



# **STAGEVERSLAG**

## **REINFORCEMENT LEARNING**

## **BINNEN HVAC-WERELD**

# STAGEVERSLAG VOOR FONTYS HOGESCHOOL ICT

**Gegevens student:**

B.C. Verhaar  
406498

**Profiel/innovatiegebied:**

Applied Data Science

**Stageperiode datum:**

van 04-02-2020 t/m 21-06-2020

**Gegevens bedrijf:**

Facilicom Energiemanagement  
BC-IT development  
Van Leeuwenhoekweg 1, 5482 TK Schijndel  
Martijn Ploos van Amstel, Senior Software  
Engineer

**Gegevens docentbegeleid(st)er:**

John litsenburg - Fontys Hogescholen

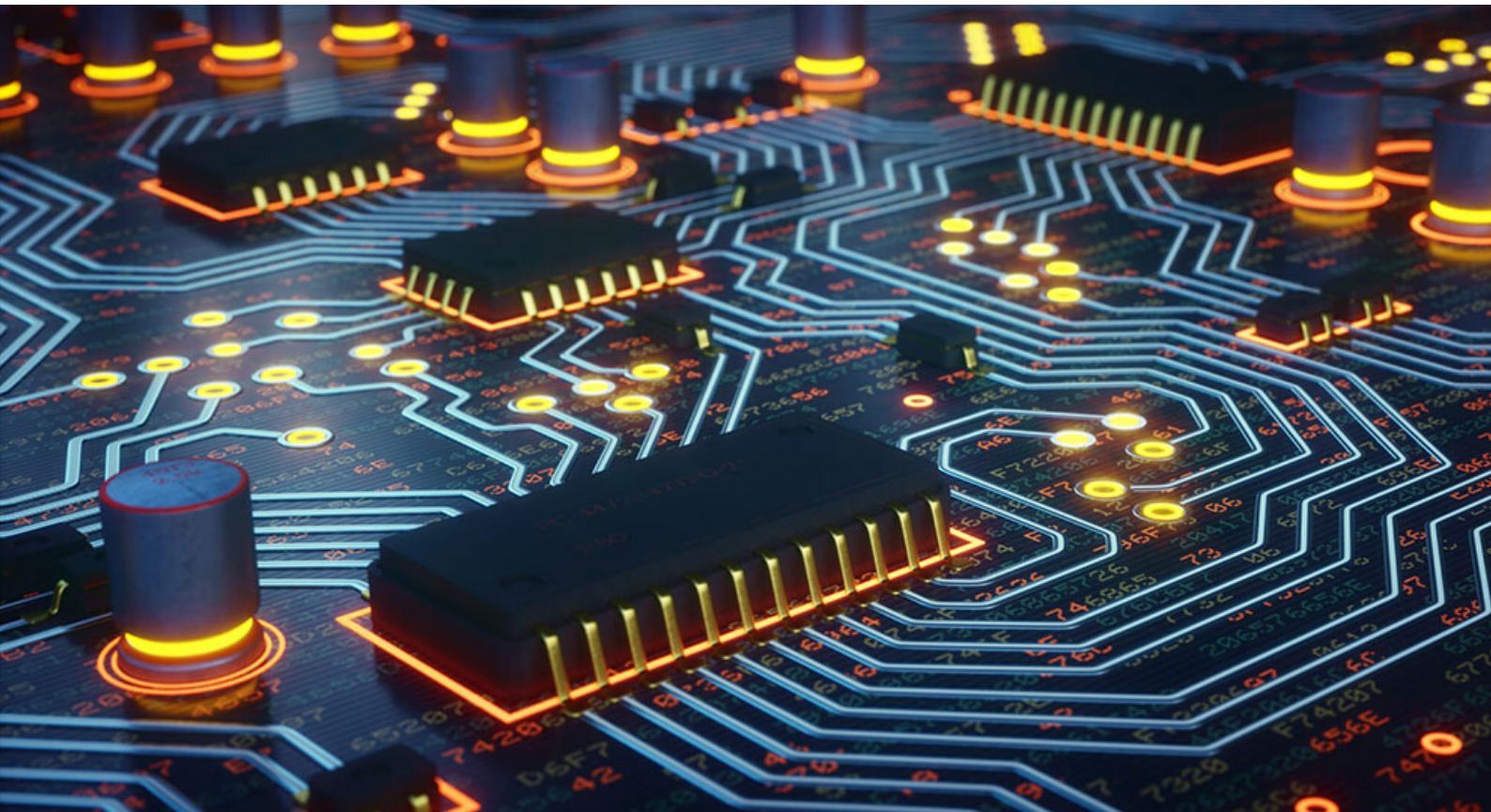
**Gegevens verslag:**

Reinforcement learning binnen HVAC-  
wereld  
21.06.20

**Getekend voor gezien door  
bedrijfsbegeleider:**

Datum:





# VOORWOORD

Dit verslag "Reinforcement learning binnen HVAC-wereld" is geschreven in het kader van mijn stage vanuit de opleiding ICT & Applied Data Science aan Fontys Hogeschool ICT. Vanaf begin februari 2020 tot en met eind juni 2020 ben ik bezig geweest met een onderzoek naar de mogelijkheden voor reinforcement learning in de Heating, Ventilation and Air Conditioning (HVAC) wereld.

Deze opdracht heb ik uitgevoerd voor Facilicom Energiemangement. Dit bedrijf is op meerder locaties gevestigd, maar ik ben voornamelijk bezig geweest in Schijndel (en thuis wegens de corona crisis...).

Graag zou ik iedereen willen bedanken die een bijdragen heeft geleverd aan deze stage! Een aantal van deze waren onmisbaar voor deze stage: Martijn Ploos van Amstel, Michel Maassen, Siem van den Reijen, Dirk de Laat, John Litsenburg, Chris Geene, Koen van Haren en Youri Bastiaan-Net.

Bastiaan Verhaar

# INHOUDSOPGAVE

<b>0 6</b>	
S A M E N V A T T I N G	
<b>0 8</b>	
01. I N L E I D I N G	
<b>0 9</b>	
02. H E T B E D R I J F	
<b>1 0</b>	
03. D E O P D R A C H T	
<b>1 1</b>	
04. R A N D V O O R W A A R D E N & R I S I C O ' S	
<b>1 3</b>	
05. G E B O U W B E H E E R S Y S T E M E N ?	
<b>1 5</b>	
06. R E I N F O R C E M E N T L E A R N I N G ?	
<b>2 2</b>	
07. C Y B E R - P H Y S I C A L S Y S T E M S ?	
<b>2 5</b>	
08. D E D A T A	
<b>2 6</b>	
09. H U I D I G E S I T U A T I E	
<b>3 2</b>	
10. I M P L E M E N T A T I E	
<b>3 5</b>	
11. D E P L O Y M E N T	
<b>3 6</b>	
12. R E S U L T A T E N	
<b>3 7</b>	
13. C O N C L U S I E & A A N B E V E L I N G	
<b>3 8</b>	
E V A L U A T I E	

# FIGURENLIJST

**Figuur 1.** Interface Priva BACnet. P13

**Figuur 2.** Voorbeeld Reinforcement learning (“Voorbeeld reinforcement learning”, z.d.). P14

**Figuur 3.** Formule linear algebra. P16

**Figuur 4.** Voorbeeld fully recurrent neural network (fdeloche, z.d.). P17

**Figuur 5.** Voorbeeld LSTM (Guillaume Chevalier, z.d.). P17

**Figuur 6.** De wiskunde achter de LSTM-cell (Wikipedia, 2020). P17

**Figuur 7.** Logo ML.NET (“Logo ML.Net”, z.d.). P16

**Figuur 8.** Logo Brain.js (“Logo Brain.js”, z.d.). P16

**Figuur 9.** Logo TensorFlow (“Logo TensorFlow”, z.d.). P17

**Figuur 10.** Formule temperatuur benadering. P22

**Figuur 11.** Plattegrond begane grond Geyssendorfferweg 5 te Rotterdam. P22

**Figuur 12.** Schematische tekening installatie meldkamer Breijer. P22

**Figuur 13.** Verbruik kantoren maand Februari 2020 in MW.h. Verwarmen (paars) & koelen (blauw). P23

**Figuur 14.** Schematische weergaven energieopwekking pand Rotterdam. P24

**Figuur 15.** Verbruik kantoren maand Maart 2020 in MW.h. Verwarmen (paars) & koelen (blauw). P24

**Figuur 16.** Temperatuur meldkamer (blauw) vs buiten (paars) op 09-03-2020. P25

**Figuur 17.** Plattegrond begane grond pand Rotterdam, rode cirkel Breijer meldkamer. P25

**Figuur 18.** Temperatuur kantoor BC-IT (blauw) vs sturing koeling per 10%-punten (paars) vs sturing verwarmen per 10%-punten (rood) op 06-05-2020. P25

# SAMENVATTING

Facilicom Energiemanagement (FEM) is een bedrijf opgericht in 1997 dat zich focust om panden van andere bedrijven aan de wettelijke verplichtingen te helpen voldoen. Hiervoor heeft FEM al meerdere tools, maar ze willen met deze stage onderzoeken of dat een machine learning tool binnen hun portfolio past.

Voor dit onderzoek is een onderzoeksvraag opgesteld: "Is het mogelijk, om met een reinforcement learning algoritme, een optimaal klimaat te behouden terwijl we beter omgaan met de energie binnen een gebouw?". Hierbij zijn deelvragen opgesteld om dit ingewikkelde onderzoek makkelijker te maken.

Een Gebouw Beheer Systeem (GBS) is een systeem waarmee je alle componenten in een gebouw kan monitoren en beheren. Samen met het BACnet protocol biedt dit de basis waarmee we data kunnen verzamelen en acties van het model kunnen uitvoeren.

Reinforcement learning is een van de drie pilaren binnen machine learning. Het is een methode waarmee je zonder data, maar met (spel-)regels, een model kan leren keuzes te maken, door het gebruik van straf- of beloningspunten. De motor die deze beslissingen maakt, heet een Neural Network (NN). Hier zijn veel verschillende varianten van.

Zo heb je een klassiek NN, dat gebruikt is in de eerste versie van de applicatie, een Recurrent NN (RNN), dat gebruikt is in de tweede versie van de applicatie. Met de RNN waren we in staat om historische data te kunnen verwerken.

Voor reinforcement learning is het belangrijk om een omgeving te hebben waar het in kan spelen/leren. Hierom is tijdens de stage een Cyber-Physical System (CPS) gemaakt. Deze had eerst een wiskundige benadering voor de actuele temperatuur, maar in een latere versie is dit vervangen met een ML model van Folmer van Rijn.

Na al dit onderzoek was het tijd voor de verkennende gegevensanalyse. Tijdens deze fase zijn er een aantal fouten uit het GBS gehaald, hiermee is de energieconsumptie met 57% gedaald. Helaas kwamen we halverwege maart 2020 erachter data sommige data in de database niet overeenkwam met de daadwerkelijke data. Hiervoor was een grote operatie vereist om dit weer recht te trekken en was pas begin mei opgelost.

Hierna is het model gemaakt met de data uit het GBS. Hiervoor zijn 2 verschillende modellen gemaakt. Een voor het CPS en een voor de Bluebot. Het resultaat van de CPS was zeer positief, helaas kon het resultaat van de Bluebot niet getest worden.

# SUMMARY

Facilicom Energy Management (FEM) is a company founded in 1997 that focuses on helping buildings belonging to other companies meet the legal obligations. FEM already has several tools for this, but they want to use this internship to investigate whether a machine learning tool can be added to their portfolio.

For this research, a research question has been formulated: "Is it possible, with a reinforcement learning algorithm, to maintain an optimal climate while we better use the energy within a building?". Sub-questions have been drawn up to make this complicated research easier.

A Building Management System (BMS) is a system with which you can monitor and manage all components in a building. Together with the BACnet protocol, this provides the basis for collecting data and executing model actions.

Reinforcement learning is one of the three pillars of machine learning. It is a method with which you can learn to make choices without data, but with (game) rules, by using penalty or reward points. The engine that makes these decisions is called a Neural Network (NN). There are many different variations of this.

For example, you have a classic NN used in the first version of the application, a Recurrent NN (RNN) used in the second version of the application. With the RNN we were able to process historical data.

For reinforcement learning it is important to have an environment where it can play / learn. This is why a Cyber-Physical System (CPS) was created during the internship. At first it had a mathematical approach for the current temperature, but in a later version this was replaced with an ML model by Folmer van Rijn.

After all this research, it was time for the exploratory data analysis. During this phase, a number of errors were removed from the BMS, reducing energy consumption by 57%. Unfortunately, in mid-March 2020, we found out that some data in the database did not match the actual data. This required major surgery to rectify this and was only resolved in early May.

