



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

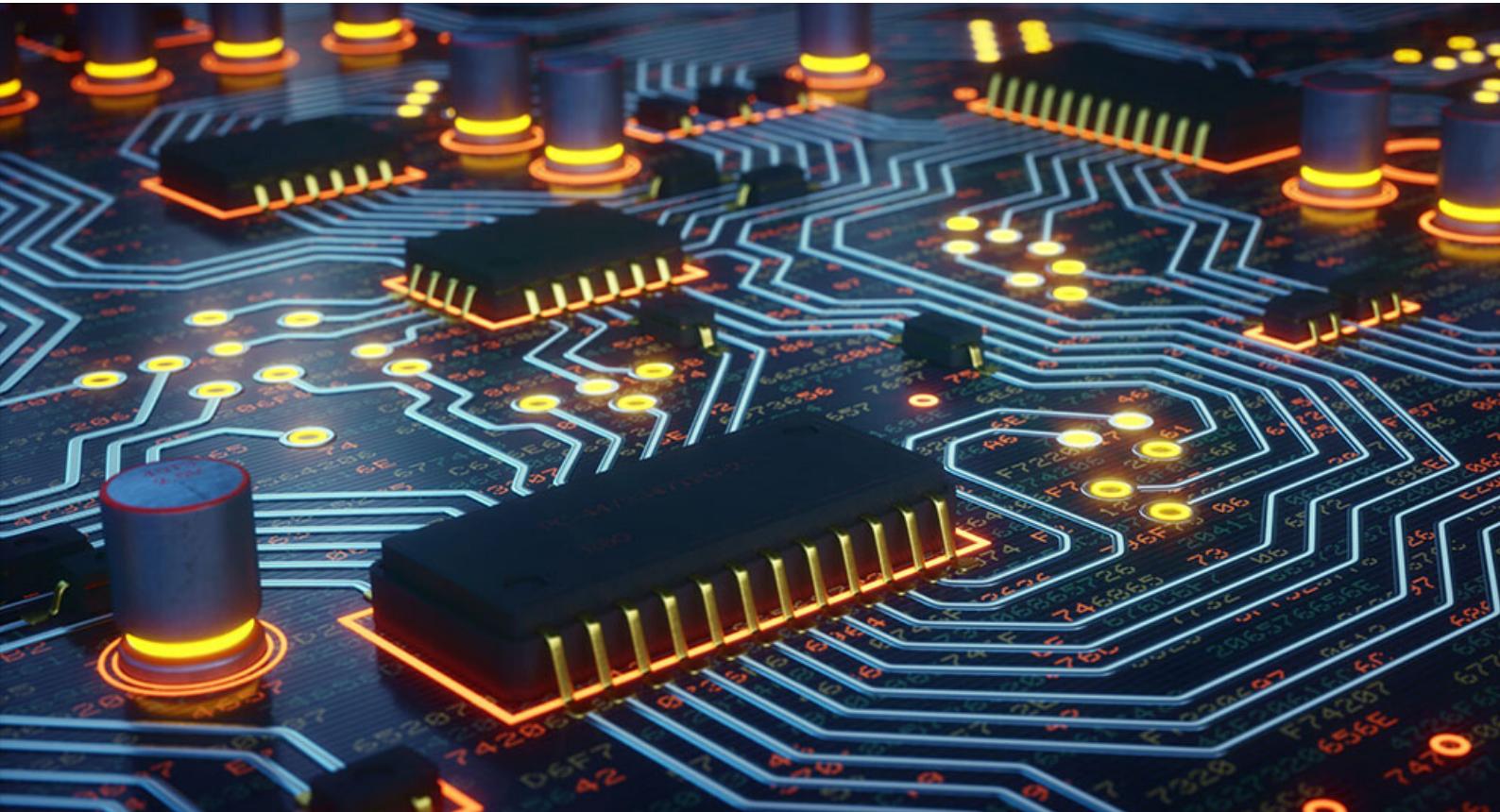
Gegevens verslag:

