

Vertrouwelijk: EMI buffer management



NAAM	STUDENTNUMMER	OPDRACHTGEVER	FUNCTIE	DATUM
S. A. Rang	1655299	EKB Houten	Afstudeerder	17-4-2018

INHOUDSOPGAVE

Begrippenlijst	4
Inleiding	5
Context	6
Het bedrijf	6
EKB Houten	6
Bedrijfsgegevens	7
Persoonsgegevens	7
Positie van de student binnen EKB	8
Relaties	8
Relatie afstudeeropdracht met SIE	8
Relaties met andere projecten	8
De Opdracht	9
De afstudeeropdracht	9
De kwestie	9
Theory Of Constraints	9
De afstudeeropdracht in het kort	10
Projectgrenzen en randvoorwaarden	10
Doelstelling van de afstudeeropdracht	11
Op te leveren producten	11
Vertrouwelijkheid	11
Het Onderzoek	12
Hoofdvraag en deelvragen	12
Onderzoeksmethoden	13
Theoretisch kader	14
Tsubaki Nakashima	14
EMI	15
Machine Learning	16
Perceptron	16
Neural Networks	16
Activation Functions	17
Supervised learning	19
Unsupervised learning	20
Reinforcement learning	21
Genetic algorithms	21
Te onderzoeken literatuur	23
Plan van Aanpak	24
Globale aanpak	24
Programmeermethoden	24
Tijdsplanning en mijlpalen	25
Risico's	26
Risico analyse	26
Persoonlijke uitdagingen	26
Literatuur	27

BEGRIPPENLIJST

BEGRIJP	DEFINITIE
Bottleneck	De zwakste machine of productielijn van een fabriek met de laagste capaciteit waardoor het aantal geproduceerde producten van de hele fabriek gelimiteerd wordt
Buffer	De voorraad van producten of halffabricaten die staat te wachten tussen twee productielijnen of machines tot het verder verwerkt kan worden
Capaciteit	Het aantal producten dat een productielijn of machine in 24 uur kan produceren
Deep Learning	Machine learning met meerdere serie geschakelde hidden layers
Efficiëntie	Een percentage van de maximale snelheid van een productielijn of machine
EKB Manufacturing Intelligence	Een industrieel webbased automatiserings software pakket voor verzameling en visualisatie van realtime informatie over productielijnen
KPI	Key Performance Indicator, een variabele om de prestatie van, in dit geval, een productielijn of machine te meten
Machine Learning	Het verwerken van data d.m.v. een algoritme dat niet zelf geprogrammeerd is, maar d.m.v. supervised, unsupervised learning of een genetic algorithm wordt gegenereerd
Normaliseren	Het standaardiseren van numerieke gegevens voor een machine learning algoritme, meestal tussen de -1 en 1 zodat het algoritme makkelijker om kan gaan met grote cijfers.
Ploeg	Een ploeg betekent dat een bepaalde hoeveelheid van een type product er 8 uur over doet om in de slijperij verwerkt te worden. Bijv. 500kg van product A is 1 ploeg en duurt 8 uur in de slijperij, maar 500kg van product B is 2 ploegen en duurt 16 uur in de slijperij.
Productielijn	Een serie geschakelde verzameling machines waarmee in een fabriek producten worden geproduceerd
Programmable Logic Controller	Een apparaat die commando's van buitenaf via een microprocessor omzet naar acties op productielijnen of machines
Redmine	Een open source management applicatie die gebruikt zal worden voor SCRUM
Theory of Constraints	Een maatschappelijk geaccepteerde manier om de bottleneck van een systeem te vinden en deze te exploiteren totdat een ander gedeelte van het systeem de bottleneck wordt. Dan wordt het proces herhaalt.

1 INLEIDING

In dit document worden de belangrijkste punten van deze afstudeeropdracht beschreven. Een aantal belangrijke begrippen die bij deze afstudeerstage van toepassing zijn; Machine learning, Theory of Constraints en buffers. Deze zullen in het vervolg van dit document verder worden toegelicht.

De afstudeerstage vindt plaats bij EKB Houten, een industrieel automatiseringsbedrijf dat processen binnen deze branche implementeert. De student zal op de development afdeling van EKB Houten onderzoek doen naar de huidige situatie bij Tsubaki Nakashima (TN), een klant van EKB, en de verschillende machine learning algoritmes. Daaruit voortvloeiend zal een eindproduct of proof of concept opgeleverd worden waarin het beste algoritme is geïmplementeerd om de grote hoeveelheid onoverzichtelijke data van verschillende productielijnen van TN te verwerken tot overzichtelijke, bruikbare data.

Het document is als volgt ingedeeld: Er wordt begonnen met wat achtergrondinformatie over het bedrijf en de betrokkenen van de afstudeeropdracht. Vervolgens wordt de afstudeeropdracht omschreven zoals deze is voortgekomen uit de gesprekken tussen afstudeerder, bedrijfsbegeleider en product owner. Hierna wordt beschreven hoe het onderzoek van de afstudeeropdracht eruit zal gaan zien. Ten slotte worden de eventuele risico's van deze afstudeeropdracht omschreven en worden er mogelijke oplossingen gegeven mochten deze risico's werkelijkheid worden.

2 CONTEXT

2.1 Het bedrijf

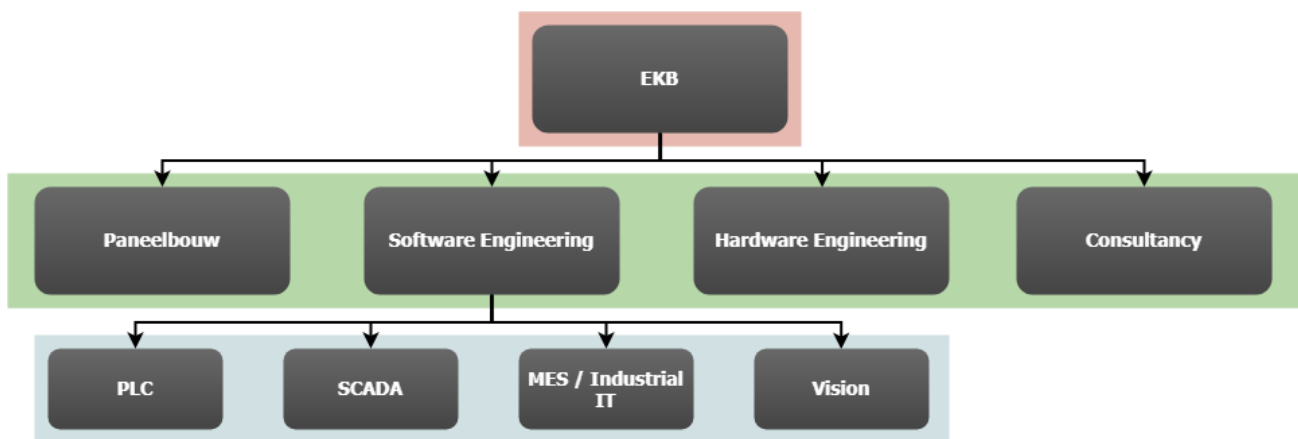
2.1.1 EKB Houten

EKB is actief op het gebied van industriële automatisering en richt zich vooral op het aaneensluiten en implementeren van processen hierbinnen.

Met 200 medewerkers verdeeld over vijf vestigingen bieden zij automatiseringsoplossingen voor de Nederlandse industrie.

EKB realiseert industriële automatiseringsprojecten voor de Nederlandse eindgebruikers en machinebouwers. EKB is vooral actief in de sectoren Metaal, Voedingsmiddelen, OffShore en Fijn Chemie. De belangrijkste activiteiten zijn weergegeven in Figuur 1. De student zal komen te werken op de MES (Manufacturing Execution Systems) afdeling van software engineering.

Andere activiteiten van EKB zijn paneelbouw, hardware engineering en consultancy.