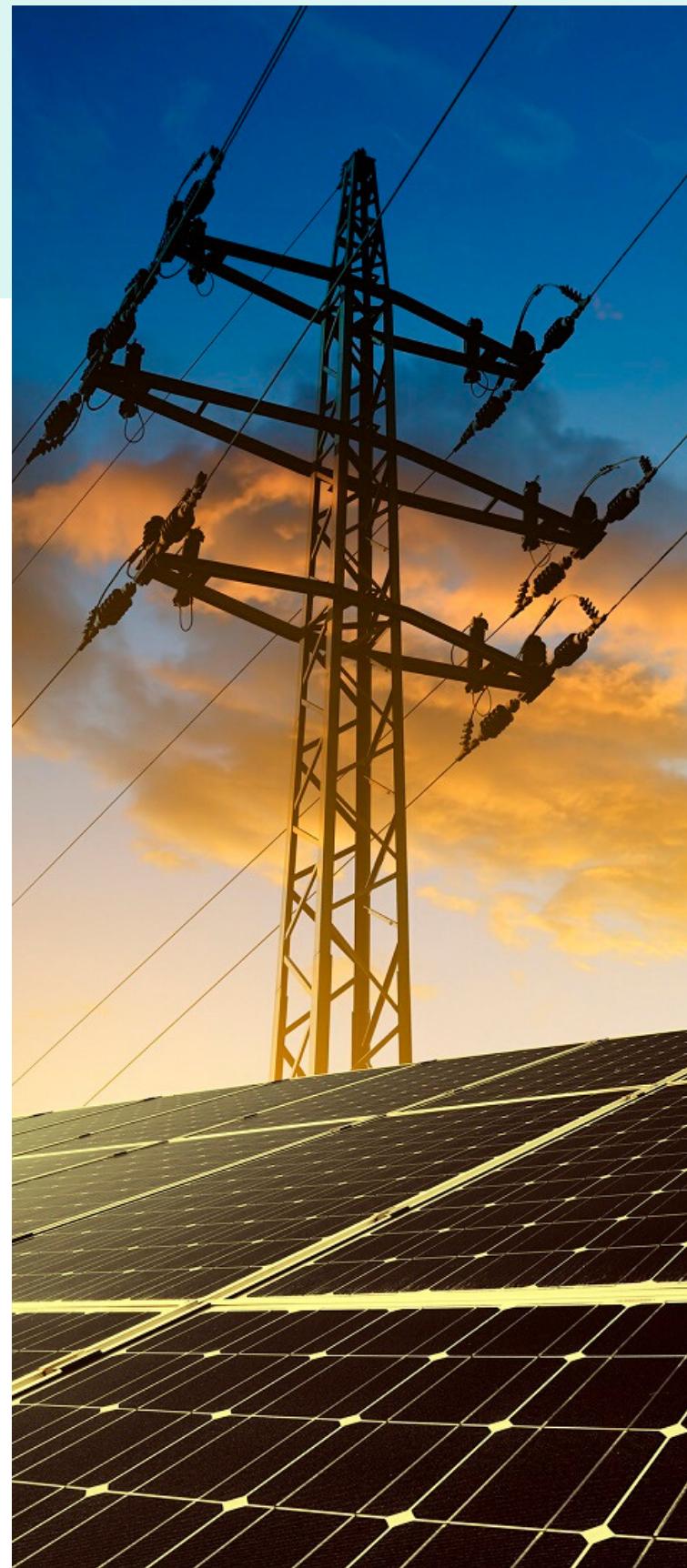
After this, the model was made with the data from the GBS. 2 different models were made for this. One for the CPS and one for the Bluebot. The result of the CPS was very positive, unfortunately the result of the Bluebot could not be tested.

# 01. INLEIDING

Laten we beginnen met iets over mijzelf. Mijn naam is Bastiaan Verhaar. Op het moment van schrijven studeer ik ICT op Fontys Hogescholen te Eindhoven. Mijn vader (Hendrik Verhaar) is Senior Software engineer bij Webeasy b.v. Webeasy is, na Priva, marktleider op het gebied van Gebouw Beheer Systemen en hier is ook mijn interesse in het vak gewekt.

De stageopdracht was kijken naar een geautomatiseerde methode om energie te besparen voor Facilicom Energiemanagement (FEM) door gebruik van Reinforcement Learning (RL).

FEM helpt organisaties te voldoen aan hun wettelijke verplichtingen, de CO<sub>2</sub> uitstoot te reduceren en dat alles budgetneutraal. Hier zijn een hele hoop methodes voor, maar FEM heeft hier 4 stappen voor bedacht: monitoren, energie besparen, vergroenen en zelfvoorzienend.





## 02. **HET BEDRIJF**

Facilicom energiemanagement (FEM) is een bedrijf dat in het leven is geroepen om aan de vraag van bedrijf te kunnen voldoen als het gaat om wettelijke verplichtingen van panden. Hierbij kan je denken aan de informatieplicht, energielabels, etc.

Dit bedrijf was eerst bekend als Ploos van Amstel Milieu Consulting in 1997 opgericht door Joost Ploos van Amstel, maar rond 2005 stond dit bedrijf bekend als Ploos energiestrategie. In 2015 is dit bedrijf opgekocht door Facilicom Group en is de naam veranderd naar de huidige naam: FEM. Het doel van FEM is om de energietransitie voor coöperaties te vermakelijken met behulp van slimme oplossingen

*“Samen met u werken wij aan een CO<sub>2</sub> neutrale organisatie” - FEM*

Een van deze oplossingen heet: EASy (Energie Afreken Systeem). Met deze oplossing kunnen organisaties ervoor zorgen dat de eindgebruiker de juiste energiefactuur ontvangt en heeft de organisatie een mooi overzicht van alle gestuurde facturen.

FEM wilt natuurlijk nog meer slimme oplossingen toevoegen aan haar portfolio en wilt hiermee de competentie voor te blijven

# 03.

## DE OPDRACHT

De onderzoeksraag luidt als volgt: Is het mogelijk, om met een reinforcement learning algoritme, een optimaal klimaat te behouden terwijl we beter omgaan met de energie binnen een gebouw? Hierbij is het dus van belang om het huidige klimaat, de temperatuur in een ruimte, gelijk te houden ten opzichten van nu. Daarnaast willen we ervoor gaan zorgen dat een gebouw minder energie gaat verbruiken, om zo een stap dichterbij de energietransitie van de rijksoverheid te komen (RVO, 2018).

Het onderzoek kan best complex worden en bevat heel veel onbekende elementen. Om deze reden zijn er deelvragen gemaakt om het proces te verduidelijken en versimpelen. Het onderzoek bevat 5 deelvragen:

- Aan welke randvoorwaardes moet dit systeem voldoen en wat zijn de risico's van dit systeem? (H4)
- Wat is een Gebouw Beheer Systeem? (H5)
- Hoe werkt een reinforcement learning algoritme en welke tools zijn er beschikbaar? (H6)
- Wat is een Cyber-Physical System en hoe werkt het? (H7)
- Wat is de huidige situatie? (H9)
- Hoe ziet de implementatie van de reinforcement learning eruit? (H10)
- Wat zijn de resultaten? (H12)

### Start stage

Voordat de stage was begonnen had Facilicom nog niets op het gebied van machine learning. Wel heeft Facilicom een aantal mensen rondlopen die mij op technisch ingezet kunnen ondersteunen. Een van deze mensen is Siem van den Reijen. Hij is een data scientist bij InSpark en wordt regelmatig ingehuurd door Facilicom om hun te helpen met hun data analyses. Verder bieden Dirk de Laat en Michel Maassen mij vakinhoudelijke ondersteuning. Hierbij moet je denken aan technische inzichten over HVAC of GBS.

### Doelen

Een van de doelen van deze opdracht is het beoordelen of dit een oplossing kan bieden voor energie optimalisatie van een gebouw. Dit is het voornaamste doel van FEM, reductie in energieverbruik. Hierbij kan deze oplossing ook nog een andere winst opleveren: De regelsoftware voor Gebouw Beheer Systeem (zie meer in hoofdstuk 5) hoeft niet meer geschreven te worden. Theoretisch zou GBS'en inregelen overbodig worden door het gebruik van ML, dit scheelt FEM heel veel manuur schelen en geld.

# 04.

## RANDVOORWAARDEN & RISICO'S

### RANDVOORWAARDEN

Voor ieder project is het van belang om van te voren duidelijk te hebben waar het product aan moet voldoen. Met deze lijst van eisen, ook wel randvoorwaarden, kan je voorkomen dat er communicatie fouten optreden en wordt er een product opgeleverd wat voldoet aan de verwachtingen van de andere kant.

Om deze reden is gelijk aan het begin van de stage de randvoorwaarden opgesteld. Hieronder is deze lijst weergegeven:

- Het model moet geplaatst kunnen worden in een NodeJS applicatie. Het communicatie middel dient MQTT of REST zijn
- Het onderdeel waar het model getraind wordt, moet centraal gebeuren. Er is één centrale plek waar Facilicom een server klaar heeft staan voor dit doeleinde. Hierdoor kunnen ze alle andere onderdelen in het proces goedkoper houden. Dit dient binnen .Net core of NodeJS te gebeuren
- Tijdens het trainen van het model, dient de temperatuur binnen conform-zone meer beloont te worden dan energieverbruik gestraft wordt.





## RISICO'S

Wanneer een proces geautomatiseerd wordt, vervallen vaak risico's maar er komen ook weer risico's bij. Hierom is het van belang om deze nieuwe risico's in kaart te brengen en te minimaliseren, zo niet te elimineren.

- Het risico dat het model een verkeerde keuze maakt. De impact van zo'n soort fout in deze scenario zou resulteren in een te koude of een te warme ruimte of te veel energie verbruik. Dit risico wordt geminimaliseerd door een dataset te hebben (dat divers is). Veel verschillende en uitwijkende data dat de realiteit weergeeft. Volledige eliminatie van dit risico is helaas niet mogelijk

- Het risico dat het model een verkeerde keuze maakt doordat de installatie in het gebouw is aangepast of niet goed is ingeregeld. Wanneer een installatie niet goed is ingeregeld, dient er een monteur gestuurd te worden om dit te herstellen. Maar wanneer een installatie veranderd, dient het model opnieuw in geleerd worden met de veranderde parameters.

## 05. GEOUW BEHEER SYSTEMEN?

Een Gebouw Beheer Systeem (GBS) is een methode voor het toepassen van domotica (Latijns: Domos -> Huis, tica -> toegepaste wetenschap). Een definitie van domotica is: De integratie van technologie en diensten, ten behoeve van een betere kwaliteit van wonen en leven. Een GBS is gespecialiseerd in het beheren van het klimaat in een gebouw. De definitie van klimaat ligt in deze context anders dan je misschien gewend ben: Klimaat is hier alles wat jouw welzijn beïnvloed. Hierbij kun je denken aan airconditioning, verwarmen, koelen, maar bijvoorbeeld ook verlichting en zonwering.

Wanneer een gebouw opgeleverd wordt, zijn deze systeem (meestal) goed ingeregeld. Door de tijd heen, gaan er kleppen, sensoren of andere componenten kapot. Deze situaties zorgen vaak voor klachten, waardoor een installateur ingeschakeld moet worden. De installateur kijkt meestal niet goed naar de oorzaak van het probleem, maar gaat bijvoorbeeld de stooktemperatuur verhogen. Een makkelijke oplossing, want er komen geen koud klachten meer, maar hierdoor gaat het gebouw wel meer energie verbruiken.

Een opstapeling van dit soort situaties zorgt ervoor dat een GBS compleet ontregeld raakt en niet meer functioneert naar behoren. Dit is waar de meerwaarde van deze stage naar voren komt. Dit systeem gaat ervoor zorgen dat de vraag van warmte/koud optimaal is, om energie consumptie minimaal te houden.

Facilicom is bezig met een ander project waarmee ze fouten vroegtijdig kunnen detecteren in een GBS. Zo maken ze een logicaset voor iedere installatie, wat zal controleren of iedere stookwaarde goed ingeregeld is, of de afgifte van warmte goed verloopt, etc. Hierdoor hoeft dit project geen rekening te houden met de scenarios zoals eerder omschreven en mogen we er vanuit gaan dat alle installaties optimaal kunnen functioneren.



Figuur 1. Interface Priva BACnet

## Nu is de volgende vraag, **hoe gaan we data beschikbaar maken vanuit het GBS en hoe kunnen we data schrijven naar het GBS?**

Het antwoord op deze vraag is relatief simpel: BACnet (Building Automation Control network). Dit is een protocol en kent vele vormen, waaronder serieel, met RS-232 of RS-485, UDP, een internet techniek, en ARCNET. Met BACnet is het mogelijk om data via een subscription methode, ook wel Change on Value (COV) genoemd, te ontvangen of simpelweg met een read actie vanuit het GBS. Daarnaast is het mogelijk om data te schrijven naar het GBS met dit protocol (Wikipedia, 2018). Dit protocol gaat als basis dienen voor het ontvangen van data en het sturen van het GBS.

Om ervoor te zorgen dat de juiste data beschikbaar wordt gesteld vanuit BACnet, hebben Michel Maassen en ik een Priva interface gemaakt, zie *Figuur 1*. Dit interface dient als gateway voor het model om data op te ontvangen en te schrijven, door middel van BACnet, maar hiernaast heeft deze interface een hogere prioriteit dan de regelstrategie geprogrammeerd in het GBS.