Stageverslag, Reinforcement learning
binnen HVAC-wereld
10.02.20

**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 op hun locatie 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: Martijn Ploos van Amstel, Michel Maassen, Dirk de Laat en John Litsenburg

Bastiaan Verhaar

INHOUDSOPGAVE

0 6

SAMENVATTING

0 7

01. INLEIDING

0 8

02. HET BEDRIJF

0 9

03. DE OPDRACHT

1 0

04. RANDVOORWAARDEN
& RISICO'S

1 2

05. GBS?

1 4

06. REINFORCEMENT
LEARNING?

2 2

07. CPS?

2 4

08. SITUATIE

0 ?

09. IMPLEMENTATIE RL

0 ?

10. RESULTATEN

0 ?

11. CONCLUSIE



FIGURENLIJST

Figuur 1. Voorbeeld schematische tekening GBS. P12

Figuur 2. Interface Priva BACnet. P13

Figuur 3. Voorbeeld Reinforcement learning (“Voorbeeld reinforcement learning”, z.d.). P14

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

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

Figuur 6. Logo ML.NET (“Logo ML.Net”, z.d.). P16

Figuur 7. Logo Brain.js (“Logo Brain.js”, z.d.). P16

Figuur 8. Logo TensorFlow (“Logo TensorFlow”, z.d.). P17

Figuur 9. Voorbeeld TensorFlow (“Voorbeeld TensorFlow”, z.d.). P18

Figuur 10. Plattegrond begane grond Geyssendorfferweg 5 te Rotterdam. P22

Figuur 11. Schematische tekening installatie meldkamer Breijer. P22

Figuur 12. Verbruik kantoren maand Februari 2020 in MW.h. Verwarmen (paars) & koelen (blauw). P23

Figuur 13. Schematische weergaven energieopwekking pand Rotterdam. P24

Figuur 14. Verbruik kantoren maand Maart 2020 in MW.h. Verwarmen (paars) & koelen (blauw). P24

Figuur 15. Temperatuur meldkamer (blauw) vs buiten (paars) op 09-03-2020. P25

Figuur 16. Plattegrond begane grond pand Rotterdam, rode cirkel Breijer meldkamer. P25

Figuur 17. Temperatuur kantoor BC-IT (blauw) vs sturing koeling per 10%-punten (paars) vs sturing verwarmen per 10%-punten (rood) op 23-03-2020. P25