Figuur 1: EKB activiteiten

2.1.2 Bedrijfsgegevens

Tabel 1: Bedrijfsgegevens van EKB

NAAM	ADRES	POSTCODE	TELEFOON	WEBSITE
EKB	Meidoornkade 19	3992 AG Houten	+31 30 711 14 80	http://www.ekb.nl/nl/home/

2.1.3 Persoonsgegevens

Tabel 2: Persoonsgegevens van betrokkenen

NAAM	FUNCTIE	E-MAILADRES	TELEFOON	LINKEDIN
S. A. Rang	Afstudeerder	stefan.rang@student.hu.nl	+31 6 34 10 04 29	https://www.linkedin.com/in/stefan-rang-8b0635101/
F. Verbruggen	Docentbegeleider	frank.verbruggen@hu.nl	+31 6 12 20 22 74	https://www.linkedin.com/in/frank-verbruggen-5080a720/
A. Roelofsen	Bedrijfsbegeleider	a.roelofsen.ekb.nl	+31 6 20 96 48 14	https://www.linkedin.com/in/auke-roelofsen-273b7918/
M. de Lange	Product Owner	w.de.lange@ekb.nl	+31 6 51 83 97 90	https://www.linkedin.com/in/michiel-de-lange-a1b04b3/
G. Bargeman	Contactpersoon Tsubaki Nakashima	ger.bargeman@europe.tsubaki-nakashima.com	+31 6 24 36 58 85	https://www.linkedin.com/in/ger-bargeman-b3332a14/
M. Kok	Software Engineer	m.kok@ekb.nl	+31 6 12 60 62 73	https://www.linkedin.com/in/maarten-kok-316374109/

Frank Verbruggen is docent bij de Hogeschool Utrecht en de docentbegeleider voor deze afstudeeropdracht.

Auke Roelofsen is projectmanager bij EKB Houten en tevens bedrijfsbegeleider voor deze afstudeeropdracht. De heer Roelofsen heeft gestudeerd aan zowel de Hogeschool Saxion als de Wageningen University & Research en heeft mijns inziens genoeg kennis en ervaring om de benodigde begeleiding te bieden op HBO niveau.

Michiel de Lange is directeur van EKB Houten. De heer De Lange zal bij deze afstudeeropdracht fungeren als Product Owner.

Ger Bargeman is LEAN Manager bij Tsubaki Nakashima in Veenendaal. De heer Bargeman is de contactpersoon voor Tsubaki Nakashima waar deze afstudeeropdracht voor uitgevoerd zal voeren in opdracht van EKB Houten.

Maarten Kok is een .NET software engineer op de development afdeling van EKB Houten en heeft kennis over PLC, EMI en machine learning wat van waarde kan zijn voor deze afstudeerstage. De heer Kok heeft dan ook aangeboden deze kennis te delen om te helpen met de vooruitgang van de afstudeeropdracht.

2.2 Positie van de student binnen EKB

De student zal bij de EKB vestiging in Houten als afstuderend developer op de development afdeling de volgende taken uitvoeren:

- Onderzoek doen
- Front-end programmeren
- Back-end programmeren
- Proof of concept/eindproduct realiseren

De student zal te allen tijde verantwoordelijk zijn voor eigen documenten, onderzoek, rapportages, code en eindproduct(en).

2.3 Relaties

2.3.1 Relatie afstudeeropdracht met SIE

In deze afstudeeropdracht komen de volgende vakken uit de Propedeuse en Hoofdphase van SIE terug:

- Human Computer Interaction (HCI)
- Patterns and Frameworks
- Programmeren in .NET
- Web Application Construction (WAC)
- Object Oriented Programming (OOP)
- Professional Skills

HCI komt terug in het user experience gedeelte van de front-end van het proof of concept/eindproduct. Om de grote hoeveelheden data overzichtelijk en begrijpelijk te houden dient er een goede user experience te zijn. Daarnaast wordt er voor de front-end net als bij WAC, HTML5 en CSS gebruikt en wordt de software voorzien van versiebeheer. Voor het programmeren van de back-end wordt .NET C# gebruikt waarbij object georiënteerd geprogrammeerd zal worden en er gebruikt zal worden gemaakt van verschillende patterns. Ten slotte worden de documenten geschreven volgens de standaarden van het vak professional skills.

2.3.2 Relaties met andere projecten

Het proof of concept/eindproduct(en) moet generiek, herbruikbaar en uitbreidbaar zijn. Dit betekent ook dat het proof of concept/eindproduct(en) door verschillende klanten van EKB met andere projecten gebruikt zou kunnen worden, maar in eerste instantie gaat de afstudeeropdracht over een enkel project van een klant van EKB.

3 DE OPDRACHT

3.1 De afstudeeropdracht

3.1.1 De kwestie

Sinds 2009 ontwikkelt EKB een eigen softwarepakket genaamd EMI (EKB Manufacturing Intelligence), gericht op industriële toepassing. EMI is vooral bedoeld om inzicht te krijgen in de productiviteit en kwaliteit van industriële productieprocessen. Deze data worden op dit moment vooral gebruikt om een overzichtelijk beeld te krijgen van de huidige situatie, maar nog niet om bepaalde productieprocessen te optimaliseren.

Vanaf de start van de ontwikkeling van EMI is er vanuit de industrie aangegeven dat er in toenemende mate beheer en sturing van interne buffervoorraden gewenst wordt. Hierbij wordt geëist dat Theory of Constraints wordt toegepast.

Als business case voor deze afstudeeropdracht wordt de data van de rollenfabriek van Tsubaki Nakashima, een klant van EKB, te Veenendaal gebruikt. De buffervoorraden worden op peil gehouden op basis van ervaring. Het is niet duidelijk hoe de buffervoorraden tussen de machines zo afgestemd kunnen worden dat de voorraden verminderen terwijl de productie vergroot wordt.

3.1.2 Theory Of Constraints

Een beknopte definitie van TOC volgens Procesverbeteren.nl:

TOC definitie Procesverbeteren.nl

Theory of Constraints (TOC) is net als Lean een (logistieke) verbetermethode die streeft naar meer *flow*, dus kortere doorlooptijden. Dit gebeurt bij de TOC echter nóg nadrukkelijker vanuit de gedachte dat alleen producten die de klant bereiken van belang zijn: $Profit = Throughput - Operating Expenses$. Een ander verschil is dat Lean uitgaat van stapsgewijs verbeteren door iedereen, terwijl de TOC bovenal grote doorbraken nastreeft en daardoor een 'strategischer' karakter heeft.

De TOC probeert steeds het knelpunt dat de afzet *op dat moment* het meeste beperkt op te sporen, en dit vervolgens maximaal uit te nutten (exploitatie) of zelfs op te heffen. Om een knelpunt te 'exploiteren' kunnen zowel TOC-, Lean- als QRM-tools worden ingezet. Het 'opheffen' van een knelpunt gebeurt niet altijd fysiek, het kan ook gaan om een oplossing voor een schijnbaar conflict, dat dan 'verdamp't'.

Knelpunten en oplossingen daarvoor vind je met behulp van de TOC *thinking processes*. Hierbij draait alles om het in kaart brengen van oorzaken en gevolgen, om "verborgen" kansen te vinden. Winst maken is binnen de TOC géén toeval, de mogelijkheden zijn schier *ongelimiteerd*. Kenmerkend is systeemdenken: een *supply chain* is zo sterk als de zwakste schakel, de *bottleneck*.

Bovenstaande definitie is tot stand gekomen op basis van onze honderden case-beschrijvingen, en vergelijking met de definities die andere TOC-denkers momenteel hanteren. Zie de Lean definitie voor overeenkomsten met Lean, en de QRM definitie voor overeenkomsten met QRM. (Van Ede, 2017)

3.1.3 De afstudeeropdracht in het kort

Het verlagen van buffervoorraden zorgt indirect voor kostenvermindering. Volgens Goldratt en Cox (2007) resulteert het verlagen van de voorraden echter alleen in een verhoging van de winst als ook de productie verhoogd wordt. Om dit te bereiken moeten dus zowel de verlaging van de buffervoorraden als de verhoging van de productie even zwaar meetellen.

Naast een verbredend en kritisch onderzoek naar TOC, machine learning en de huidige situatie bij TN (Tsubaki Nakashima) welke EKB van de student eist, dient de student een werkend product op te leveren waarin daadwerkelijk TOC toegepast en aantoonbaar gemaakt is voor de gebruiker. De afstudeerstage betreft dan ook een productopdracht.

Omdat het onduidelijk is hoe zowel de buffervoorraden te verlagen als de productie te verhogen zal machine learning worden gebruikt om een algoritme tot stand te laten komen die op basis van een fabriekssimulatie de variabelen af kan wegen. Om dit te kunnen implementeren in EMI moet echter wel eerst worden onderzocht hoe de simulatie zo realistisch mogelijk te maken is terwijl het wel generiek moet blijven zodat ook andere klanten van EKB deze simulatie kunnen gebruiken om een algoritme te genereren.

3.1.4 Projectgrenzen en randvoorwaarden

Ondanks dat er tientallen scenario's te bedenken zijn waarin machine learning zou kunnen worden gebruikt om het productieproces van TN te verbeteren kan er maar een gekozen worden voor deze afstudeeropdracht. Hierdoor kan de implementatie binnen de tijd en precies worden uitgevoerd. Ook is het dan makkelijker de implementatie herbruikbaar en generiek te houden.