Dit laatste is belangrijk; een GBS heeft altijd een regelstrategie, in andere woorden hoe wordt een installatie aangestuurd. Neem een vloerverwarming, je zou ervoor kunnen kiezen om deze te regelen op basis van de ruimte temperatuur, om zo een minimale temperatuur te garanderen, maar je zou er ook voor kunnen kiezen om de vloerverwarming te laten regelen op basis van de stooklijn, bijvoorbeeld van de CV ketel.

Door de nieuwe interface is het van belang dat deze regelstrategie niet gevolgd wordt, omdat het Machine Learning model dit voortaan gaat bepalen. Maar de regelstrategie zal wel blijven bestaan als fail-safe, als er een communicatiefout optreedt bijvoorbeeld.

## Conclusie

Een GBS is een systeem dat in staat is om HVAC componenten, zoals CV-ketel, airconditioning, etc, te besturen / monitoren. Een GBS kan op zijn beurt ook weer bestuurd worden door het BACnet protocol. Dit onderzoek is gedaan met hulp van een literatuuronderzoek, interview met een expert (Michel Maassen) en prototyping (Priva BACnet interface).

# 06.

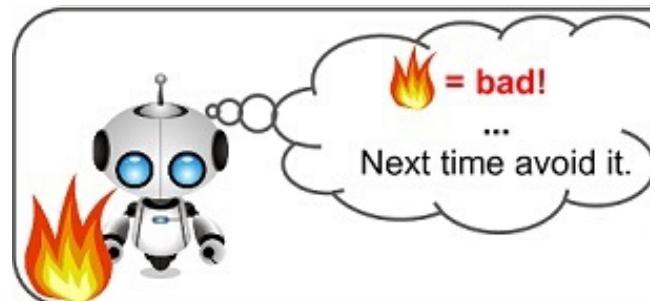
## REINFORCEMENT LEARNING?

Wat is nu reinforcement learning? Wanneer zou je het moeten gebruiken? Waar wordt reinforcement learning al toegepast? Al deze vragen zullen beantwoord worden in dit hoofdstuk.

De eerste vraag: **Wat is nu reinforcement learning?** Reinforcement learning (RL) is het principe van een agent in een omgeving dat al lerend acties kan ondernemen. Een klassiek voorbeeld van RL is een kind dat Super Mario Bros speelt. Een kind weet dat hij ongeveer zeven acties kan uitvoeren: Naar links (rennend), naar rechts (rennend), springen en item oppakken/gooien. Hoe eerder hij bij de finish is, hoe meer punten hij krijgt, maar hij moet oppassen dat hij niet doodgaat, want dan verliest hij al zijn punten en moet hij opnieuw beginnen. Met deze principes leert een kind Super Mario Bros spelen en is uiteindelijk in staat om het spel uit te spelen.

“Reinforcement learning differs from supervised learning in not needing labeled input/output pairs presented, and in not needing sub-optimal actions to be explicitly corrected” - [wikipedia.org](https://en.wikipedia.org)

Het idee achter reinforcement learning is ongeveer hetzelfde. Je hebt een agent (het kind), een virtuele omgeving (Super Mario Bros), acties (lopen, rennen, springen, ect) en beloningen (punten). De agent bevat alle intelligentie. Deze intelligentie is meestal in de vorm van een Neural Network (NN). Dit NN wordt getraind wanneer de agent in de virtuele omgeving aan het "spelen" is.



Figuur 3. Voorbeeld Reinforcement learning

Het doel van de agent is om zoveel mogelijk punten te verzamelen. Om ervoor te zorgen dat de kans hiervoor zo groot mogelijk is, moet er nog een concept geïntroduceerd worden: Exploration. Soms is het goed om een random actie te doen, op deze manier kan de agent nieuwe manieren proberen en kijken of hij meer punten van deze actie krijgt of juist niet. Op deze manier kan de agent echt leren, want je leert niet door alles maar op een manier te doen.

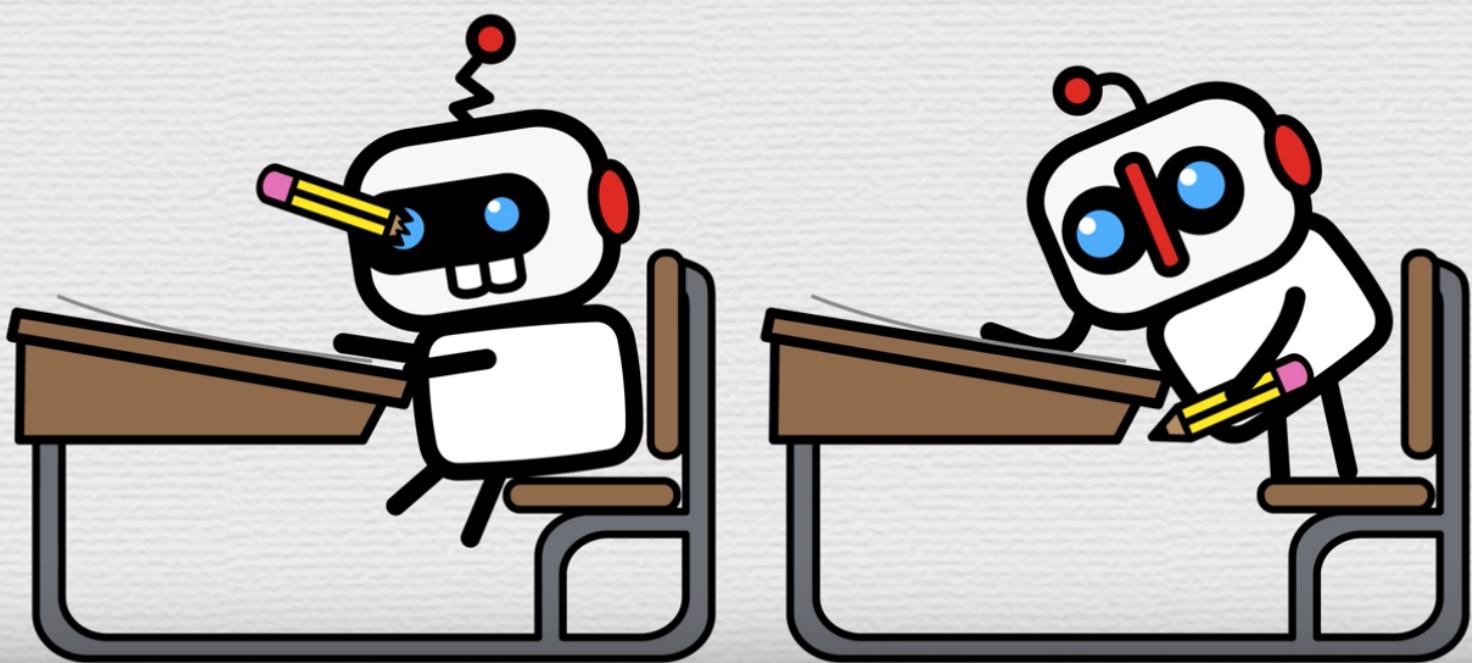
**Wanneer gebruik je RL?** RL kan je perfect toepassen als je een virtuele omgeving of representatie hebt van het probleem dat je wilt oplossen. In andere woorden: Je hebt iets nodig waarop je interactie kan uitvoeren dat resulteert in een aanpassing van de omgeving.

Een mooi voorbeeld hiervan zijn video games. Stel je hebt zojuist een vijand verslagen, dan heb je de omgeving veranderd. Deze verandering zou je kunnen belonen of straffen. Een relevanter bijvoorbeeld zou zijn: een gebouw sturen voor optimaal verbruik zonder comfort te verliezen, ook wel deze stage.

De laatste vraag: **Waar wordt RL al toegepast?** Een van de bekendste voorbeelden van RL is AlphaGo. Dit is een model dat de beste Go spelers verslagen heeft: zoals Fan Hui (in 2015) of Lee Sedol (in 2016).

Maar volgens een onderzoek van Towards Data Science wordt reinforcement learning op veel meer plekken toegepast. Een paar voorbeelden zijn: Sturing van verkeerslichten, de robotics, maar ook in de scheikunde om chemische reacties te simuleren (Towards Data Science, 2018).

Om dit onderwerp af te ronden, is er een klein onderzoek nog gedaan naar de technieken en wiskunde achter RL. Deze is bijgevoegd in *Bijlagen D*.





## WAT ZIJN NEURAL NETWORKS?

Een Neural Network (NN) is geïnspireerd door het neuron systeem van het menselijk brein. Je hebt een reeks aan neurons die aan elkaar verbonden zitten en samen als geheel kunnen ze van de ingevoerde waardes een antwoord vormen.

Binnen een NN, heb je inputs (de invoer waardes), output (het resultaat uit het NN), weights en biases. De berekeningen gaan via matrixen (Wikipedia, 2020), die lineaire vergelijkingen oplossen. De weights representeren de multiplicatie (a) en de biases de additie (b) van de formule (Wikipedia, 2018a).

$$y = a^*x + b$$

Figuur 3. Formule linear algebra

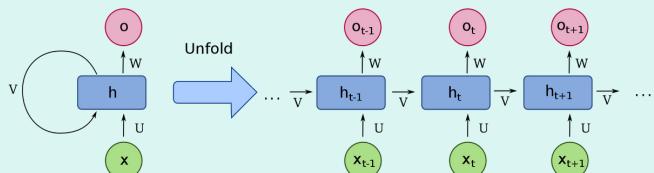
Als laatste hebben we nog bovenop iedere lineaire vergelijking een activatie functie. Deze activatie functies zijn de uitkomst van de neuron. Er zijn een hele hoop verschillende soorten activatie functies, van lineaire, tot niet-lineaire. Een aantal voorbeelden hiervan zijn: ReLU, Tanh, Arctan (Towards Data Science, 2017).

De lineaire vergelijking en de activatie functies samen is de wiskundige vertaling van een NN, waarbij de matrix-vergelijkingen de koepel is voor efficiënte calculaties.

## RECURRENT NEURAL NETWORKS

Een NN is geschikt voor een hele hoop verschillende problemen, van classificatie tot actie bepalingen. Maar een onderdeel waar een NN niet mee om kan gaan, is historische of sequentiële data, in andere woorden: trends. **Is er een manier om NN's te gebruiken met deze type data?** Hiervoor hebben wetenschappers een ander variant NN bedacht: Een Recurrent Neural Network (RNN).

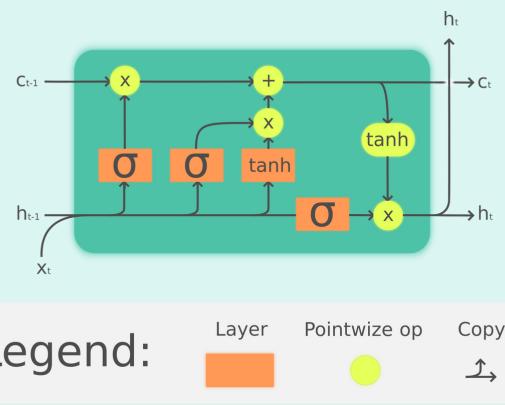
Voor deze variant van NN geef je historische data mee om efficiënte trendanalyses te kunnen doen. In *Figuur 4*, is een voorbeeld gegeven van een fully recurrent NN. Hier is te zien dat iedere neuron data van de vorige berekening meeneemt. Doordat deze wordt meegegeven en er andere berekeningen gebruikt worden in de Neuron, zie *Figuur 6*, is dit type NN in staat om patronen over de tijd te detecteren.



Figuur 4. Voorbeeld fully recurrent neural network

In 1997 is een nieuwe vorm van RNN geïntroduceerd, genaamd: Long Short-Term Memory (LSTM). Dit type RNN is uitgevonden door Sepp Hochreiter and Jürgen Schmidhuber en is op het moment de meest gebruikte vorm voor RNN's (Brownlee, 2019).

Verder zijn er nog veel meer types RNN: van fully recurrent tot Gated recurrent unit met ieder zijn eigen voor- en nadelen. Maar omdat, volgens het bovenstaande artikel, met een LSTM het vaak meeste succes geboekt wordt, is er gekozen voor dit type RNN voor deze stage.



Legend:

	Layer
	Pointwise op
	Copy

Figuur 5. Voorbeeld LSTM

$$\begin{aligned}
 f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
 \tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
 h_t &= o_t \circ \sigma_h(c_t)
 \end{aligned}$$

Figuur 6. De wiskunde achter de LSTM-cell

De werking van een LSTM-cel is weergegeven in *Figuur 5*. Zo'n soort cel werkt als volgt: Een cel heeft drie inputs, 2x historisch en 1x de actuele data. Verder bevat een cel drie "gates": een Forget-, input- en output-gate. Deze drie gates zijn weergegeven in *Figuur 5* als de verticale lijnen.

De input-gate is ontworpen om de historische data te kunnen combineren met de nieuwe data. Hiermee is de LSTM-cel in staat om tijd gevoelige patronen in data te herkennen. De output-gate is ontworpen door te linken naar de volgende laag in het NN of dient als uitkomst van het NN. De forget-gate dient voor een zeer bijzondere functie. Een van de grote problemen bij RNN's is het netwerk dingen "vergeet" of juist "overdrijft", ook wel het "Vanishing gradient problem" genoemd.