SAMENVATTING

lol jk

lol jk



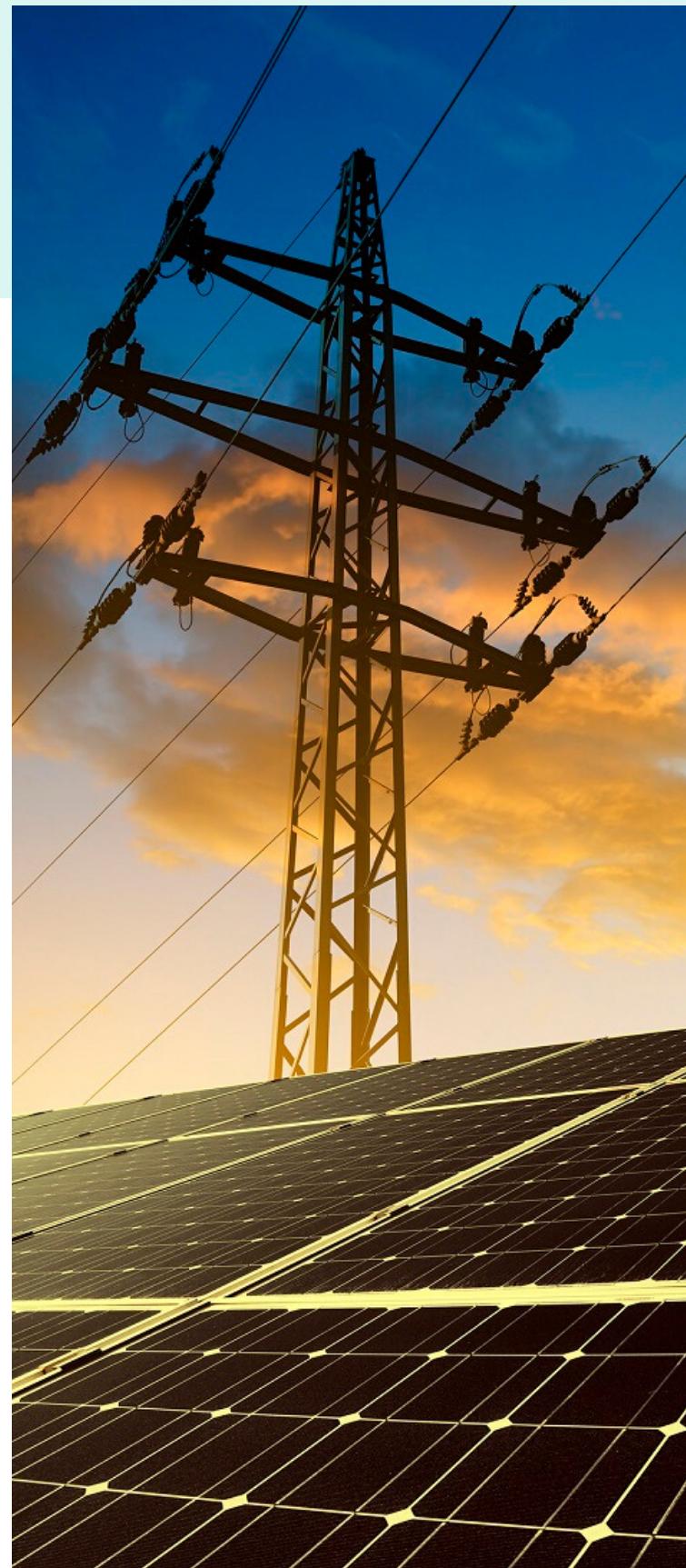
01. INLEIDING

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 bv. 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 CO₂ 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.

FEM heeft meerdere methodes om energie te besparen, al deze methodes vereisen een gedragsverandering. Bijvoorbeeld, verlichting uitzetten bij verlaten van kantoor, de verwarming uitlaten tijdens vakantie- en weekenddagen.





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. Bijvoorbeeld wanneer je spreekt over de informatieplicht of energielabels en nog veel meer

Dit bedrijf was eerst bekend als Ploos van Amstel Milieu Consulting in 1997, maar rond 2005 stond dit bedrijf bekend als Ploos energiestrategie. Pas in 2015 is dit bedrijf opgekocht door Facilicom Group en is de naam veranderd naar de huidige naam: Facilicom Energiemanagement (FEM). Opgericht door Joost Ploos van Amstel

is het doel van FEM om de energietransitie voor coöperaties te vermakelijken, met behulp van slimme oplossingen.

“Samen met u werken wij aan een CO₂ 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 heeft daarom deze opdracht uitgeschreven

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 energie consumptie binnen een gebouw?

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:

- Wat is een Gebouw Beheer Systeem? (H4)
- Aan welke randvoorwaarden moet dit systeem voldoen en wat zijn de risico's van dit 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 (begin) situatie? (H8)
- Hoe ziet de implementatie van de reinforcement learning eruit? (H9)
- Wat is het (eind) resultaat? (H10)

Start stage

Voordat de stage was begonnen had Facilicom nog niks op het gebied van machine learning.

Wel heeft Facilicom een aantal mensen rondlopen die mij op technisch gebied 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 vak inhoudelijke ondersteuning. Hierbij moet je denken aan technische inzichten over HVAC of GBS.