TN heeft aangegeven dat de productieplanning op basis van klantorders nog gedeeltelijk met de hand wordt gedaan. Wel wordt er al een voorstel gedaan om voor bepaalde deadlines orders te produceren, maar het definitief inplannen van orders op de verschillende productielijnen wordt met de hand gedaan. Na enig onderzoek naar deze werkwijze is samen met de heer Bargeman en de heer Roelofsen besloten dit niet mee te nemen in de afstudeeropdracht omdat het lastig is om 'menselijke' beslissingen op basis van jarenlange ervaring aan een machine learning algoritme over te laten. Dit ook met oog op de vele uitzonderingen en aparte situaties die zorgen voor een andere productieplanning.

Een ander probleem van TN is slecht gereedschapsmanagement. In de huidige situatie worden de gereedschappen van machines op vaste tijden verwisseld. Hierdoor staan de machines tijdelijk stil en worden ook gereedschappen vervangen die nog niet vervangen hoeven te worden. Er worden op een van de productielijnen metingen gedaan waarmee een machine learning algoritme een voorspelling zou kunnen doen wanneer het gereedschap vervangen moet worden. Dit probleem valt echter buiten de scope van deze afstudeeropdracht.

Deze afstudeeropdracht zal gaan over het tunen van buffervoorraden. Oftewel, het verbeteren van de productie flow of doorstroom (Throughput). Dit zal worden gerealiseerd met een of meerdere machine learning algoritmes die generiek in EMI geïmplementeerd worden. Hierbij wordt zoveel mogelijk bestaande data van EMI gebruikt om te zorgen dat er geen externe data nodig zijn om de functionaliteiten bij een nieuwe klant te gebruiken.

3.1.5 Doelstelling van de afstudeeropdracht

De te bouwen uitbreiding van EMI moet ervoor gaan zorgen dat klanten van EKB hun buffervoorraden zo laag mogelijk kunnen houden terwijl de productie gelijk blijft en beter nog hoger wordt. Eventueel zou de software ook kunnen zorgen voor een minimale hoeveelheid omstellingen van machines en het gebruiken van minder productielijnen om dezelfde hoeveelheid productie te bereiken.

Door middel van de beschikbare data in EMI over de productielijnen moet het mogelijk worden de doorlooptijden van halffabricaten te verminderen. Hierdoor worden de voorraden tussen de verschillende machines en productielijnen kleiner en verminderen de kosten per product. De data over deze productielijnen die in EMI beschikbaar zijn, zijn gestandaardiseerd. Hierdoor blijft de nieuwe software generiek en kan het ingezet worden voor alle klanten van EKB die voldoende data opslaan in EMI.

3.1.6 Op te leveren producten

EKB verwacht aan het einde van de afstudeerstage ten minste een proof of concept/eindproduct in EMI. In Tabel 3 is de MoSCoW analyse te zien die voortgekomen is uit het eerste contact met de bedrijfsbegeleider.

Tabel 3: MoSCoW analyse van op te leveren producten

FUNCTIONALITEIT	MoSCoW
Machine learning algoritme voor het tunen van buffers rekening houdend met buffer kosten	Must Have
Generieke en uitbreidbare implementatie in EMI zodat met minimale aanpassing ook andere klanten gebruik kunnen maken van de functionaliteiten	Should Have
Visualisatie van de buffervoorraden van de eindklant	Should Have
Visualisatie van de toe-en afname van de verschillende buffers binnen een productielocatie	Should Have
Bottleneck rapportage	Could Have
Machine learning algoritme voor productieplanning	Would Have
Machine learning algoritme voor gereedschapsmanagement	Would Have

Een belangrijk punt voor EKB is dat de implementatie uitbreidbaar en generiek is. Echter is dit een Should Have, omdat het eindproduct ook kan functioneren zonder dat het uitbreidbaar en generiek is. Het plannen van de verschillende soorten producten die in een fabriek worden geproduceerd zou volgens de bedrijfsbegeleider ook een goede functionaliteit voor een algoritme zijn. Uiteindelijk is besloten dat dit de afstudeeropdracht te breed maakt en eigenlijk uit twee opdrachten zou gaan bestaan. Hierdoor is dit een Would Have functionaliteit.

3.2 Vertrouwelijkheid

EKB heeft duidelijk gemaakt dat zowel het werk bij TN, als het eindproduct/proof of concept vertrouwelijk is en alleen gedeeld mag worden met EKB, TN en de Hogeschool Utrecht. Daarbij moet vermeld worden dat het delen met de Hogeschool Utrecht alleen is bedoeld voor deze afstudeeropdracht en niet voor openbare doeleinden die de Hogeschool Utrecht zou kunnen hebben.

4 HET ONDERZOEK

4.1 Hoofdvraag en deelvragen

Hoe kunnen machine learning algoritmes, gericht op TOC, in EMI worden geïmplementeerd om de buffervoorraden van EKB klanten te verminderen?

Hoofdvraag decompositie

Om een machine learning algoritme te kunnen trainen zijn er allereerst een of meerdere relevante datasets nodig. Het is dus van belang dat er eerst onderzocht wordt welke data er nodig zijn. Om het eindproduct onafhankelijk van andere software te houden moet de data zoveel mogelijk uit de EMI database komen.

Om de machine learning algoritmes in de praktijk toe te kunnen passen moet de training gericht zijn op de werkelijkheid. Om dit te bereiken is er aanvullend onderzoek nodig naar de huidige situatie bij Tsubaki Nakashima om een realistische simulatie te kunnen maken waarin de machine learning algoritmes op de datasets kunnen trainen.

Voordat de simulaties met de verschillende algoritmes van start kunnen gaan is er nog de vraag hoe nu precies Theory of Constraints toe te passen? Hoe hebben andere bedrijven en afstudeerders de buffervoorraden proberen te verminderen met TOC? Met deze kennis kunnen de initiële parameters van de algoritmes worden ingesteld om vervolgens te kunnen simuleren. Deze simulaties resulteren in een lijst van algoritmes gesorteerd op prestaties.

Wanneer de keuze voor een bepaald algoritme is gemaakt kan deze worden geïmplementeerd in EMI. Om dit te kunnen doen moet eerst nog wel duidelijk worden welke architectuur het meest geschikt is om te kunnen implementeren.

De hieruit voortvloeiende deelvragen zijn als volgt:

- Welke data uit EMI en externe data zijn er nodig om een zo realistisch mogelijk algoritme te kunnen trainen?
- Hoe kan de training zo realistisch mogelijk worden gemaakt met de beschikbare data?
- Hoe hebben anderen met TOC de buffervoorraden verlaagd?
- Welke soorten machine learning algoritmes zijn geschikt om in combinatie met TOC toe te passen?
- Welke architectuur is het meest geschikt om de machine learning algoritmes volgens de randvoorwaarden en requirements in EMI te implementeren?

4.1.1 Onderzoeksmethoden

Per deelvraag is in Tabel 4 vastgesteld welke onderzoeksmethoden gebruikt zullen worden om tot het gewenste resultaat te komen en uiteindelijk de hoofdvraag te kunnen beantwoorden.

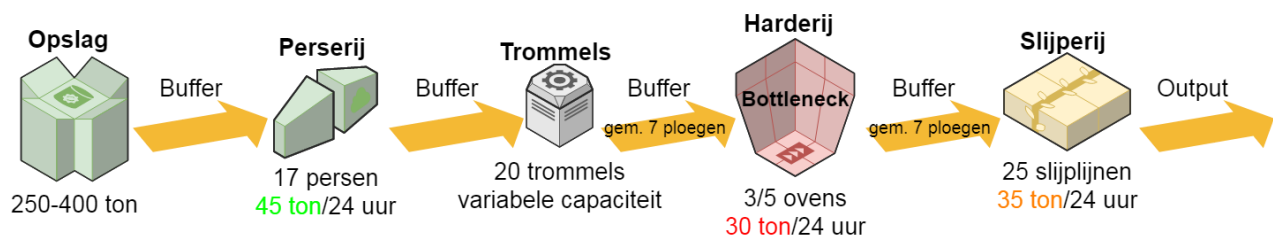
Tabel 4: Methoden matrix

DEELVRAAG	KWALITATIEF OF KWANTITATIEF?	ONDERZOEKSMETHODE	RESULTAAT
Welke data uit EMI en externe data zijn er nodig om een zo realistisch mogelijk algoritme te kunnen trainen?	Kwantitatief	Deskresearch	Verzameling van data waar het algoritme op kan trainen
Hoe kan de training zo realistisch mogelijk worden gemaakt met de beschikbare data?	Kwalitatief	Exploratief onderzoek / veldonderzoek	Een of meerdere training strategieën
Hoe hebben anderen met TOC de buffervoorraden verlaagd?	Kwalitatief	Deskresearch	Een of meerdere toepassingen van TOC ter inspiratie voor dit onderzoek
Welke soorten machine learning algoritmes zijn geschikt om in combinatie met TOC toe te passen?	Kwantitatief	Experimenteel onderzoek / laboratorium onderzoek / vergelijkend onderzoek	Een of meerdere algoritmes die gebruikt kunnen worden voor het eindproduct
Welke architectuur is het meest geschikt om de machine learning algoritmes volgens de randvoorwaarden en requirements in EMI te implementeren?	Kwalitatief	Deskresearch	Architectuur voor het eindproduct