De forget-gate voorkomt dit probleem door de nieuwe data te verwerken in de "cell state". Deze cell state wordt blijft opgeslagen zodat de nieuwe set berekeningen niet vergeet om te gaan met data die hij in het begin heeft verkregen (Colah, z.d.).



## WELKE TOOLS ZIJN ER VOOR RL?

Tools voor reinforcement learning. Er zijn ongelofelijk veel tools voor machine learning in alle verschillende programmeertalen. Maar we kunnen de zoektocht makkelijker maken door de randvoorwaarden erbij te pakken:

- Het model moet centraal getraind worden op een NodeJS of .Net applicatie
- Het doel moet on-site gedeployed kunnen worden op een NodeJS applicatie.

Deze voorwaarden zorgen ervoor dat er heel weinig tools overblijven. Het totale lijstje afgebakend tot de talen individueel is:

- ML.net (voor .Net)
- Brain.js (voor JS)
- Tensorflow (voor JS en Python)

### ML.net

Deze tool is ontwikkeld door Microsoft zelf voor haar eigen .Net omgeving. ML.net is een open source, cross platform framework voor al je machine learning problemen. Dit framework maakt gebruik van autoML. Dit is een ander framework van Microsoft waarmee je via de command line of via Visual Studio models kan genereren met 5 muisklikken!

ML.Net is een super handig framework voor .Net ontwikkelaars, maar daar ligt het grote probleem ook, alleen voor .Net ontwikkelaars... De models van ML.Net zijn alleen te gebruiken binnen een .Net applicatie en onze modellen dient gebruikt te worden in zowel een .Net als een NodeJS omgeving. Hierdoor is ML.Net afgevallen als bruikbare tool.



Figuur 7. Logo ML.NET

### Brain.js

Deze tool is speciaal gemaakt voor Neural Networks in de javascript taal. Het bevat een hoop wiskundige trucjes waardoor deze tool als volledige machine learning tool gezien kan worden. Zo kan je RL models maken met enkele regels aan code!

Voor deze tool geldt helaas dat het heel klein is, met een kleine actieve community. Dus om praktische redenen is deze optie vervallen



Figuur 8. Logo Brain.js

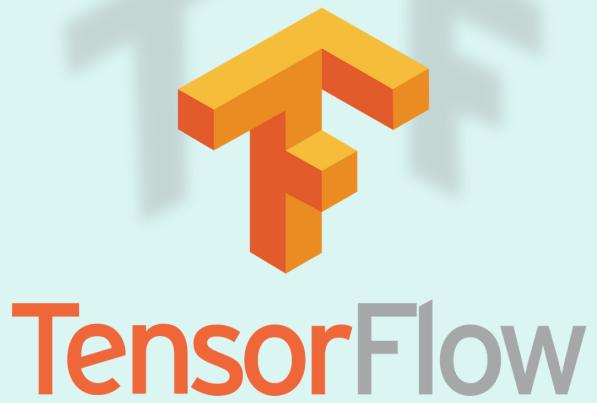
## TensorFlow

Deze tool is een open source, end to end machine learning framework, ontwikkeld door Google zelf. Het is een volledig machine learning framework met alle tools die je misschien nodig hebt en meer! TensorFlow is origineel geschreven in C++, wat gecompileerd is naar Python. Later heeft Google ondersteuning voor Javascript geschreven, waardoor het ook in de browser kan draaien.

Omdat deze library is geschreven door Google, zit er ook een grote community achter met actieve developement. Ook is het makkelijk om models te verplaatsen van een machine naar de andere, door het gebruik van Graph file. Deze bestanden kan je vergelijken met een JSON bestand. Wanneer je een Graph file maakt, bevat het de huidige staat van iedere Tensor. Hiermee heb je het complete netwerk inclusief de weights en biases.

## Keuze

Het is mogelijk om een model op het ene framework te trainen en vervolgens op een ander model te deployen. In theorie heb je ander de vorm van je neural network nodig, en alle weights & biases. Dit kost alleen heel veel extra tijd en energie om dit goed te doen, terwijl we een oplossing hebben waar dit allemaal geen probleem is.



Figuur 9. Logo TensorFlow

Dus omdat van uitgebreidheid van de tool en tijd is er dus gekozen voor het TensorFlow framework. In *Bijlagen B*, zit een onderzoek naar de werking van Tensorflow.

## Conclusie

RL is een type algoritme wat zonder data zelf kan leren. Voor het type probleem wat hier aangepakt wordt, is er gekozen voor een Recurrent Neural Network voor de beslissingsmechaniek en is TensorFlow de meest bruikbare tool, wanneer dit naast de randvoorwaarden neer gelegd wordt. Tijdens dit onderzoek is er een grote literatuur studie gedaan, is er een interview geweest met Siem van den Reijen (Inspark) en zijn er prototypes gemaakt.



## 07. CYBER-PHYSICAL SYSTEMS?

Een Cyber-Physical System (CPS) is een concept waarbij fysieke elementen gedigitaliseerd worden om interacties te kunnen simuleren. Het concept is heel groot en globaal. Hierom heeft de US National Science Foundation (NSF) dit concept geïdentificeerd als een belangrijk onderzoeksgebied. Beginnend in 2006 hebben ze meerdere onderzoekspreisen uitgereikt aan inzendingen van de NSF workshop (IEEE, 2007).

Omdat een CPS vrij globaal geïnterpreteerd kan worden, zal voor deze stage het concept afgebakend worden naar temperatuur representatie in een ruimte.

In een ruimte spelen er een aantal zaken:

- De verwarming kan aan (actieve verwarming)
- Er kan een airconditioning aanwezig zijn (actieve koeling)
- De ruimte verliest standaard warmte aan zijn omgeving (passieve koeling)
- De ruimte heeft een minimale temperatuur. Deze is altijd gelijk aan de temperatuur van zijn omgeving

Dit samen leidt tot een wiskundige weergaven van de temperatuur in een ruimte. Deze formule is weergegeven in Figuur 10.

$$\check{T} = T - a + \mu_h * (Th + a) + \mu_c * (a - Tc)$$

Figuur 10. Formule temperatuur benadering

**T̄:** Nieuwe temperatuur als resultaat van tijdsverandering

**a:** Standaard warme verlies aan de buitenomgeving

**T:** Huidige temperatuur

**Tr:** Minimale temperatuur in de ruimte

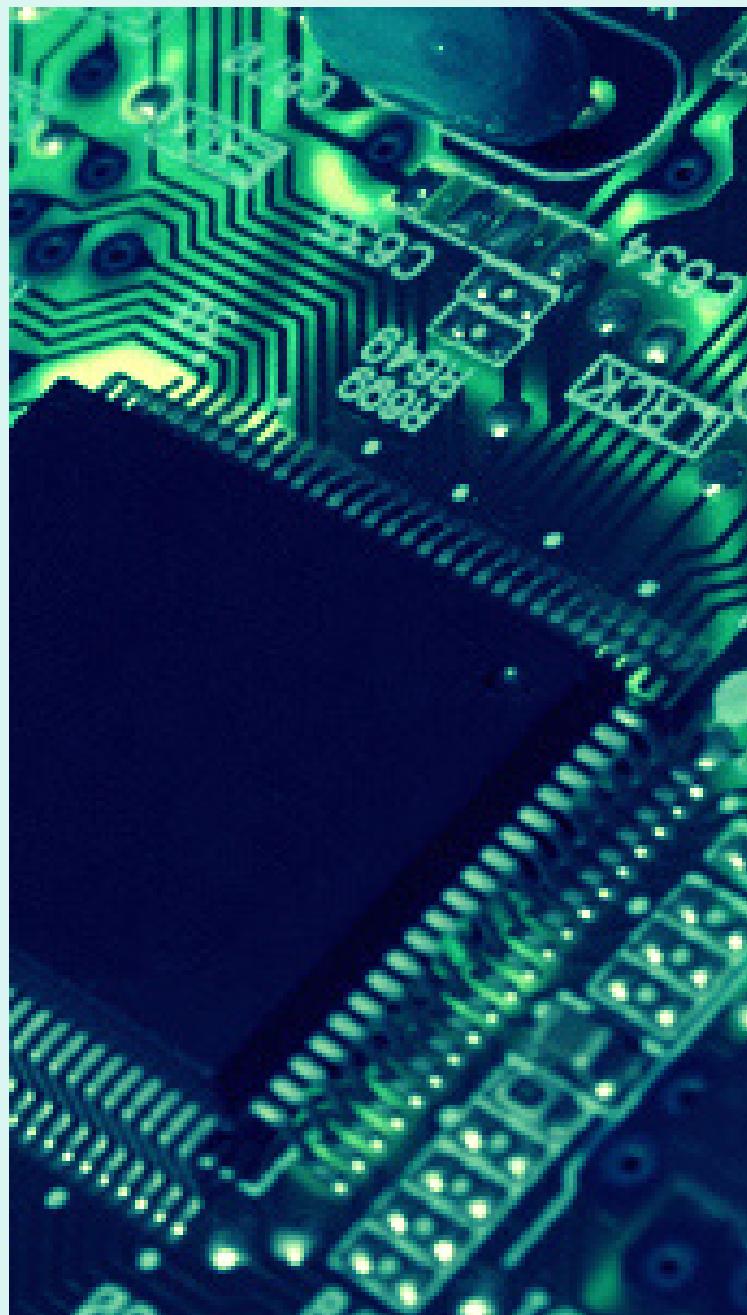
**Th:** Verandering in temperatuur door het inschakelen van de verwarming (bijvoorbeeld)

**Tc:** Verandering in temperatuur door het inschakelen van airconditioner (bijvoorbeeld en optioneel)

**μh:** Actie van de machine learning model m.b.t. verwarmen

**μc:** Actie van de machine learning model m.b.t. koelen

Met deze wiskundige formule is het mogelijk om de temperatuur in een ruimte te benaderen. Dit zal nooit een exacte weergaven kunnen worden, omdat dit een lineaire benadering is en te veel verschillende elementen aanwezig zijn die (nog) niet meegenomen zijn. Je kunt hierbij denken aan: ramen en deuren (die open kunnen staan), de zon die de ruimte opwarmt, etc. Wel is het mogelijk om al deze element kan mee te nemen in de formule. Dit is alleen buiten de scope van de stage.



**Is er een betere manier om de temperatuur te benaderen?** FEM heeft eerdere onderzoeken gedaan naar dit onderwerp. Een van deze onderzoeken is gedaan door een TU/E student, Folmer van Rijn. In zijn onderzoek stelt hij voor om machine learning te gaan gebruiken om de volgende ruimtetelemparatuur in het interval te voorspellen (van Rijn, 2018). Een super interessante thesis en zeker een aanrader om te lezen.

Een van de grootste problemen van deze aanpak is dat het ongelofelijk veel rekenkracht kost om deze aanpak nuttig te gebruiken. Niet alleen om te trainen, maar ook om te gebruiken. Omdat dit model voorspelt wat de volgende ruimtetelemparatuur zou zijn, was het vereist om een vorm van brute force te gebruiken om de meest optimale instelling te vinden. Hierdoor moest het gebruik van het model op een centrale plek gebeuren, wat ervoor zorgde dat het onderzoek niet voldeed aan de randvoorwaarden.

Dit onderzoek, van Folmer van Rijn, zou wel gebruikt kunnen worden binnen het CPS. Zo zouden we de wiskundige benadering voor temperatuursverandering kunnen vervangen met het onderzoek van Folmer van Rijn. De verwachting is dat het CPS een betere virtuele representatie kan bieden en er dus op die manier beter getraind kan worden.

Dit model gaat op precies dezelfde manier getraind worden als dat Folmer van Rijn voorgesteld in zijn onderzoek. Er gaat supervised machine learning gebruikt worden, alleen wel met andere data. Het onderzoek van van Rijn was gebaseerd op het B30 gebouw in Den Haag, voor het Centraal PlanBureau (CPB), en er was andere data beschikbaar voor de training. Zo was er in dat gebouw data beschikbaar over de bezettingsgraad van een ruimte, doordat er poortjes aanwezig waren. Dit is niet beschikbaar voor deze stage, maar was er wel een agenda beschikbaar per ruimte om de bezetting te kunnen benaderen. Een precieze omschrijving van de gebruikte data is te vinden in Hoofdstuk 08. *De data*.

## Conclusie

Een CPS is een virtuele representatie van de realiteit. De manier waarop dit gedaan wordt, is helemaal vrij. Hierom is tijdens de stage eerst gekozen voor een lineaire wiskundige benadering voor temperatuursverandering, maar later veranderd is naar een machine learning model. Voor dit onderzoek heeft er een literatuur studie, tinkering/prototyping en expert interview plaats gevonden (met Folmer van Rijn).



## 08. DE DATA

Een machine learning model kan maar net zo goed zijn als de data het krijgt. Alle data dat gebruikt is voor deze stage, komt uit drie verschillende bronnen: Priva history, KNMI en outlook agenda.

