben een sterke robuustheid overfitting, vooral in hogere dimensionale vectorruimtes.

Nadelen: SVMs zijn heel erg geheugen intensief. Ze vragen ook meer expertise in het afstemmen door het grote aanbod in mogelijke kernels. SVMs hebben de eigenschap om minder effectief te zijn bij het schalen naar grotere datasets.

- **Geregulariseerde Regressie:**

Voordelen: Outputs hebben een gemakkelijk leesbare probabilistische interpretatie. Ook kan er bescherming tegen overfitting geïmplementeerd worden en kunnen modellen eenvoudig geüpdatet worden.

Nadelen: Niet-lineaire relaties zijn een zwak punt voor lineaire regressie. Het is moeilijk om een correcte fitting te vinden voor een gegeven netelige relatie. Bovendien is het onvoldoende flexibel om complexe patronen op te vangen.

Besluit

Na een afweging gedaan te hebben van de belangrijkste kandidaten bij zowel regressie als classificatie toepassingen kunnen we besluiten dat Neurale netwerken een goede keuze is als model. We steunen vooral op het feit dat NN uitstekend werk levert in beide klassen in het analyseren van

data. Bovendien zijn de voornaamste nadelen minder van toepassing in het kader waarin NN toegepast zal worden. Het is vooral van belang hoe de netwerken in de executiefase presteren. Het trainen van de verschillende netwerken met grote hoeveelheden data kan op aparte systemen gebeuren. De trainingsfase is dus minder van belang voor het doel van deze thesis. Bovendien is het niet nodig om een breed general-purpose netwerk te voorzien.

Een mogelijk alternatief voor NN is het gebruiken van Regressie Analyse. De bepalende factor hiervoor is dat het karakter van de verscheidene applicaties vaak uit lineaire relaties bestaat.

1.2.3 Leertechnieken

er zijn verschillende mogelijkheden van een neurale netwerk te laten leren. De drie belangrijkste methodes om een mathematische functie te verkrijgen worden hieronder opgesomd.

Supervised Learning

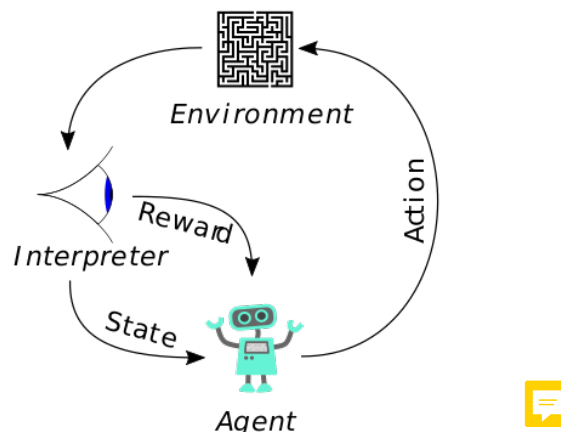
Deze techniek maakt gebruik van gepaarde datasets van inputobjecten en de te verwachten outputobjecten. Het doel is om een mathematische functie te creëren waarbij de gegenereerde outputs zo nauw mogelijk overeenkomen met de gelabelde outputs uit de datasets. Men optimaliseert deze mathematische functie 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 meest gebruikte 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 te verdelen in een cluster van gelijkaardige objecten. Een belangrijke toepassing 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 onderscheidt zich van Supervised en Unsupervised Learning door de onafhankelijkheid van gelabelde outputdatasets. Hierbij worden bovendien minder optimale acties niet manueel gecorrigeerd. De techniek heeft als doel om een evenwicht te vinden tussen exploratie van ongekend gebied en exploitatie van de huidige kennis. In figuur 1.7 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 1.7: Routine bij Reinforcement Learning