4.2 Theoretisch kader

4.2.1 Tsubaki Nakashima

Deze afstudeeropdracht zal in eerste instantie worden uitgevoerd met de data van TN (Tsubaki Nakashima). Tijdens de implementatie zal echter wel rekening worden gehouden met het feit dat het eindproduct herbruikbaar en generiek moet zijn voor andere klanten van EKB. Omdat de algoritmes zullen leren van data van productielijnen van TN, is er tijdens het vooronderzoek informatie verzameld over de data van deze productie units en de samenhang hiertussen met interviews met contactpersoon Ger Bargeman. Hieronder volgt een samenvatting van de interviews.



Figuur 2: Productieflow van de rollenfabriek van Tsubaki Nakashima

De rollenfabriek van TN in Veenendaal maakt stalen cilindrische lagers voor o.a. auto onderdelen en hydraulische apparaten en is onderverdeeld in vijf hallen zoals te zien is in Figuur 2. In de eerste hal wordt het staal opgeslagen in hoepelvorm met een volume van gemiddeld 250 tot 400 ton. In de tweede hal wordt het staal in de juiste vorm geperst door 17 parallelle persmachines met een capaciteit van 45 ton/24 uur, maar zijn dan alleen grofweg in de juiste vorm. In de derde hal worden de halffabricaten geschuurd in 20 parallelle ronddraaiende trommels. De capaciteit van deze trommels is variabel en hangt af van zowel het type product als hoe de halffabricaten uit de perserij komen. Het is echter nog nooit gebeurd dat er te weinig trommels aanwezig waren om alle halffabricaten verder door te produceren, dus voor deze afstudeeropdracht is het niet belangrijk te veel aandacht te schenken aan de trommels. In de vierde hal worden de halffabricaten gehard door 5 parallelle ovens met een capaciteit van 30 ton/24 uur. De ovens zijn de bottleneck van de fabriek vanwege de lage capaciteit en staan daarom ook 24 uur/dag 7 dagen/week aan, maar kunnen niet elke soort product verwerken. Ook hierdoor zijn de ovens de bottleneck. Daarnaast worden er maar 3 van deze 5 ovens door EMI gemonitord en alleen van deze 3 zijn er data beschikbaar voor deze afstudeeropdracht. In de laatste hal worden de halffabricaten uiteindelijk in de juiste vorm geslepen door 25 parallelle lijnen van slijpmachines met een capaciteit van maximaal 35 ton/24 uur.

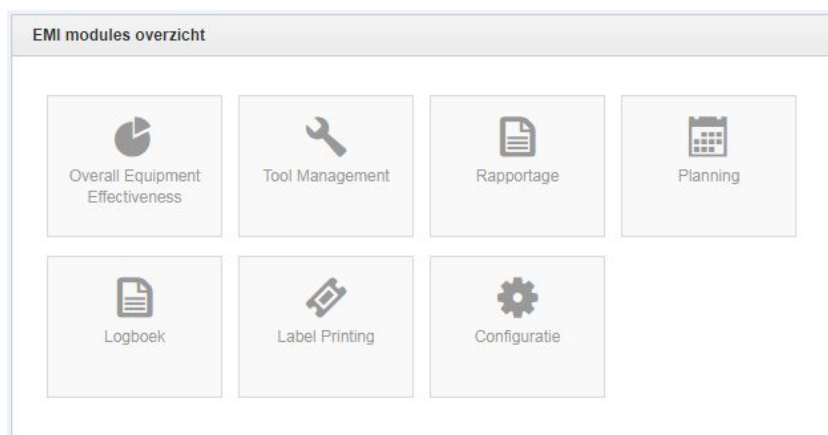
Bij TN spreken de werknemers van zogeheten ploegen. Dit is een door TN Veenendaal zelf vastgestelde maatstaf. Als bijvoorbeeld 500kg van een bepaald type product 1 ploeg is, betekent dit, dat het 8 uur duurt om deze 500kg producten door de slijperij te laten verwerken.

Tussen de hallen zijn zogenaamde buffers. Dit is opslag voor de halffabricaten die in containers worden bewaard met een track-en-trace-nummer wachtend op verwerking in de volgende hal. De buffers voor en na de ovens zijn gemiddeld 7 ploegen groot, nooit meer dan 14 en nooit minder dan 4 ploegen. Dit is zo geregeld zodat de bottleneck, oftewel de harderij, altijd kan produceren en de slijperij ook. Omdat de slijperij sneller is dan de harderij kan deze stil komen te staan als de buffer weggewerkt wordt.

Door de machines 'om te stellen' kunnen de machines voor verschillende producten worden gebruikt. Dit kost echter vaak uren tijd met meerdere werknemers. Het is daarom voor TN van groot belang dat het aantal omstellingen minimaal is.

4.2.2 EMI

EKB Manufacturing Intelligence is de applicatie waarin de software van deze afstudeeropdracht geïmplementeerd zal worden. Het bevat de modules Overall Equipment Effectiveness (OEE), Tool Management, Rapportage, Planning, Logboek, Label Printing en Configuratie zoals te zien in onderstaande foto (Rang, 2018).



Rang, S. (2018, 3 april). EMI modules van de rollenfabriek van Tsubaki Nakashima [Foto], Geraadpleegd van http://emi-demo.ekb.nl/EMI_NNN/Default.aspx

EMI communiceert met een SQL Server database waarin voor elke klant van EKB data voor deze zeven modules gestandaardiseerd worden bijgehouden. Deze data zijn dus goed te gebruiken om een generiek algoritme te schrijven, omdat er niet per klant dataconfiguratie plaats hoeft te vinden.



Rang, S. (2018, 3 april). EMI OEE analyse van de rollenfabriek van Tsubaki Nakashima [Foto], Geraadpleegd van http://emi-demo.ekb.nl/EMI_NNN/OEE/OEEAnalyse.aspx

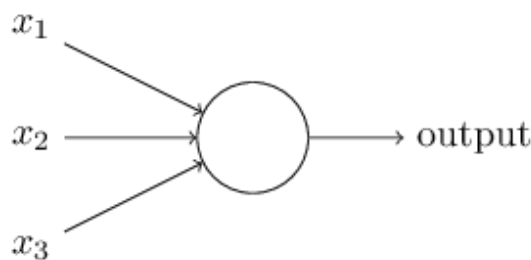
In bovenstaande foto (Rang, 2018) is een voorbeeld van de OEE analyse in EMI te zien van een persmachine genaamd 101RH1. De rode staaf van de KPI's is de beschikbaarheid van de

machine, oftewel hoe lang de machine heeft geproduceerd, ongeacht de snelheid of kwaliteit. De blauwe staaf van de KPI's is de prestatie van de machine. Dit gaat over de snelheid ten opzichte van de normsnelheid. Een prestatie van 50% betekent dat de machine gedurende de geselecteerde periode op de helft van de normsnelheid heeft geproduceerd. De gele staaf van de KPI's is de kwaliteit, oftewel hoeveel van de productie geaccepteerd is na een bepaalde check. Een kwaliteit van 75% betekent dat 25% van de productie niet goed genoeg is en wordt gezien als uitval of afkeur. Deze drie percentages van de KPI's worden vermenigvuldigd om te komen tot de groene OEE staaf. Dit is de Overall Equipment Effectiveness en geeft een samenvattend inzicht op de prestatie van de machine. In het geval van de 101RH1 is de OEE van januari 49,75%. Dit komt voornamelijk doordat de machine niet bediend is. Deze reden is terug te vinden rechtsonderin het scherm waar de status redenen gesorteerd staan gesorteerd op tijdsduur.