Priva history is de data zoals omschreven in hoofdstuk 09. *Huidige situatie*. Deze data omvat alles uit het GBS, bestaande uit ruimte temperaturen en klep standen. Het is mogelijk om daar meer data in te hebben, bijvoorbeeld CO<sub>2</sub> metingen of luchtvochtigheid, maar in deze installatie worden dit niet gemeten.

De KNMI data omvat data zoals: buitentemperaturen, relatieve luchtvochtigheid, windsnelheden, etc. Deze data is nodig om de effecten van het buitenklimaat te kunnen zien op de binnen temperatuur.

De outlook agenda is een bijzonder datapunt. Met deze data kunnen we in beeld brengen of de ruimte in gebruik is. Ook aanwezigheid in een ruimte heeft effect op de temperatuursveranderingen in een ruimte. Hier is wel een groot probleem mee, deze data is inaccuraat. Zo zou iemand wel de ruimte hebben gereserveerd, maar gebruikt hem vervolgens niet of andersom. Hier moeten we rekening mee houden.

In *bijlagen A* is een compleet overzicht te zien van alle beschikbare data.



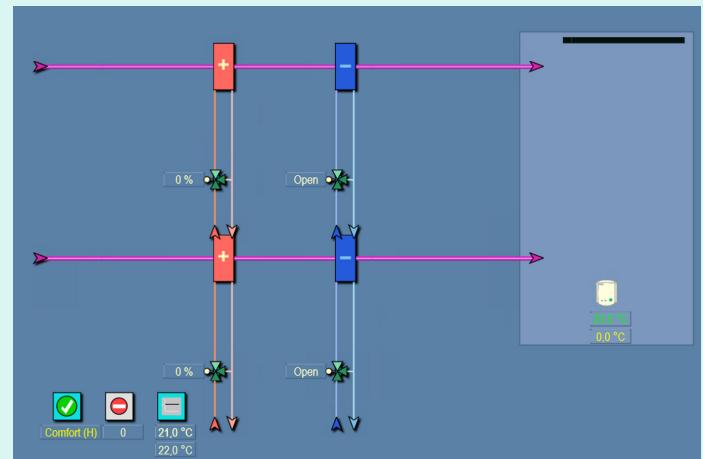
## 08. HUIDIGE SITUATIE

Het kantoor waar dit alles op getest is, is het hoofdkantoor van Breijer zelf. Dit kantoor is gevestigd op Geyssendorfferweg 5 te Rotterdam, gebouwd in 1959 en bevat 4826 m<sup>2</sup> aan oppervlakte op de begane grond (Kadaster, z.d.). Het is in beheer van de Breijer groep zelf, waar in 2006 een Priva HX GBS is geplaatst.

"Onder de vuurtoren schijnt geen licht". Dit is een gezegde waar de Breijer groep heel erg goed in is. Ze zorgen heel goed voor panden van klanten, maar niet haar eigen panden.

Dit is zeer goed te zien in de kantoren van dit pand. Breijer had het vermoeden dat dit niet helemaal lekker ging en heeft daarom mij een analyse laten doen, hiervoor heeft Breijer Priva History geactiveerd. Dit is een functionaliteit van Priva om meetgegevens in een MS-SQL database te krijgen. Deze gegevens zijn op basis van een interval en niet op basis van verandering. Dit interval staat op acht minuten en wordt eens per dag naar de database geüpload.

Dit systeem heeft vervolgens drie maanden aan data zitten uploaden naar deze database server. Aan de hand van deze gegevens is er een analyse gemaakt met Power BI (Verhaar, 2020).



# 21.4°C

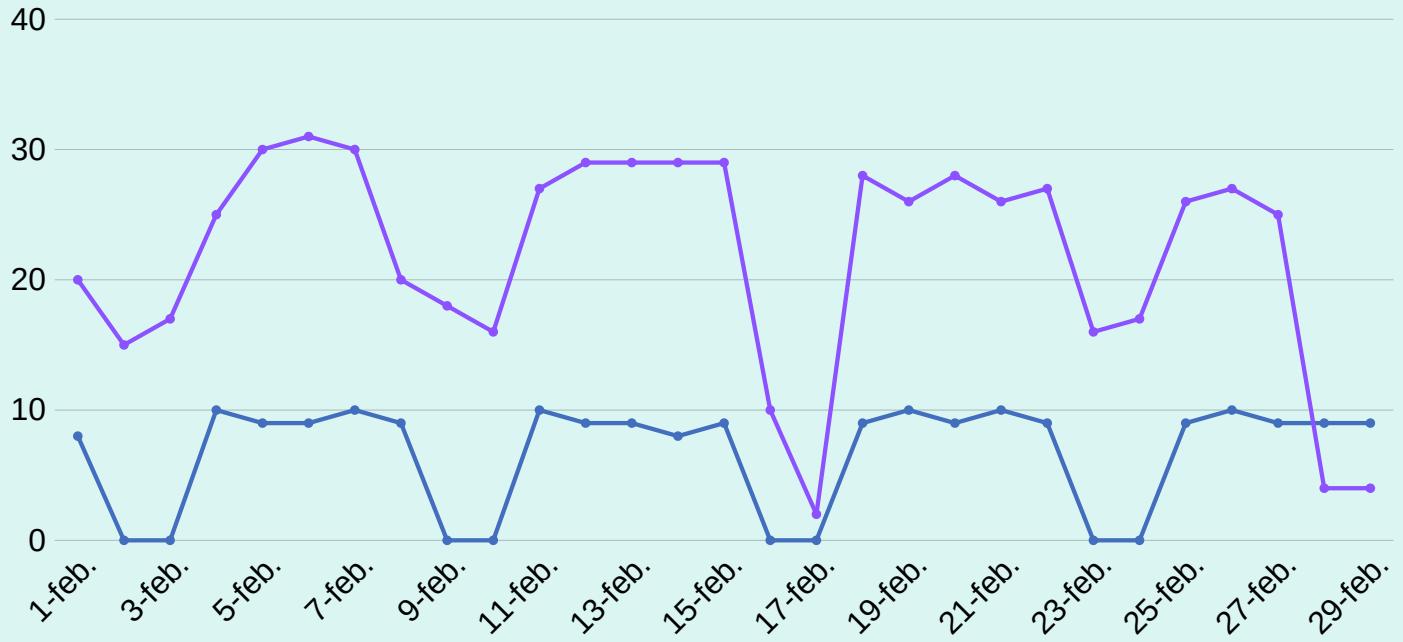
De gemiddelde temperatuur van alle ruimtes

# 373 KW.H

De gemiddelde verbruik per dag per ruimte

# €4.088

De gemiddelde prijs voor temperatuur bewaking per jaar per ruimte



Figuur 13. Verbruik kantoren maand Februari 2020 in MWh. Verwarmen (paars) &amp; koelen (blauw)

In Figuur 14 is het verbruik van de kantoren in het pand over de maand februari 2020 te zien in MWh. Je ziet heel duidelijk dat er een hoge vraag is naar warmte vanuit de kantoren, maar tegelijk draait het koelvermogen ook tegen de capaciteit aan.

Op een "top"-dag, 6 februari bijvoorbeeld, werd in deze periode 40 MWh aan energie verbruikt, puur en enkel aan klimaat / temperatuur bewaking in de kantoren. In het pand zijn 69 kantoren. Dit zou betekenen dat op een "top"-dag gemiddeld gezien ieder kantoor 579.7 KWh zou verbruiken. Dit is ongelofelijk veel, volgens Michel Maassen zou een verbruik van 200 KW.h mogelijk zijn voor deze kantoren.

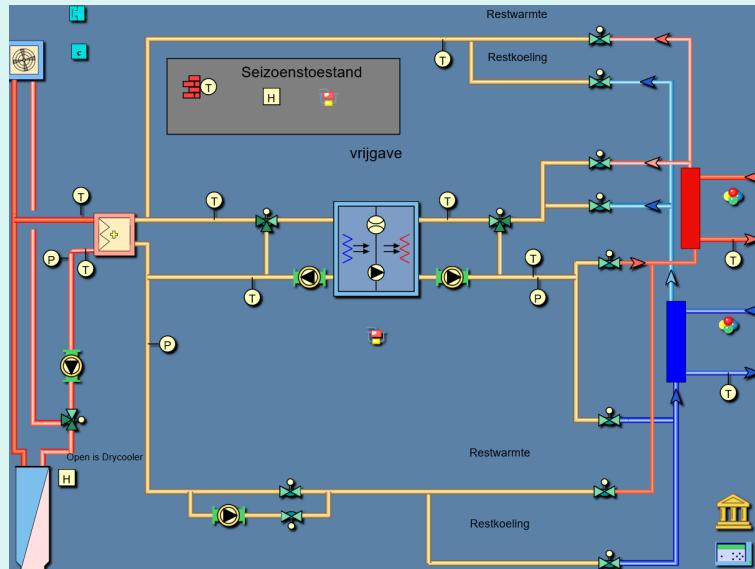
Het is mogelijk om dit verbruik uit te drukken in geld. Hiervoor geldt een warmteconversie met een factor 4 om het verbruik om te rekenen naar primaire energie (elektrische energie voor de warmtepomp in ons geval), voor iedere KWh primaire energie geldt een prijs van 12 cent. Voor onze "top"-dag zou dat een prijs opleveren van 69 euro per kantoor per dag.

Michel Maassen (teamleider Breijer Groep) vond dit veel te veel. Om deze redenen hebben we samen gekeken naar de installatie. Als we het hebben over een gebouw klimaat installatie, hebben we het over drie onderdelen: Opwekking, transport en afgifte.

**Opwekking** is het onderdeel wat verantwoordelijk is voor de opwekking van warmte of koud. Je kunt hier denken aan een warmtepomp, een CV-ketel, koelelement, etc.

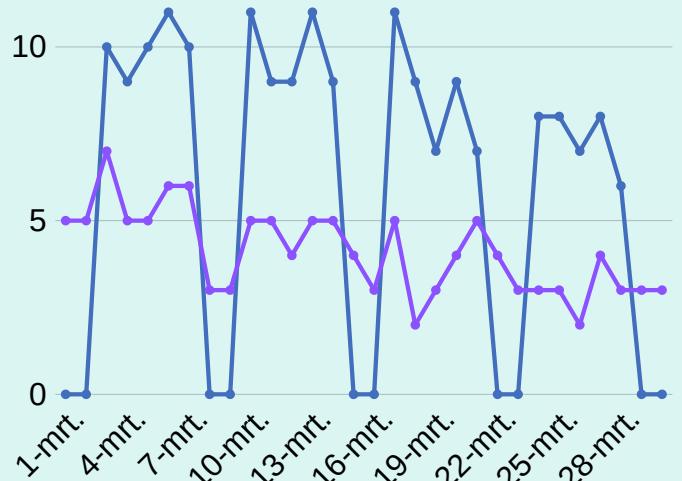
**Transport** is het onderdeel wat verantwoordelijk is voor de verplaatsing van deze opgewekte energie. Dit zijn bijvoorbeeld je leidingen in een gebouw. Als laatste heb je de **afgifte**. Dit zorgt ervoor dat je de opgewekte energie in een ruimte krijgt. Je kunt hierbij denken aan je radiatoren en luchtschachten.

Wanneer je zo'n soort installatie gaat analyseren, is het altijd een goed idee om te beginnen bij opwekking van je installatie.



Figuur 14. Schematische weergaven energieopwekking pand Rotterdam

15



Figuur 15. Verbruik kantoren maand Maart 2020 in MW.h. Verwarmen (paars) &amp; koelen (blauw)

Tijdens de analyse van de energie opwekking, kwamen we erachter dat de Omloop Vrije Koeling op handmatig open/aan stond. Dit gedeelte van de installatie is te zien in *Figuur 15* onderin. Hier zie je twee kleppen boven elkaar, waarbij een klep een pomp ervoor heeft. Deze omloop zorgt ervoor dat er geforceerd koelwater door de installatie kan rond gepompt worden. Omdat dit op handmatig open/aan stond, werd er continue koelwater rond gepompt waarvoor de CV-ketels moesten compenseren.

Een aanpassing voor deze fout was op 26 februari 2020 uitgevoerd en je zag meteen de dag erna een daling in totale verbruik. Deze daling was met een factor twee tot drie meetbaar. Waarbij het verbruik per kantoor per dag in de maand februari op 373 KW.h lag, lag het in de maand maart op 153 KW.h. Hierdoor dalen de jaarlijkse lasten per kantoor voor temperatuur bewaking naar 4.59 euro. Een daling van maar liefst 57.4%!

Het verdere verloop van de aanpassing is te zien in *Figuur 16*. Uit *Figuur 16*, hebben Michel Maassen en ik geconcludeerd dat het systeem beter functioneert en dat de rest van de winsten te behalen zijn in een betere / slimmere warmte / koud vraag vanuit de kantoren.