1.3 Evolutie SBC

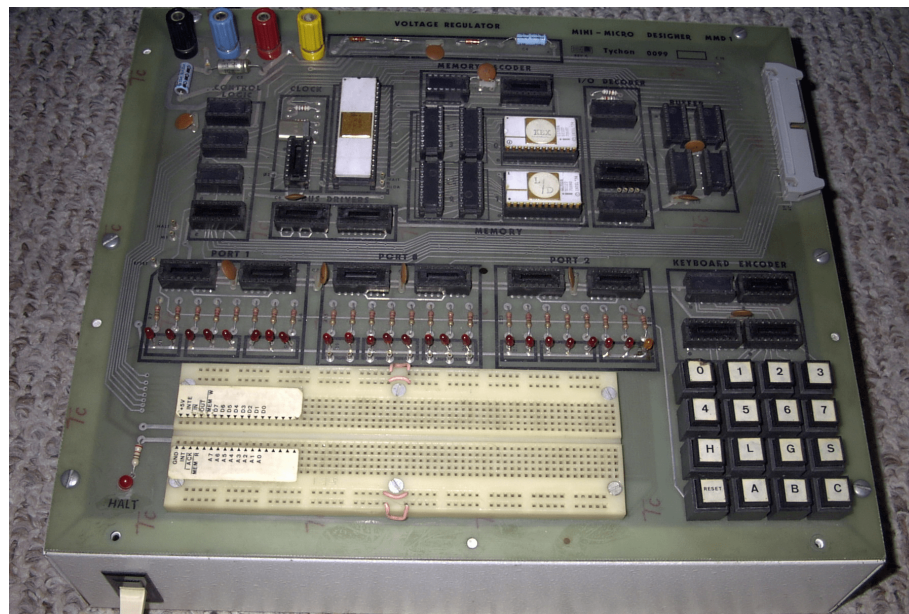
Een SBC is een volledige computer gemaakt op 1 enkele printplaat. Het bevat onderdelen zoals een microprocessor, geheugen, inputs en outputs. De SBC werd ontwikkeld als een voorstel-hulpmiddel bij educatieve doelstellingen of het gebruik als een embedded computer controller. Tegenwoordig zijn ook vele (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).

1.3.1 Geschiedenis

De eerste echte SBC was de zogenaamde "dyna-micro" uit figuur 1.8 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 de markt voor desktops en PC's groeide, nam de belangstelling voor SBC in computers meer en meer af. De focus van de markt werd verlegd 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 bracht de Universiteit van Cambridge 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.



Figuur 1.8: Eerste SBC: MMD-1 [5]

De laatste jaren kende een grote explosie aan nieuwe SBCs. 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.

1.4 Assortiment aan 'on the shelf' toestellen

1.4.1 Beaglebone AI

BeagleBone Ai (BB AI) is een SBC dat verder bouwt op de succesvolle BeagleBoard-series[7]. 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.

Specificaties

- **GPU:**
- **CPU:** Texas Instruments AM5729
- **Memory:** 16 GB on-board eMMC flash
- **Storage:** 1GB RAM + micro SD-slot
- **Power:**
- **Prijs:** \$139.37

1.4.2 Coral Dev Board

De Coral Dev Board is een development board gemaakt door het Amerikaanse technologiebedrijf Google[8]. Het board is ontworpen om het ontwikkelen van on-device ML producten te vergemakkelijken. Hiervoor heeft het een aantal belangrijke voordelen gekregen door zijn designers. Het is vooral de aangepaste Tensor Processing Unit (TPU) AI chips die hier opvalt. De TPU is een Application Specific Integrated Circuit (ASIC) speciaal ontworpen voor NN ML, en is in staat om video in hoge resolutie te analyseren aan 30 frames per second. Deze System-on-Module (SoM) is geoptimaliseerd om Tensorflow Lite te kunnen draaien aan meerdere tera operations per second (TOPS).

Specificaties

- **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:** 0.5 watts for each TOPS - 2 Watt
- **Prijs:** \$149.99

1.4.3 Nvidia Jetson Nano

De Jetson Nano is een populair bord uit de Jetson Series van Nvidia[9]. Het is een kleine maar krachtige computer ontwikkeld voor embedded applicaties en low-power AI-Internet-of-Things (IoT). Deze SBC wordt ondersteund door meerdere bibliotheken in sectoren zoals deep learning, computer vision, beeld en multimedia. De hardware bevat zowel een GPU als een Central Processing Unit (CPU). De GPU bestaat uit een krachtige Maxwell architectuur die beeld kan decoderen aan 500 MP/sec. De CPU is van het type Cortex-A57 met 4 kernen.