4.2.3 Machine Learning

4.2.3.1 Perceptron

Aan de basis van Machine Learning staan zogeheten Neural Networks. Dit is een manier om op basis van bepaalde inputs, outputs te berekenen. De meest simpele vorm hiervan is een *perceptron*. Volgens Nielsen (2017) kan een perceptron op basis van een som van meerdere inputs een output berekenen van 0 (uit) of 1 (aan). In Figuur 3 wordt een perceptron weergegeven. x_1 , x_2 en x_3 zijn de drie inputs van deze perceptron. De output is te berekenen met de formule van Figuur 6. Elke lijn van de inputs in Figuur 3 is een *weight* (w in Figuur 4). Dit is in essentie een getal, meestal tussen de -5 en 5 waarmee de input vermenigvuldigd wordt.



Figuur 3: Perceptron

Noot. Herdrukt van "Using neural nets to recognize handwritten digits", door Nielsen, M., (2017, 1 december). Geraadpleegd van <http://neuralnetworksanddeeplearning.com/chap1.html>

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

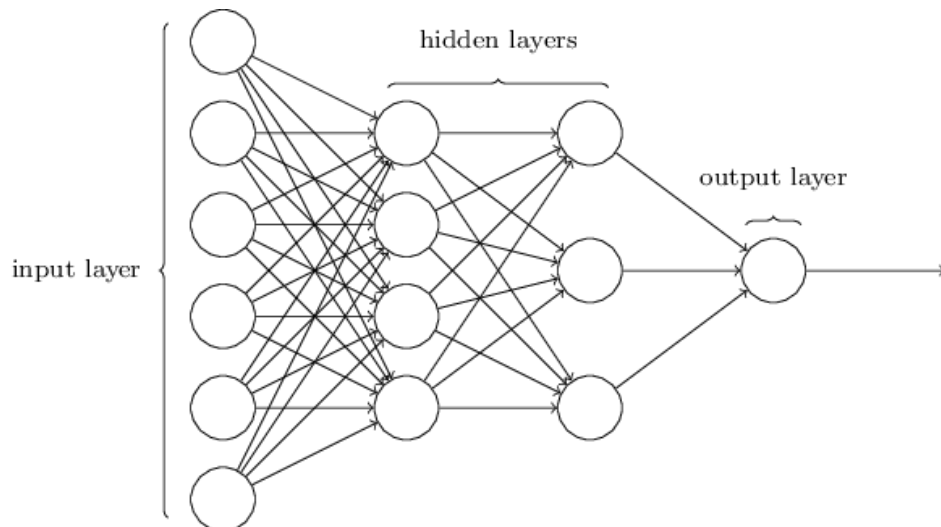
Figuur 4: Perceptron output formule

Noot. Herdrukt van "Using neural nets to recognize handwritten digits", door Nielsen, M., (2017, 1 december). Geraadpleegd van <http://neuralnetworksanddeeplearning.com/chap1.html>

Alle drie inputs worden vermenigvuldigd met hun bijbehorende weight. Deze drie uitkomsten worden bij elkaar opgeteld. Als de uitkomst hiervan boven de *threshold* ligt, is de output van de perceptron 1 en anders 0. De j in de formule van Figuur 4 is het nummer van de input x .

4.2.3.2 Neural Networks

Een neurale netwerk is niet meer dan een netwerk van meerdere lagen met perceptrons. Zoals te zien is in Figuur 5 bestaat een neural network uit een *input layer*, een *hidden layer* en een *output layer*. De input layer krijgt de waardes van de buitenwereld en berekend met de perceptrons en weights de waardes van elke perceptron in de opvolgende layers totdat de output perceptrons in de output layer berekend zijn.



Figuur 5: Neural Network

Noot. Herdrukt van “Using neural nets to recognize handwritten digits”, door Nielsen, M., (2017, 1 december). Geraadpleegd van <http://neuralnetworksanddeeplearning.com/chap1.html>

In het geval van neural networks spreken we echter over *neurons* in plaats van perceptrons. Het verschil met perceptrons is dat neurons een extra berekening hebben om tot de output te komen en deze output is niet digitaal maar analoog. Naast de waarde x en de bijbehorende weights w_n heeft een neuron ook een *bias* b . Dit is een extra variabele per neuron die opgeteld wordt bij de som van de weights keer de waardes van de neurons in de vorige layer. Daarna wordt extra berekening gedaan genaamd de *activation function*.

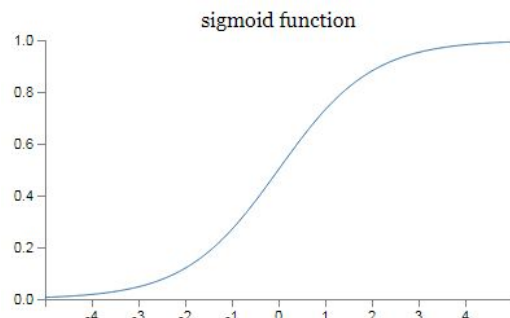
4.2.3.3 Activation Functions

Een van de meest gebruikte activation functions is de *sigmoid function*. Dit is een functie om elke willekeurige waarde om te zetten naar een getal tussen de 0 en de 1. Er zijn vele variaties op deze functie, maar de meest gebruikte versie is te zien in Figuur 6.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Figuur 6: Sigmoid function

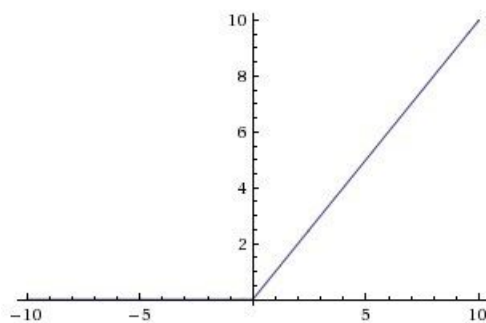
Zoals te zien is in Figuur 7 worden de waardes gelijkmatig verdeeld tussen de 0 en de 1. Volgens Rojas (1996) is het noodzakelijk een functie te gebruiken met een gelijkmatige helling om te zorgen dat het algoritme de weights van de neurons ook gelijkmatig kan veranderen om zo langzaam bij het gewenste resultaat te komen.



Figuur 7: Plot van de sigmoid function

Noot. Herdrukt van "Using neural nets to recognize handwritten digits", door Nielsen, M., (2017, 1 december). Geraadpleegd van <http://neuralnetworksanddeeplearning.com/chap1.html>

Een andere activation function die eigenlijk steeds vaker wordt verkozen boven de sigmoid function is de *ReLU (Rectified Linear Unit)*. Dit is een hele simpele maar effectieve activation function die alle negatieve waardes omzet naar 0 en niets doet met de positieve waardes. De functie luidt: $A(x) = \max(0, x)$. In Figuur 8 staat de plot van de ReLU functie.



Figuur 8: Rectified Linear Unit (ReLU) activation function

Noot. Herdrukt van "Rectified-Linear unit Layer", door Santos, L., (2018, 1 januari).

Geraadpleegd van

https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/relu_layer.html

In 2017 onderzocht Sharma de voor- en nadelen van de ReLU function. Een belangrijk voordeel is dat de negatieve waardes van neurons niet meer doorberekend hoeven te worden omdat ze door de ReLU function nul worden. Dit scheelt dus in de snelheid waarmee het neural network berekend kan worden. Een nadeel van ReLU is dat de functie geen helling heeft voor negatieve waardes. Dit betekent dat de neuron niet meer beïnvloed kan worden door kleine aanpassingen in de weights, omdat dit gebeurt op basis van de helling. Volgens Sharma (2017) heet dit de 'dying ReLU problem' waardoor deze neurons geen nut meer hebben voor het netwerk. Dit kan eventueel opgelost worden met een zogeheten *leaky ReLU*. Deze functie is: $A(x) = 0.01x$ voor negatieve waardes van x en $A(x) = x$ voor positieve waardes. Hierdoor is de helling voor negatieve waardes 0.01 waardoor de 'dying ReLU problem' wordt verholpen.

Tot slot is er ook de simpele hyperbolische tangens $f(x) = \tanh(x)$ die eigenlijk een variatie is op de sigmoid function: $\tanh(x) = 2\sigma(2x) - 1$. Het verschil is dat deze functie alle waardes van x omzet naar waardes tussen de -1 en 1 en dat de helling twee keer steiler is.

4.2.3.4 Supervised learning

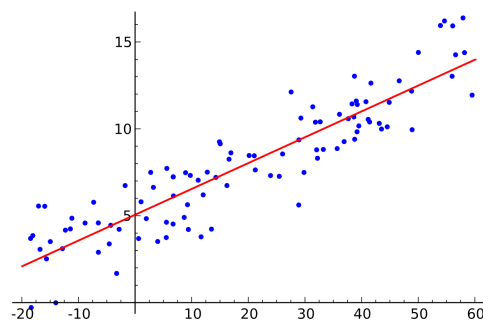
Supervised learning is een manier om een neural network of ander algoritme bepaalde patronen aan te leren. De uitkomsten van het neural network of algoritme zijn bekend. Hierdoor is het trainingsproces vaak vrij kort.