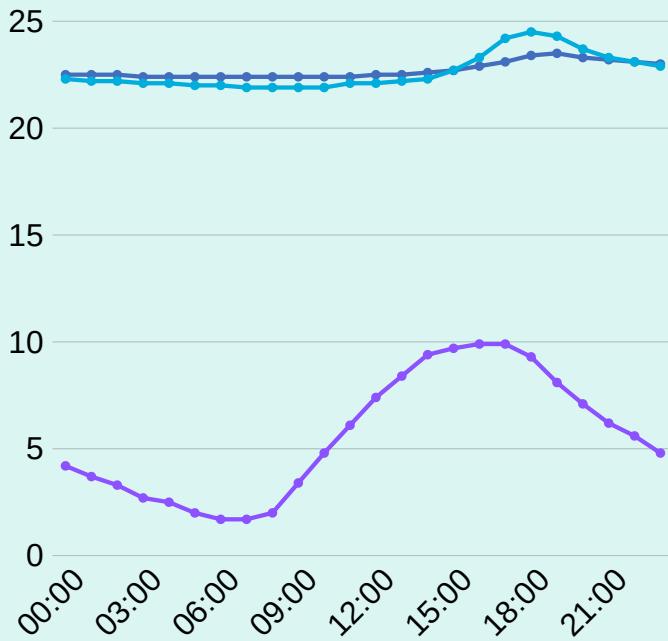
Halverwege maart is een fout in de database gevonden. Deze fout zorgde ervoor dat de data over de klep standen niet overeenkwam met de daadwerkelijk klep standen. Dit heeft heel veel tijd gekost om op te lossen, waardoor er pas begin mei de goede data verzameld kon worden. Pas na een 2 weken was er genoeg data om te beginnen met de modellen.

# 57.4%

*Daling in energieverbruik per kantoor. Van 373 KW.H naar 159 KW.H*

# €1.745

*De gemiddelde prijs voor temperatuur bewaking per jaar per ruimte. Daling vanaf €4088*



Figuur 16. Temperatuur kantoor BC-IT (donker blauw) vs buiten (paars) vs meetingruimte Maaszaal (licht blauw) op 22-03-2020



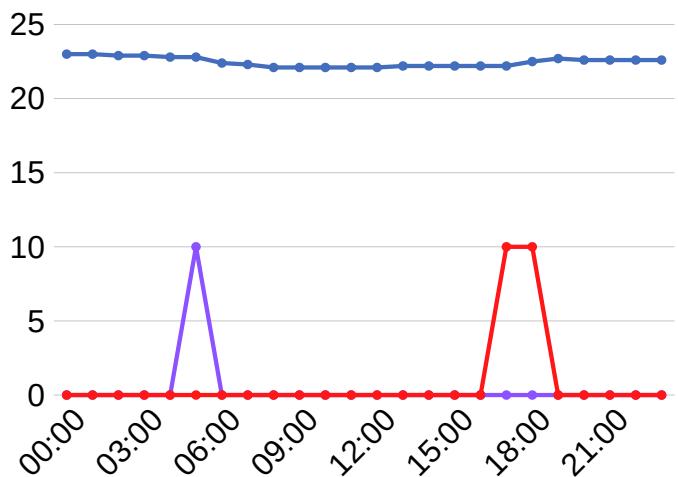
Figuur 17. Plattegrond tweede verdieping pand Rotterdam, rode cirkel kantoor BC-IT, blauwe cirkel meetingruimte Maaszaal

De volgende stap in de analyse was de vraag van energie (warmte of koude) vergelijken met het verloop van de temperatuur. De definitie van een optimale temperatuur hebben we vastgesteld op een temperatuur tussen de 20 en 21.5 graden Celsius, waarbij een temperatuur van 18 tot 20 graden Celsius acceptabel gezien wordt. In *Figuur 17* is het verloop weergegeven.

Een aantal conclusies kunnen hier getrokken worden:

- De ruimtes zijn goed geïsoleerd. Dit is zichtbaar doordat de temperatuur bijna niet veranderd in de avonduren. Het warmte verlies aan je omgeving en de veranderingen in buitentemperatuur zijn niet tot minimaal te zien zijn. In het kantoor van BC-IT is er geen verandering te zien, maar bij de meetingruimte Maaszaal zie je een kleine schommeling van 0.5 graden celsius. De Maaszaal licht aan de buitenrand van het pand, hierdoor is het gevoeliger voor veranderingen in buitentemperatuur (Zie *Figuur 18*).

- Om ruimtes te koelen heb je relatief weinig tijd nodig, dit is te zien in *Figuur 19*. Zo kan je het kantoor van het BC-IT team met een halfuur 0.5 graden celsius koelen. Ditzelfde geldt ook voor verwarmen. Ook hier kan je de ruimte met 0.5 graden celsius verwarmen in een halfuur. Dit zijn belangrijke gegevens om te weten voor het Cyber Physical System.



Figuur 18. Temperatuur kantoor BC-IT (blauw) vs sturing koeling per 10%-punten (paars) vs sturing verwarmen per 10%-punten (rood) op 06-05-2020

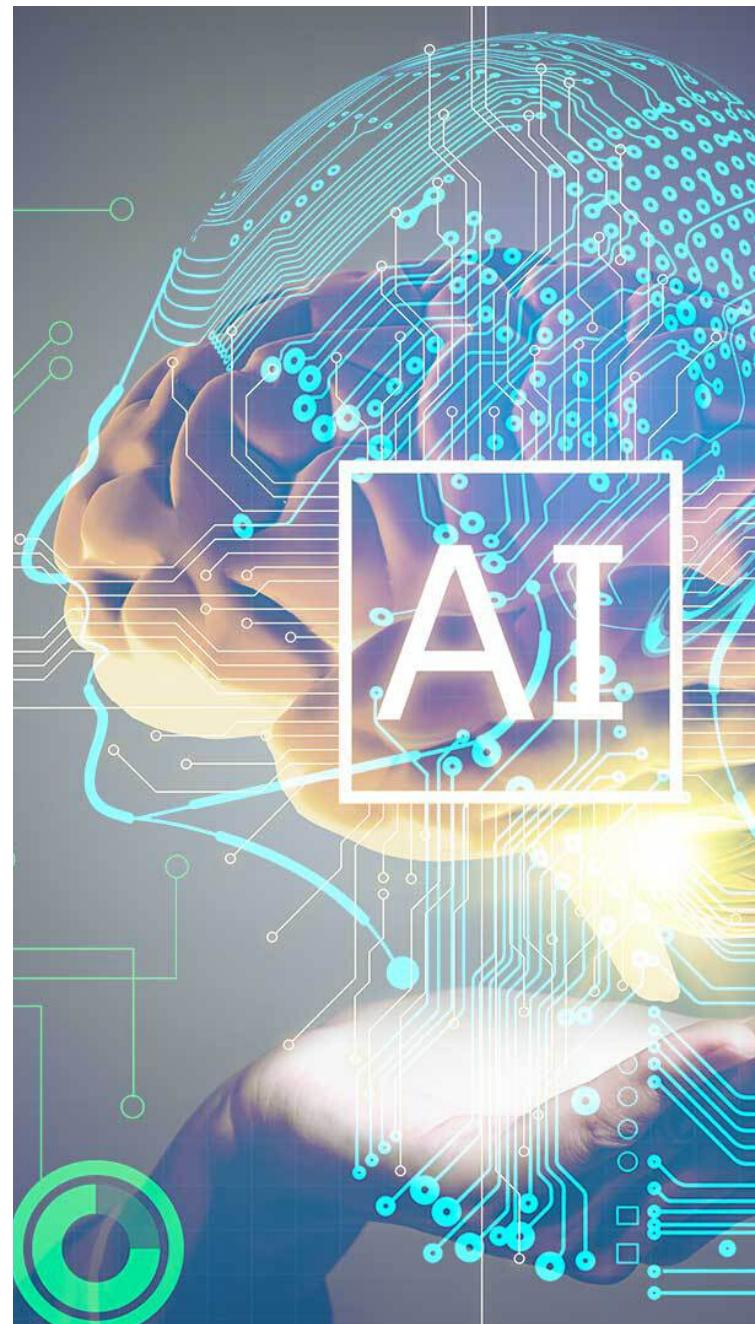
# 10.

## IMPLEMENTATIE

Zoals omschreven in hoofdstuk 04. *Randvoorwaarden & risico's*, bestaat dit project uit twee delen: het centrale en decentrale onderdeel. Het centrale gedeelte heeft als functie om een RNN te trainen volgens de reinforcement learning methode. Het decentrale gedeelte is verantwoordelijk voor het gebruiken van het model. Dit decentrale gedeelte wordt binnen Facilicom de **BlueBot** (Blue -> kleur Facilicom, Bot -> Robot) genoemd.

Omdat er gebruik gemaakt wordt van TensorFlow, is de implementatie voor de BlueBot relatief simpel. Door de export en import functie in dit framework is het niet nodig om een netwerk te trainen, nog te definiëren. Hierdoor wordt de implementatie heel simpel: Het was alleen nodig om een aparte class te maken, die het model importeert, en vervolgens in deze class een predict functie aan te maken, die het model vraagt om een voorspelling te doen.

Hierna kwam het centrale gedeelte van het project, **Skynet** (terminator). Skynet is een NodeJS applicatie wat gebruikt maakt van de GPU (Graphical Processing Unit). Hiermee zorgen er ervoor dat het trainen van een model zo snel mogelijk verloopt.



Skynet heeft vele iteraties meegemaakt. In de rest van dit hoofdstuk omschrijf ik kort wat iedere iteratie nieuw bracht en hoe dit conceptueel uitgewerkt is.

### **Versie 1.0.0**

Voor versie 1.0.0 van Skynet heb ik gebruik gemaakt van de volgende JS library van TensorFlow: @tensorflow/tfjs-node-gpu versie 1.7.4. Op het moment van coderen was dit de laatste beschikbare versie.

In deze versie heb ik een NN gemaakt dat in staat zou moeten zijn om de huidige temperatuur te nemen en op basis daarvan beslissingen te nemen. Hiervoor had de NN 1 input neuron (temperatuur, genormaliseerd tussen -20 en 40 graden celsius) en 3 output neuronen (niks doen, verwarmen en koelen). Eerst heb ik supervised machine learning (SML) gebruikt om aan te tonen dat deze vorm van het NN zal werken. Hiervoor dat ik zelf een dataset gemaakt en gebruikte. Uiteindelijk werkte het prototype en ben ik doorgegaan om het SML te vervangen met een RL implementatie, wat uiteindelijk ook succesvol afgerond was.

### **Versie 1.1.0**

Versie 1.0.0 heeft aangetoond dat de aannames van het project correct waren. Hierdoor kon er in versie 1.1.0 nagedacht worden over de vorm en structuur van het project. Hiernaast zijn er ook unit testen toegepast op het project en is het CPS geïmplementeerd.

### **Versie 1.2.0**

In deze versie van het project is er een extra parameter voor het NN geïntroduceerd: Tijd. Ook deze waarde is genormaliseerd, naar minuten. Met deze parameter is het NN zich bewust, op de minuut exacte,

### **Versie 1.3.0**

In deze versie wordt er afgestapt van de datageneratie en wordt er gebruik gemaakt van echte data. Voor RL is het niet erg om zelf gegenereerde data te gebruiken aangezien je de CPS traint met de data en niet direct het NN. Ondanks dat, was dit zeker geen onbelangrijke versie.

### **Versie 2.0.0**

Versie 2.0.0 heeft een heel groot probleem aangepakt: Sequentiële data. In andere woorden, hoe gaan we om met data dat patronen vertoont over tijd. Dit is gedaan door een LSTM layer in het NN te plaatsen. Hierdoor is het type NN veranderd naar een RNN. Het resultaat was zeer succesvol! Het reward systeem moet in deze versie nog wel geavanceerder worden om nog meer besparingen te kunnen winnen met het gebruik van dit systeem.