Specificaties

- **GPU:** 128-core Maxwell met 128 CUDA-cores
- **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
- **Prijs:** \$99

1.4.4 Nvidia Jetson TX2

De TX2, uit de zelfde Jetsonserie, is de high end versie van de hiervoor besproken Nano[10]. Het is het snelste en meest power-efficiëntst van de embedded AI toestellen. Deze gigant verbruikt een 7.5 Watt en brengt het zware AI-rekenwerk naar de edge. Zijn bekwame GPU met 256 CUDA kernen en een duo CPU zijn in staat om de meest geavanceerde Machine Leertechnieken uit te voeren. De grote geheugenvoorzieningen zorgen bovendien dat de datasetgrootte geen beperkende factor meer kan spelen. Verder wordt er nog gezorgd voor een grote ondersteuning via een grote variatie aan hardware interfaces. Hierdoor wordt het integreren van producten aanzienlijk makkelijker.

Specificaties

- **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
- **Prijs:** \$399

1.4.5 Raspberry Pi

Het laatste SBC die we bespreken is de Raspberry Pi 4[11]. Dit is een heel goedkoop en simpel device. Het komt uit de heel gekende Raspberry-reeks zoals al besproken in paragraaf 1.3.1. Ondanks de enorm lage prijs heeft de vierde editie nog mooie specificaties. Het heeft een behoorlijke Cortex Quad core-CPU, degelijk geheugen met de mogelijkheid om uit te breiden m.b.v. een SD-kaart en een verrassend laag verbruik.

Specificaties

- **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
- **Prijs:** \$35

1.5 Benchmarking van ML algoritmes

Om een betekenisvolle resultaten te verkrijgen is het nodig om een goede benchmark op te stellen. Een benchmark is een onderzoek waarbij de prestaties van programma's met elkaar vergeleken worden. Dit komt tot stand als elk programma op identieke wijze wordt onderzocht. Hoe de benchmark exact in elkaar zit is gebonden aan de kwaliteitscriteria die onderzocht worden. In het kader van deze thesis, zal er vooral aandacht voor latency zijn.

Om te voorkomen dat de resultaten afhankelijk zijn van slechts één programma, algoritme, toepassing... zal er voor gekozen worden om spreiding te introduceren. Dit kan gebeuren door op meerdere toepassingen en algoritmes te keuren. Zo moet men niet testen op enkel classificatie- of regressietoepassingen. Een combinatie van beide geeft een beter inzicht en voorkomt dat één anomalie de resultaten bezoedelt.

Hoofdstuk 2

Inleiding

In dit hoofdstuk gaan we een voorbeeld geven van een voetnoot¹. Een referentie naar hoofdstuk 3, dat zich op pagina 23 bevindt, is dus ook een koud kunstje. Zorg er wel voor dat je de namen van de labels een beetje verstandig kiest. Hoofdstukken label je het best als hfdstk:naam, plaatjes als img:naam en tabellen als tabel:naam. Zo verlies je zelf de bomen in het bos niet.

¹ Dit is dus een voetnoot

SDffjfhdsffhhsf fh fhf shf klfh fffsdfklfhklfhklfhfhklfhklfhklfhffhsdfhfhfhfhfh

dhhffh hf fh fh fhfh fhfh hfh fhffhsdfhfhfhfhfhhsdfh hfh fh

Hoofdstuk 3

Bespreking

In het vorig hoofdstuk hebben we naar deze tekst verwezen.

Hoofdstuk 4

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 XXXXX
 XXXX XXXXX XXXXX XXXX XXXXX XXXX XXXXXXXXXXXX

4.1 This is a Section Heading

Figuur 4.1: This is the Caption for Figure 1

Bibliografie

- [1] <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [2] <https://medium.com/machine-learning-bites/machine-learning-decision-tree-classifier-9eb67cad263e>.
- [3] https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788994170/5/ch05lvl1sec36/support-vector-machines.
- [4] https://en.wikipedia.org/wiki/Regression_analysis.
- [5] <https://nl.wikipedia.org/wiki/Singleboardcomputer>.
- [6] <https://elitedatascience.com/machine-learning-algorithms>.
- [7] <https://beagleboard.org/ai>.
- [8] <https://venturebeat.com/2019/10/22/googles-coral-ai-edge-hardware-launches-out-of-beta/>.
- [9] <https://developer.nvidia.com/embedded/jetson-nano>.
- [10] <https://developer.nvidia.com/embedded/jetson-tx2>.
- [11] <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>.

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**