Een vorm van supervised learning is *backpropagation*. Zoals al eerder vermeld worden bij deze training de weights en biases van neurons in een neural network op basis van de helling van de activation function beetje bij beetje aangepast om steeds dichterbij het gewenste resultaat te komen. Deze techniek heet *gradient descent* en is al sinds de jaren 80 gebruikt, maar wordt steeds minder gebruikt. De reden hiervoor is dat voor veel problemen de oplossing nog niet bekend is. Wel wordt backpropagation veel gebruikt om dingen te automatiseren die mensen simpel en saai werk vinden zoals het herkennen van objecten in een video of afbeelding. Hierbij is voor een mens al van tevoren duidelijk wat er staat maar is het doel om deze patroonherkenning te automatiseren zodat nieuwe afbeeldingen op dezelfde manier verwerkt kunnen worden. Een voorbeeld is Google die in de Google Image Search dit algoritme gebruiken om relevantere resultaten te vinden.

Een andere vorm van supervised learning is *classification*. Bij deze techniek wordt niet gebruik gemaakt van een neural network, maar een algoritme om de training data te verwerken.

Bijvoorbeeld een hele hoop afbeeldingen van stukken fruit die moeten worden verdeeld in de groepen appels, peren en overigen. Deze groepen worden van tevoren vastgesteld.

Ook *regression* is een supervised learning methode. Ook hierbij wordt niet gebruik gemaakt van een neural network maar van een algoritme. Dit algoritme wordt gebruikt om een schatting te maken voor een functie die het dichtst in de buurt van de training data komt. Het maakt niet uit hoeveel parameters er gebruikt worden, maar wat wel van invloed is op de uitkomst is het aantal termen dat het algoritme gebruikt om de functie te formuleren.

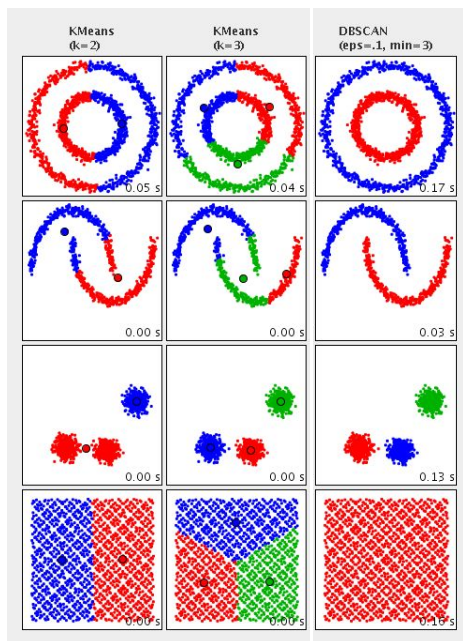


Figuur 9: Regression

Noot. Herdrukt van "Regression analysis", door Wikipedia, (2018, 1 april). Geraadpleegd van https://en.wikipedia.org/wiki/Regression_analysis

4.2.3.5 Unsupervised learning

Unsupervised learning is een manier om onbekende patronen te herkennen in data. Een voorbeeld hiervan is *clustering* van data zonder dat het aantal clusters of de namen hiervan bekend zijn. Een bekend algoritme hiervoor is *K-means* of *DBSCAN* (Figuur 10).

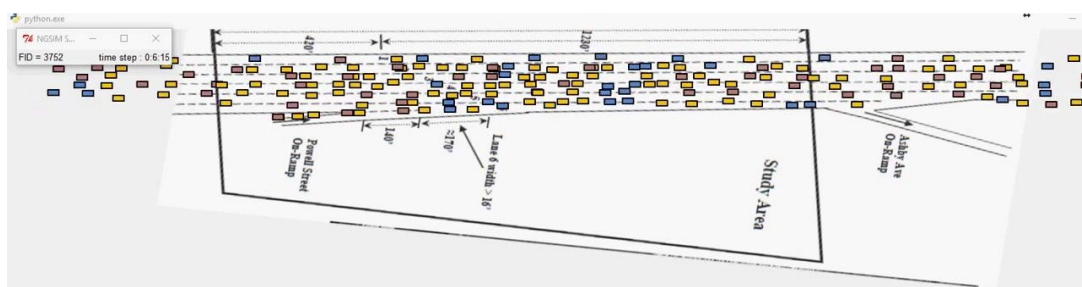


Figuur 10: K-means (2 en 3) en DBSCAN clustering algoritmes

Noot. Aangepast van “Clustering algorithms and distance measures”, door Apache, (2016, 28 augustus). Geraadpleegd van

<http://commons.apache.org/proper/commons-math/userguide/ml.html>

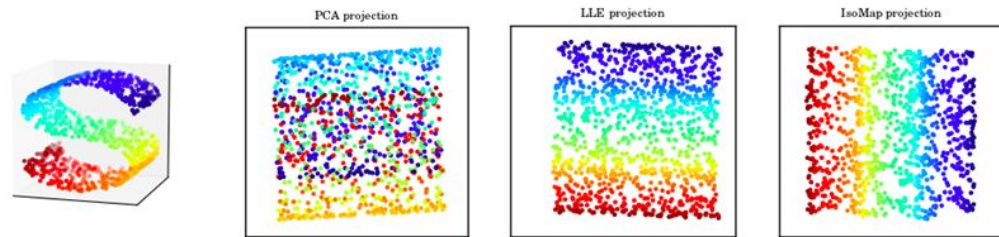
Naast clustering vallen onder unsupervised learning ook *density estimation* en *dimensionality reduction*. Density estimation wordt gebruikt om van een aantal datapunten een overzichtelijk beeld van de verdeling te krijgen. Dit kan bijvoorbeeld worden gebruikt om verkeersdrukte over een groot gebied te analyseren (Figuur 11). Dimensionality reduction is een manier om redundante of overbodige parameters uit een dataset te filteren of om data overzichtelijker weer te geven met minder parameters (Figuur 12).



Figuur 11: Density estimation algoritme voor verkeersdrukte

Noot. Aangepast van “NGSIM Simulator to evaluate a traffic density estimation algorithm by using sensor equipped vehicles”, door Nam, D., (2017, 23 oktober). Geraadpleegd van

https://www.youtube.com/watch?v=kpgR_DxhzQE

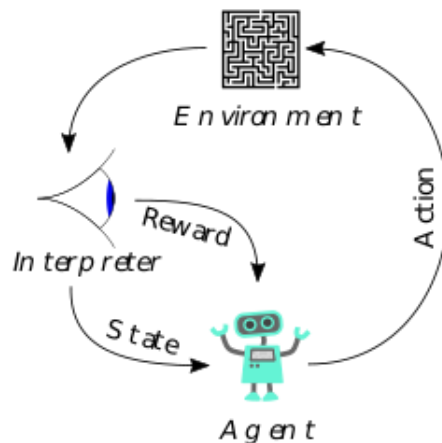


Figuur 12: Diverse dimensionality reduction algoritmes

Noot. Aangepast van “Comparison of PCA and Manifold Learning”, door Vanderplas, J., (2012). Geraadpleegd van http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html

4.2.3.6 Reinforcement learning

Volgens Van Otterloo en Wiering (2009) is reinforcement learning een combinatie van supervised- en unsupervised learning waarbij een *agent* in een omgeving streeft naar het maken van goede beslissingen op basis van gelimiteerde feedback. De agent voert acties uit op basis van de huidige omgeving, de *state*. Een *interpreter* analyseert vervolgens het resultaat van deze actie en geeft feedback op de agent dmv *rewards* en geeft de agent vervolgens de nieuwe state om de agent een nieuwe actie uit te laten voeren. Een voorbeeld is een spel zoals super mario waarin de state het huidige scherm is en de reward hoe dichtbij de agent is van de finish en de score in het spel. Wanneer de agent een hoge reward krijgt leert de agent dat de actie die hierbij hoorde goed was en wordt hierdoor beter in het uitvoeren van de juiste acties in vergelijkbare states.