### **Versie 2.1.0**

Deze versie zorgde voor nog meer data voor het RNN om beslissingen mee te maken. Zo is er buitentemperatuur, zonnen inslag, relatieve luchtvochtigheid, windsnelheid, windrichting, neerslag en dag van de week toegevoegd. Bezetting van de ruimte zou nog een waardevolle toevoeging zijn, maar dat komt later

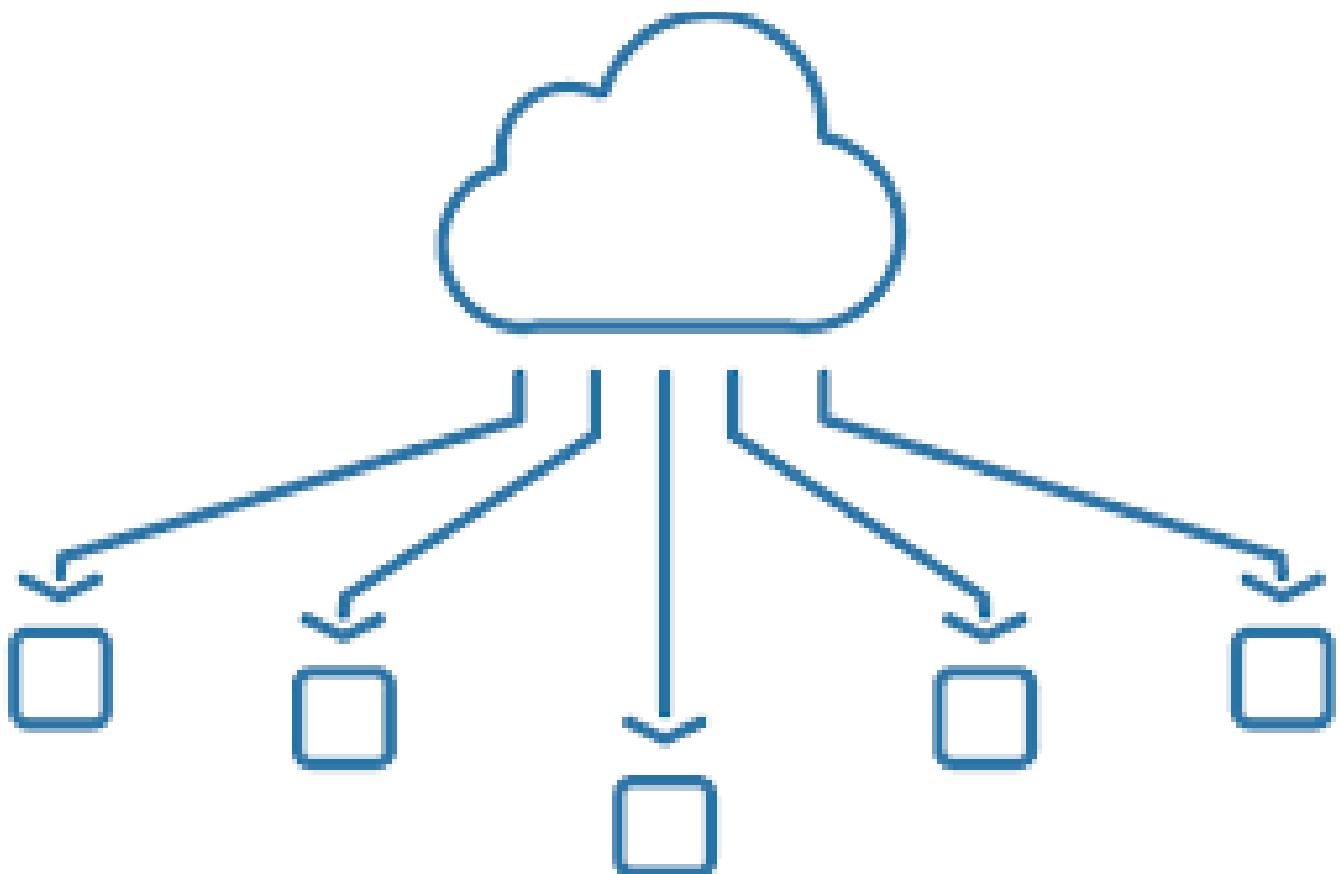
### Versie 3.0.0

In deze versie is het CPS omver gegooid. Zo is het onderzoek van Folmer van Rijn geïmplementeerd en is de wiskundige benadering verwijderd. Ook is de outlook agenda toegevoegd om de bezetting van een ruimte in beeld te krijgen. Dit alles zorgde ervoor dat het resultaat veel stabieler werd.

### Versie 3.1.0

Na een meeting gehad te hebben met Siem van den Reijen, is versie 3.1.0 uitgebracht. In deze versie heeft Siem zijn feedback verwerkt. Zo is de loss functie veranderd van Categorical Crossentropy naar de Bellman equation. Daarnaast is ook het reward systeem aangepast om meer of minder te belonen op basis van de verbruikte energie.





## 11. DEPLOYMENT

Machine learning is een best pittig process. Zowel voor de data scientist als voor de server zelf. Hierom heeft Facilicom geïnvesteerd in een server specifiek voor de (komende) machine learning opdracht(en). Hieronder staan de specificaties van deze server:

- CPU: Intel i5-4440 @ 3.3GHz
- Geheugen: 24Gb DDR3
- Opslag: 250GB SSD
- GPU: Nvidia RTX 2070 SUPER
- OS: Ubuntu 18.04

Het project is zelf verantwoordelijk voor het verzamelen van data en bepalen voor wanneer het het training process moet starten. Hierom was het een wens om dit project onder een process manager te laten vallen.

Binnen Linux is hier een hoop in mogelijk, maar voor NodeJS heb je 1 koning daarin: PM2. Met PM2 is het mogelijk om je NodeJS applicatie te monitoren, maar ook om nieuwe versie van je applicatie te deployen. Gecombineerd met Github actions is hier een CI/CD pipeline gebouwd, met unit tests.

Tijdens de deployment is er ook een benchmark gedaan met 3 verschillende servers. De resultaten zijn bijgevoegd in *Bijlagen E*.

# 12.

# RESULTATEN

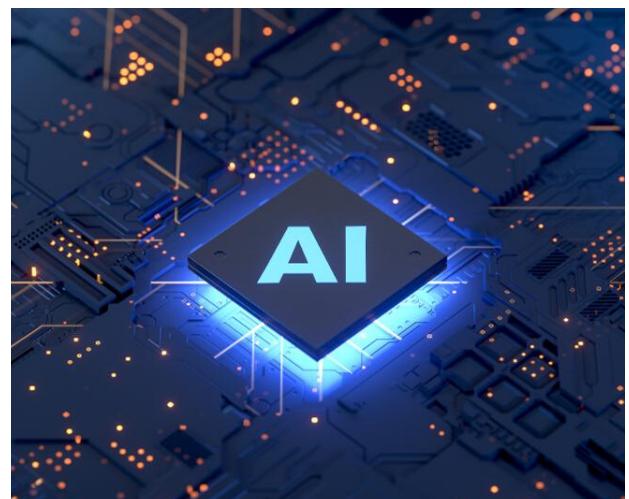
Helaas is het niet mogelijk om de performance van dit onderzoek te testen in een real world omgeving. De BACnet interface stond klaar, maar er was geen BACnet koppeling naar het netwerk toe. Wat wel mogelijk is, is het testen van de modellen tegen een test dataset. In deze stage wordt er gebruik gemaakt van 2 modellen: Het model omschreven volgens het onderzoek van Folmer van Rijn (van Rijn, 2018) en het Reinforcement learning model, wat verantwoordelijk is voor de keuzes.

## **Model voor CPS:**

Dit model wordt getraind op basis van Supervised machine learning. Omdat er pas vrij laat in de stage de correcte data beschikbaar is, was er maar een gelimiteerde hoeveelheid aan data beschikbaar. Ondanks dit, functioneert dit model, met andere data, ongeveer even goed als het model van Folmer van Rijn. In Appendix M van Folmers onderzoek is te zien dat hij een mean absolute error van 0.100 gemiddeld gezien. In *Bijlagen C* van dit verslag is te zien dat dit model een mean absolute error heeft van 0.0964. Dit betekend dat het model ongeveer 0.1 graden Celcius van de werkelijkheid verwijderd zit, zeker geen slechte score!

## **Reinforcement learning model:**

Omdat het niet mogelijk is om dit model te testen in het echt en dit model gebruik maakt van Reinforcement learning, is het onwijs lastig om echt resultaten te laten zien. De theorieën omschreven in dit verslag, tonen aan dat het mogelijk is om met dit type algoritme een oplossing te maken voor de HVAC wereld, maar er is onvoldoende data om hier echte conclusies uit te kunnen trekken.





## 13. **CONCLUSIE & AANBEVELING**

Zoals in de resultaten aan gegeven, in theorie zal deze aanpak werken en is het dus mogelijk om "reinforcement learning" te gebruiken om een optimaal klimaat te behouden terwijl we beter omgaan met de energie binnen een gebouw". Helaas is hier geen real world onderbouwing voor, doordat er te weinig data aanwezig is en de applicatie geen BACnet koppeling kon maken met de priva GBS installatie.

### **Aanbevelingen:**

Een van de makkelijkste aanbevelingen die ik kan doen aan de hand van deze stage is om een BACnet koppeling klaar te hebben. Hiermee kan je echt testen of het model functioneert naar behoren.

Verder raadt ik ook aan om meer en betrouwbaardere data beschikbaar te hebben. Meer data als in langer dan 1 maand & data over CO<sub>2</sub> meetingen bijvoorbeeld. Betrouwbaardere data als in geen data van outlook agenda, maar echte sensor meetingen.

# EVALUATIE

Met deze stage heb ik het maximale van mijzelf gevraagd op ieder vlak: kennis & vaardigheden (over data science maar ook over HVAC), planning (omdat onderwerp nogal pittig is, moet daar rekening gehouden worden in de planning) en kennisoverdracht (omdat Facilicom bijna geen kennis heeft over data science, is er een goot gat om te overbruggen).

Voor de stage was begonnen, was ik al in dienst van Facilicom. Ik heb daar meer dan 6 maanden al gewerkt. Deze periode heb ik besteed aan het voorbereiden van mijn stage: De technische ondersteuning ontmoeten, een platform voorbereiden en klanten kennis opbouwen. Dit heeft als sterke basis gediend voor mijn stage.

Ik was al best thuis binnen de HVAC-wereld en had al veel kennis daarover, ondanks dat heb ik een hoop geleerd. Mensen, zoals Dirk de Laat en Michel Maassen, hebben mij ondersteuning aangeboden waar ik deze nodig had en dit heeft heel erg geholpen tijdens mijn stage.

Hiernaast heeft Martijn Ploos van Amstel mij veel vrijheid gegeven om goed te kunnen werken aan mijn stage en deze vrijheid heb ik ook genomen. Ik zou niet meer weten hoeveel ik dagen niet te bereiken was, omdat ik druk bezig was met testen of researchpapers aan het lezen was.

Zelf vind ik het heel erg jammer dat ik geen proof of concept heb, enkel theorieën en een model dat zou moeten werken. Dit komt doordat er geen BACnet koppeling beschikbaar was met het GBS. Dit had meegenomen moeten worden in de voorbereidingen, samen met de controle op de database om de grote vertraging door de reset van de database te voorkomen.

Zoals ik al eerder heb gezegd, ik heb met deze stage het maximale van mezelf gevraagd. Dit heeft, samen met andere, een hele leuke stage gemaakt, waar ik ontzettend veel geleerd en gelachen heb.

De leerdoelen, die ik voor mezelf had vastgesteld, waren als volgt:

- **Documenteren**

Deze stage heb ik uitzonderlijk veel aandacht besteed aan documenteren. Dit in het belang van de kennisoverdracht die moeilijk zou worden. Ik denk dat met dit document dat goed gelukt is.

- **Analyse**

Ik wilde graag met deze stage ook goed in beeld brengen waarom deze stage zo belangrijk is. Dit heb ik gedaan met een PowerBI analyse.

- **Toepassen van RL algoritme**

Ook wilde ik graag snappen hoe RL werkt en waar het voor gebruikt kan worden.

# LITERATUURLIJST

- Voorbeeld reinforcement learning. (z.d.). Geraadpleegd op 10 maart 2020, van <https://qph.fs.quoracdn.net/main-qimg-b135e50fd568eac846f112ee8a0a1bbc>
- Logo ML.Net. (z.d.). Geraadpleegd op 10 maart 2020, van [https://res.cloudinary.com/practicaldev/image/fetch/s--UO68-pH3--/c\\_limit%2Cf\\_auto%2Cfl\\_progressive%2Cq\\_auto%2Cw\\_880/https://cdn-images-1.medium.com/max/1024/0\\*JXOjhOJM\\_SXfFuCvv.png](https://res.cloudinary.com/practicaldev/image/fetch/s--UO68-pH3--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https://cdn-images-1.medium.com/max/1024/0*JXOjhOJM_SXfFuCvv.png)
- Logo Brain.js. (z.d.). Geraadpleegd op 10 maart 2020, van <https://avatars2.githubusercontent.com/u/23732838?s=200&v=4>
- Logo TensorFlow. (z.d.). Geraadpleegd op 10 maart 2020, van <https://upload.wikimedia.org/wikipedia/commons/thumb/1/11/TensorFlowLogo.svg/1200px-TensorFlowLogo.svg.png>
- Voorbeeld TensorFlow. (z.d.). Geraadpleegd op 11 maart 2020, van <http://nlpx.net/wp/wp-content/uploads/2015/11/TensorFlow-graph1.jpg>
- Brownlee, J. (2019, 19 augustus). When to Use MLP, CNN, and RNN Neural Networks. Geraadpleegd op 15 april 2020, van <https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>
- Guillaume Chevalier. (z.d.). Long Short-Term Memory (LSTM) cell [Foto]. Geraadpleegd van [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory#/media/File:The\\_LSTM\\_cell.png](https://en.wikipedia.org/wiki/Long_short-term_memory#/media/File:The_LSTM_cell.png)
- fde洛che. (z.d.). Fully recurrent neural network [Foto]. Geraadpleegd van [https://en.wikipedia.org/wiki/Recurrent\\_neural\\_network#/media/File:Recurrent\\_neural\\_network\\_unfold.svg](https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Recurrent_neural_network_unfold.svg)
- Colah. (z.d.). Understanding LSTMs. Geraadpleegd op 15 april 2020, van <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Wikipedia. (2018, 7 februari). BACnet. Geraadpleegd op 16 april 2020, van <https://nl.wikipedia.org/wiki/BACnet>
- Wikipedia. (2020, 17 april). Long short-term memory. Geraadpleegd op 21 april 2020, van [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory)
- Towards Data Science. (2017, 6 september). Activation Functions in Neural Networks. Geraadpleegd op 21 april 2020, van <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- Wikipedia. (2018a, 12 januari). Lineaire vergelijking. Geraadpleegd op 21 april 2020, van [https://nl.wikipedia.org/wiki/Lineaire\\_vergelijking](https://nl.wikipedia.org/wiki/Lineaire_vergelijking)