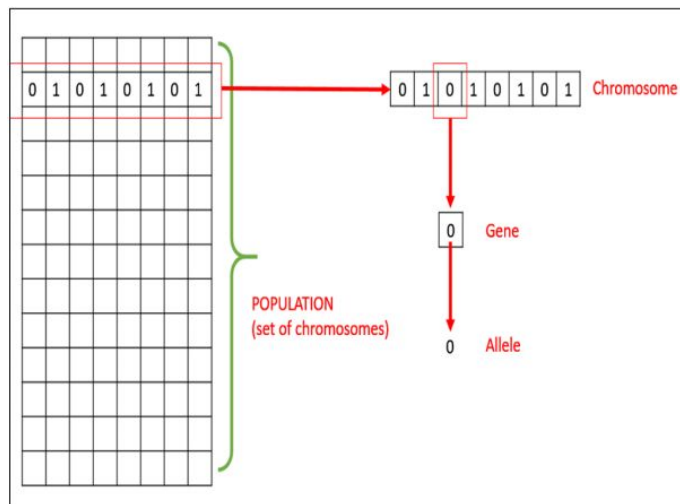


Figuur 13: Reinforcement learning algoritme

Noot. Herdrukt van “Reinforcement learning”, door Wikipedia, (2018, 2 april). Geraadpleegd van https://en.wikipedia.org/wiki/Reinforcement_learning

4.2.3.7 Genetic algorithms

Genetic algorithms kunnen ook weer worden onderverdeeld in vele verschillende algoritmes, maar in het kort is een genetic algorithm vergelijkbaar met reinforcement learning. Echter gebruikt een genetic algorithm een hele hoop agents tegelijkertijd zonder een interpreter. Deze agents worden *genomes* of soms *chromosomes* genoemd. Een genome is simpelweg een lijst van bits of analoge waarden die *genes* heten (Figuur 14). Deze waarden worden in de omgeving gescoord. Dit is de *fitness* waarmee via natuurlijke selectie een nieuwe *generation* (generatie of populatie) van genomes wordt gegenereerd waarvan opnieuw de fitness berekend wordt.



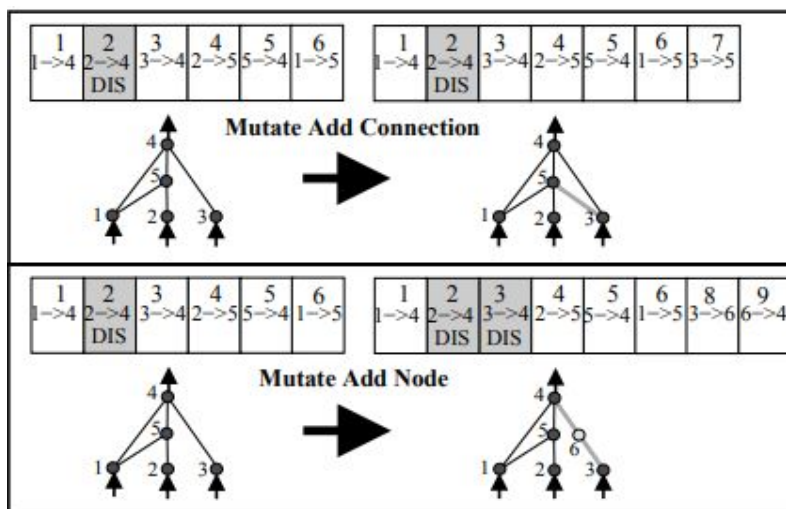
Figuur 14: De basis structuur en terminologie van genetic algorithms

Noot. Herdrukt van "Genetic Algorithms", door Tutorials Point (I) Pvt. Ltd., (2016).

Geraadpleegd van

https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_tutorial.pdf

Volgens Tutorials Point (I) Pvt. Ltd., (2016) is er geen helling nodig om de genomes te trainen, kun je meerdere genomes parallel laten trainen en train je niet een goed algoritme, maar een hele lijst van genomes die goed presteren. Echter is een genetic algorithm niet geschikt voor simpele problemen waar wel helling informatie over beschikbaar is of waarvan de oplossing al bekend is.



Figuur 15: De twee soorten mutatie van een neural network dmv het NEAT algoritme

Noot. Herdrukt van "Evolving Neural Networks through Augmenting Topologies", door Stanley, K., (2004). Geraadpleegd van <http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf>

Bij genetic algorithms staat het aantal genes vast en wordt er niet gebruik gemaakt van neural networks. Echter in het NEAT (NeuroEvolution of Augmenting Topologies) algoritme wordt er wel gebruik gemaakt van neural networks en wordt de architectuur van deze netwerken ook dmv natuurlijke selectie en evolutie verder ontwikkeld. Er kan zowel een weight-lijn als een een compleet nieuwe perceptron worden toegevoegd (Figuur 15). Daarnaast wordt de zoektocht naar het beste netwerk ook verbreed door het gebruik van zogeheten *species*. Volgens Stanley

(2004) wordt in het NEAT algoritme innovatie van nieuwe architecturen beschermd dmv species en het verdelen van de fitness onder genomes van dezelfde species. Daarnaast begint het NEAT algoritme ook met een zo klein mogelijke architectuur van het netwerk om een zo simpel en efficiënt mogelijke oplossing te vinden. Dit blijkt ook uit de experimenten die met NEAT zijn uitgevoerd door K. O. Stanley.

4.2.4 Te onderzoeken literatuur

Om de afstudeerstage te kunnen starten moet er eerst kennis worden opgedaan over een aantal belangrijke onderwerpen en termen. Deze zijn terug te vinden in Tabel 5.

Tabel 5: Voorlopige literatuurlijst

ONDERWERP/TERM	LITERATUUR
TOC	<p>Goldratt, E. M. (1986). Het Doel. Houten, Nederland: Spectrum.</p> <p>Goldratt, E. M., & Cox, J. (2007, 2 april). The Goal, A Process of Ongoing Improvement. Geraadpleegd van http://www.2ndbn5thmar.com/lean/Notes%20on%20The%20Goal.pdf</p> <p>Gattiker, T. (2015, 21 oktober). Virtual Lecture: Theory of Constraints / The Goal (Part 1) [Video]. Geraadpleegd van https://www.youtube.com/watch?v=riqvCu5FBiw</p> <p>Gattiker, T. (2015, 21 oktober). Virtual Lecture: Theory of Constraints / The Goal (Part 2) [Video]. Geraadpleegd van https://www.youtube.com/watch?v=rzuDLFTkolg</p> <p>Ede, J. van. (2017, 6 september). TOC definitie Procesverbeteren.nl. Geraadpleegd van https://www.procesverbeteren.nl/TOC/ToC.php</p>
EMI	<p>Volg en vind met EMI - Traceability in de procesindustrie. (2014, 1 januari). Geraadpleegd van http://www.ekb.nl/nl/nieuws/volg_en_vind_met_emi</p>
Machine Learning	<p>Stanley, K. O. (2004). Efficient Evolution of Neural Networks through Complexification (Report AI-TR-04-314). Geraadpleegd van http://nn.cs.utexas.edu/downloads/papers/stanley.phd04.pdf</p> <p>Olah, C. (2015, 27 augustus). Understanding LSTM Networks [Blogpost]. Geraadpleegd van https://colah.github.io/posts/2015-08-Understanding-LSTMs/</p> <p>Heaton, J. (2014, 1 oktober). Encog 3.3: Development Guide. Geraadpleegd van https://s3.amazonaws.com/heatonresearch-books/free/encog-3_3-devguide.pdf</p> <p>Nielsen, M. A. (2017). <i>Neural Networks and Deep Learning</i>. Geraadpleegd van http://neuralnetworksanddeeplearning.com/index.html</p> <p>Rojas, R. (1996). The Backpropagation Algorithm. Geraadpleegd van https://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf</p> <p>Tutorials Point (I) Pvt. Ltd.. (2016). Genetic Algorithms. Geraadpleegd van https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_tutorial.pdf</p> <p>Otterlo, M. van, & Wiering, M. (2009). Reinforcement Learning and Markov Decision Processes. Geraadpleegd van http://www.ai.rug.nl/~mwiering/Intro_RLBOOK.pdf</p>

5 PLAN VAN AANPAK

5.1 Globale aanpak

Nadat het Plan van Aanpak goedgekeurd is zal er gestart worden met extra kennis vergaren over machine learning, TOC en EMI om een plan te maken voor de verschillende experimenten. Vervolgens zal er een experimentele omgeving bij TN worden opgezet om met verschillende machine learning algoritmes te experimenteren op testdata van een of meerdere productielijnen van TN. Uit deze experimenten volgen een of meerdere geschikte algoritmes die ingezet kunnen worden voor het eindproduct. Daarna zullen er meerdere interviews met de opdrachtgever plaatsvinden over de uitstraling en werking van het eindproduct waarna via SCRUM het eindproduct met het machine learning algoritme wordt gerealiseerd.