- Towards Data Science. (2018, 2 augustus). Applications of Reinforcement Learning in Real World. Geraadpleegd op 21 april 2020, van <https://towardsdatascience.com/applications-of-reinforcement-learning-in-real-world-1a94955bcd12>
- IEEE. (2007, 1 november). News Briefs - IEEE Journals & Magazine. Geraadpleegd op 21 april 2020, van <https://ieeexplore.ieee.org/document/4385266>
- Kadaster. (z.d.). Kadaster Geyssendorfferweg 5 Rotterdam. Geraadpleegd op 21 april 2020, van <https://bagviewer.kadaster.nl/lvbag/bag-viewer/index.html#/searchQuery=geyssendorfferweg%205%20Rotterdam&resultOffset=0&detailsObjectId=0599010000391204&objectId=0599100000628292&geometry.x=90018.82&geometry.y=432604.72&zoomlevel=6>
- van Rijn, F. (2018, 12 december). Thesis Folmer van Rijn. Geraadpleegd op 29 april 2020, van [https://github.com/bastiaanv/stage-verslag/blob/master/Thesis\\_Folmer\\_v1.0.docx?raw=true](https://github.com/bastiaanv/stage-verslag/blob/master/Thesis_Folmer_v1.0.docx?raw=true)
- van Turnhout, K. (2015, 21 december). CMD Methods. Geraadpleegd op 6 mei 2020, van <https://cmdmethods.nl/>
- RVO. (2018, 1 januari). Monitor energiebesparing gebouwde omgeving. Geraadpleegd op 6 mei 2020, van [https://www.rvo.nl/sites/default/files/2020/01/monitor-energiebesparing-gebouwde-omgeving-2018\\_0.pdf](https://www.rvo.nl/sites/default/files/2020/01/monitor-energiebesparing-gebouwde-omgeving-2018_0.pdf)
- Wikipedia. (2020, 11 april). Matrix (wiskunde). Geraadpleegd op 7 mei 2020, van [https://nl.wikipedia.org/wiki/Matrix\\_\(wiskunde\)](https://nl.wikipedia.org/wiki/Matrix_(wiskunde))
- Verhaar, B. C. (2020, 13 mei). PowerBI analyse stage. Geraadpleegd op 13 mei 2020, van <https://github.com/bastiaanv/stage-verslag/blob/master/Analyse%20stage.pbix?raw=true>

# BIJLAGEN A - OVERZICHT DATA

## **Priva history:**

- Ruimte temperatuur (in °C)
- Klepstand verwarming (van 0 tm 10000 mV)
- Klepstand Koeling (van 0 tm 10000 mV)

## **KNMI:**

- Buiten temperatuur (in °C, meternummer: 344-00000000000069)
- Globale straling (in J/cm<sup>2</sup>, meternummer: 344-00000000000070)
- Relatieve luchtvochtigheid op 1.5 meter hoog (in %, meternummer: 344-00000000000072)
- Uurgemiddelde windsnelheid (in m/s, meternummer: 344-00000000000073)
- Windrichting (in graden, meternummer: 344-00000000000074)
- Uursom neerslag (in mm, meternummer: 344-00000000000076)

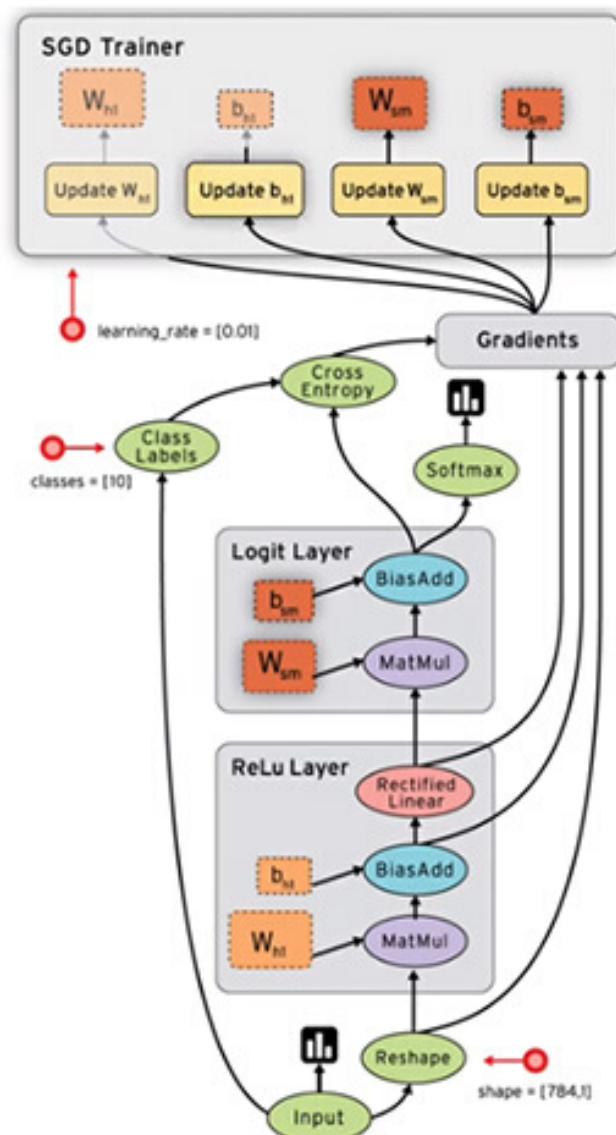
## **Outlook agenda:**

- bezetting (als boolean)

# BIJLAGEN B - HOE WERKT TENSORFLOW?

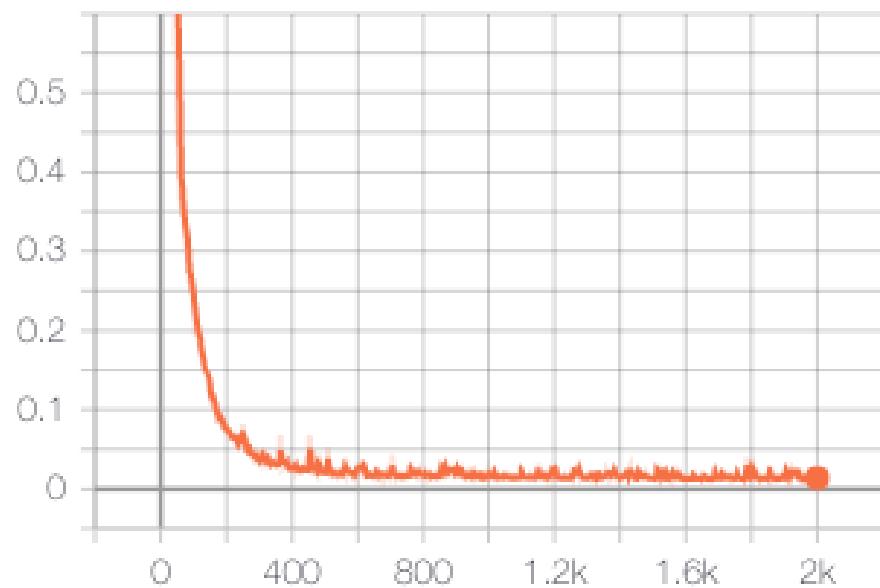
TensorFlow werkt met het principe van tensoren. Het is een algebraïsch object dat de relatie tussen verschillende algebraïsche objecten omschrijft. Deze objecten kunnen scalair, matrixen of andere tensoren zijn. Wat je met TensorFlow kunt doen, is een reeks van deze tensoren omschrijven, of te wel een flow van tensoren omschrijven.

Een mooi voorbeeld hiervan, is hiernaast weergegeven in de afbeelding hiernaast. De flow begint, onderin, bij de input. Hier komen de initiële waarden in terecht. Deze wordt in een matrix vorm gebracht en gaat vervolgens de verschillende layers in. Deze layers samen representeren een neural network, maar dit had ook een regressie kunnen zijn bijvoorbeeld. Uit dit neural network kunnen nog 2 zaken gebeuren: Het antwoord vormen of de loss functie. De loss functie word gebruikt door de gradients en de SGD Trainer om de  $W$  en  $b$  te verbeteren om zo het neural network te trainen. Het antwoord wordt gevormd door de Softmax functie en kan gebruikt worden in andere delen van een applicatie.

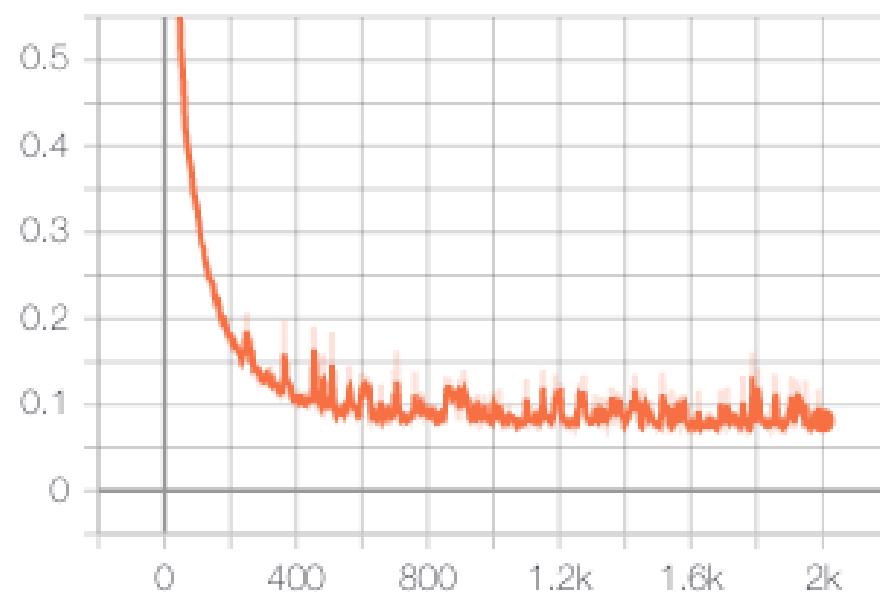


## BIJLAGEN C - RESULTAAT MODEL VOOR CPS

epoch\_mse



epoch\_meanAbsoluteError



# BIJLAGEN D - DE THEORIE ACHTER REINFORCEMENT LEARNING

Er zijn een hoop verschillende soorten manieren om een reinforcement learning algoritme toe te passen. Een van de meest gebruikte manier is Q-Learning (QL) en Deep-Q-Learning (DQL).

QL maakt gebruik van een Q-table (zie afbeelding hiernaast). Deze table heeft over zijn x-as alle acties staan en over de y-as de verschillende states. Iedere cel in die tabel bevat een waarde genaamd: heuristic. Deze waarde geeft aan hoe groot de kans is dat dit de goede keuze is. Deze waarde heeft een range tussen 0 en 1, waarbij 1 heel zeker is en 0 heel onzeker.

Het grote verschil tussen DQL en QL, is dat er bij DQL er geen gebruik gemaakt wordt van een Q-tabel, maar van een NN. Iedere output van het NN represeneert een actie op de Q-table, iedere input represeneert de state en de heuristic is de waarde in de output neuron zelf.

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	327	0	0	0	0	0	0
	.	.	.	.	.	.	.
	499	0	0	0	0	0	0

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	.	.	.	.	.	.	.
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

Om de heuristic te kunnen berekenen heeft Richard Bellman in 1959 de Bellman equation gepubliceerd. De formule is in de afbeelding hieronder te zien:

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \underbrace{\left( \underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\substack{\text{estimate of optimal future value}}} \right)}_{\text{temporal difference}} - \underbrace{Q(s_t, a_t)}_{\text{old value}}$$

new value (temporal difference target)

# BIJLAGEN E - BENCHMARK LEARNING PROCESS

Het benchmarken is op 3 verschillende computers gedaan:

## **Deployment server van Facilicom:**

- CPU: Intel i5-4440 @ 3.3GHz
- Geheugen: 24Gb DDR3
- Opslag: 250GB SSD
- GPU: Nvidia RTX 2070 SUPER
- OS: Ubuntu 18.04

## **Macbook Pro 2018:**

- CPU: Intel i9-8900HK @ 2.6GHz
- Geheugen: 16 Gb DDR4
- Opslag: 0.5TB SSD
- GPU: Niet gebruikt tijdens de benchmark
- OS: Mac OS Catalina 10.15.4

## **Server Hexite software:**

- CPU: Intel i7-8700K @ 4.7 GHz
- Geheugen: 32GB DDR4
- Opslag: 0.5TB SSD
- GPU: Nvidia Quadro P4000
- OS: Ubuntu 20.04

## **Resultaten** gem over 5 iteraties:

- Deployment server Facilicom:

MLP = 360ms

RL = 60ms

- Macbook Pro 2018:

MLP = 240ms

RL = 23ms

- Server Hexite software:

MLP = 235ms

RL = 40ms