5.2 Programmeermethoden

Bij het eventueel realiseren van een of meerdere machine learning algoritmes, maar ook voor het realiseren van het eindproduct zal er gewerkt worden volgens de SCRUM-project techniek waarbij M. de Lange als product owner zal fungeren en de student zelf als SCRUM-master het project zal doorlopen. Voor de backlog van SCRUM wordt Redmine gebruikt. EKB verwacht dat de software modulair opgebouwd zal worden en herbruikbaar en uitbreidbaar is. Ten slotte wordt het eindproduct van een bijgehouden versienummer voorzien en wordt de nieuwste versie en oudere versies opgeslagen in Git.

De front-end wordt geprogrammeerd in HTML5, CSS, en Javascript. De back-end zal worden gerealiseerd in het ASP .NET framework in de C# taal en er zal gebruik worden gemaakt van SQL voor communicatie met de database.

5.3 Tijdsplanning en mijlpalen

In Tabel 6 wordt de tijdsindeling en de verschillende mijlpalen van deze afstudeerstage weergegeven.

Tabel 6: Tijdsplanning en mijlpalen

ACTIVITEIT	START- EN EINDDATUM	UREN
Plan van Aanpak	Maandag 5 maart 2018 - Dinsdag 10 april 2018	216
Deelvraag 1	Woensdag 11 april 2018 - Donderdag 12 april 2018	16
Deelvraag 2	Vrijdag 13 april 2018 - Donderdag 19 april 2018	40
Deelvraag 3	Vrijdag 20 april 2018 - Maandag 23 april 2018	16
Deelvraag 4	Dinsdag 24 april 2018	8
Deelvraag 5	Woensdag 25 april 2018 - Donderdag 26 april 2018	16
Back-end programmeren	Vrijdag 27 april 2018 - Donderdag 3 mei 2018	40
Front-end programmeren	Vrijdag 4 mei 2018 - Maandag 7 mei 2018	16
Scriptie schrijven	Dinsdag 8 mei 2018 - Donderdag 24 mei 2018	104
Eerste conceptversie scriptie inleveren	Donderdag 24 mei 2018	n.v.t.
Scriptie herschrijven	Woensdag 30 mei 2018 - Dinsdag 12 juni 2018	80
Tweede conceptversie scriptie inleveren	Dinsdag 12 juni 2018	n.v.t.
Definitieve versie scriptie schrijven	Woensdag 20 juni 2018 - Dinsdag 3 juli 2018	80
Definitieve versie scriptie inleveren	Dinsdag 3 juli 2018	n.v.t.
Totaal	Maandag 5 maart 2018 - Dinsdag 3 juli 2018	632

6 Risico's

6.1 Risico analyse

In Tabel 7 worden de verwachte risico's van deze afstudeeropdracht weergegeven. De impact van de risico's is eigenlijk altijd redelijk hoog aangezien het anders geen risico is. De kans dat het risico zich tijdens het afstudeertraject voordoet samen met de impact geeft een beeld van de prioriteit van het risico.

Tabel 7: Risico analyse

OMSCHRIJVING	IMPACT	KANS	RISICO	MAATREGELEN
Onvoldoende tijd voor documentatie/scriptie	Hoog	Gemiddeld	Hoog	Aanpassen van de tijdsplanning zodat er meer tijd gereserveerd wordt voor het documenteren
Onvoldoende of niet representatieve data voor de machine learning algoritmes	Hoog	Laag	Gemiddeld	Overwegen of er meer externe data gebruikt zou kunnen worden om de data uit te breiden en representatiever te maken, of aannames doen over bepaalde variabelen die aanpasbaar gemaakt kunnen worden in de implementatie
Onvoldoende tijd voor Must Have functionaliteiten	Hoog	Laag	Gemiddeld	Deze functionaliteiten kunnen worden opgenomen in het Proof of Concept en/of worden uitbesteed aan de developers van EKB
Onvoldoende uitbreidbare en generieke implementatie	Gemiddeld	Gemiddeld	Gemiddeld	Goede documentatie, code commentaar en het Proof of Concept zorgen ervoor dat EKB over de kennis beschikt dit later alsnog te realiseren
Te lage prestaties van de machine learning algoritmes	Gemiddeld	Laag	Laag	Meer tijd reserveren voor aanvullende experimenten met andere parameters of eventueel andere algoritmes. Eventueel de implementatie van het algoritme opnemen in het Proof of Concept

De kans dat er onvoldoende of niet representatieve data beschikbaar zijn, is vrij laag, omdat er volgens M. Kok in EMI al sinds begin 2017 elke minuut van bijna alle productielijnen van TN metingen worden gedaan van de productie. Ook wordt er per productielijn bijgehouden aan welke orders er op dit moment wordt gewerkt en worden machine snelheden en storingen opgeslagen in de database van EMI. Voor de kostenafweging is er volgens G. Bargeman voldoende financiële data beschikbaar bij TN om dit te implementeren.

6.2 Persoonlijke uitdagingen

Een van de grootste uitdagingen aan de afstudeeropdracht is het schrijven van de scriptie zelf. Ik vind het lastig om beschrijvingen van experimenten of begrippen kort te houden en ga vaak snel de details in. Om de scriptie op een goede manier te schrijven zal ik mijn experimenten en begrippen kort en krachtig moeten omschrijven. Daarnaast zal het inplannen van de verschillende fases van de afstudeerstage ook een uitdaging worden, omdat zowel de onderzoeks- als de implementatiefase veel tijd in beslag zal nemen. Daarentegen zullen de meeste experimenten gedeeltelijk zonder toezicht verlopen waardoor er tijd vrij is voor andere activiteiten.

LITERATUUR

- Apache, (2016, 28 augustus). Clustering algorithms and distance measures. Geraadpleegd van <http://commons.apache.org/proper/commons-math/userguide/ml.html>
- Ede, J. van. (2017, 6 september). TOC definitie Procesverbeteren.nl. Geraadpleegd op 23 november 2017, van <https://www.procesverbeteren.nl/TOC/ToC.php>
- EMI OEE analyse van de rollenfabriek van Tsubaki Nakashima [Foto]. (2017). Geraadpleegd op 3 april 2018, van http://emi-demo.ekb.nl/EMI_NNN/OEE/OEEAnalyse.aspx
- EMI modules overzicht [Foto]. (2017). Geraadpleegd op 3 april 2018, van http://emi-demo.ekb.nl/EMI_NNN/Default.aspx
- Goldratt, E. M. (1986). Het Doel. Houten, Nederland: Spectrum.
- Goldratt, E. M., & Cox, J. (2007, 2 april). The Goal, A Process of Ongoing Improvement. Geraadpleegd van <http://www.2ndbn5thmar.com/lean/Notes%20on%20The%20Goal.pdf>
- Nam, D. (2017, 23 oktober). NGSIM Simulator to evaluate a traffic density estimation algorithm by using sensor equipped vehicles [Video]. Geraadpleegd van https://www.youtube.com/watch?v=kpgR_DxhzQE
- Nielsen, M. A. (2017). *Neural Networks and Deep Learning*. Geraadpleegd van <http://neuralnetworksanddeeplearning.com/index.html>
- Nielsen, M. (2017, 1 december). Using neural nets to recognize handwritten digits. Geraadpleegd van <http://neuralnetworksanddeeplearning.com/chap1.html>
- Otterlo, M. van, & Wiering, M. (2009). Reinforcement Learning and Markov Decision Processes. Geraadpleegd van http://www.ai.rug.nl/~mwiering/Intro_RLBOOK.pdf
- Rojas, R. (1996). The Backpropagation Algorithm. Geraadpleegd van <https://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf>
- Santos, L. (2018, 1 januari). Rectified-Linear unit Layer. Geraadpleegd van https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/relu_layer.html
- Sharma, A. (2017, 30 maart). Understanding Activation Functions in Neural Networks [Blogpost]. Geraadpleegd van <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>
- Stanley, K. O. (2004). Efficient Evolution of Neural Networks through Complexification (Report AI-TR-04-314). Geraadpleegd van <http://nn.cs.utexas.edu/downloads/papers/stanley.phd04.pdf>
- Tutorials Point (I) Pvt. Ltd.. (2016). Genetic Algorithms. Geraadpleegd van https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_tutorial.pdf
- Vanderplas, J. (2012). Comparison of PCA and Manifold Learning. Geraadpleegd van http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html

Wikipedia, (2018, 1 april). Regression analysis. Geraadpleegd van https://en.wikipedia.org/wiki/Regression_analysis

Wikipedia, (2018, 2 april). Reinforcement learning. Geraadpleegd van https://en.wikipedia.org/wiki/Reinforcement_learning