Doelen

Een van de doelen van deze opdracht is om te kijken 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 die ik een Gebouw Beheer Systeem gaat (zie meer in hoofdstuk 5) hoeft niet meer geschreven te worden. Dit zou allemaal door dit model opgelost kunnen worden, waardoor het GBS niet ingeregeld hoeft te worden. Dit zou FEM heel veel manuur schelen en dus geld.

04. RANDVOORWAARDEN & RISICO'S

RANDVOORWAARDEN

Voor ieder project is het altijd handig om van te voren duidelijk te hebben waar het aan moet voldoen. Met deze lijst van eisen, of randvoorwaarden, kan je voorkomen dat er communicatie fouten optreden en lever je een product op wat aan de verwachtingen van de andere kant voldoet.

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

- Het model moet gedeployed kunnen worden naar een NodeJS applicatie. Het communicatie middel moet MQTT zijn of via Rest
- Het onderdeel waar het model getraind wordt, moet centraal gebeuren. Er is één centrale plek waar Facilicom een sterke server klaar heeft staan. Hierdoor benutten ze deze server en kunnen ze alle andere onderdelen goedkoper houden. Dit dient binnen .Net core of NodeJS te gebeuren
- De temperatuur binnen conformzone, dient meer beloont te worden dan energieverbruik gestraft wordt voor het model





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 deze 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 kan je minimaliseren 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 installaties in het gebouw aangepast worden of niet goed ingeregeld is. 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. GBS?

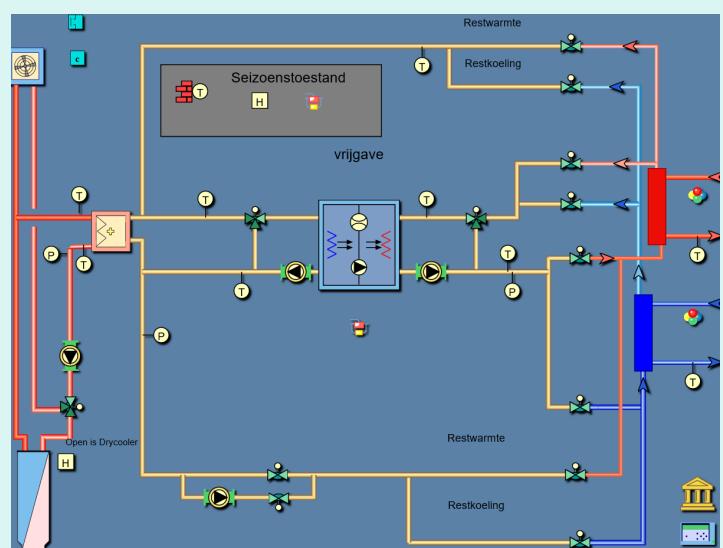
Een Gebouw Beheer Systeem (GBS) is een methode voor het toepassen van domotica (Latijns: Domos -> Huis, tica -> toegepaste wetenschap). Een definitie van domotica is: De integratie van technologie en diensten, ten behoeve van een betere kwaliteit van wonen en leven. Een GBS is gespecialiseerd in het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

Wanneer een gebouw opgeleverd wordt, zijn deze systeem, meestal, goed ingeregeld. Maar door de tijd heen, gaan er kleppen kapot of andere componenten. Deze situaties zorgen vaak voor koud klachten, waardoor een installateur ingeschakeld moet worden. De installateur gaat niet goed kijken 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 dit project 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 is en of de afgifte van warmte goed is, bijvoorbeeld. Hierdoor hoeft dit project geen rekening te houden met de senarios zoals eerder omschreven en mogen we er vanuit gaan dat alle installaties optimaal functioneren.

In *Figuur 1* is een schematische afbeelding te zien van de energieopwekking op de geyssendorfferweg 5 te Rotterdam (zie hoofdstuk 8 voor meer informatie).



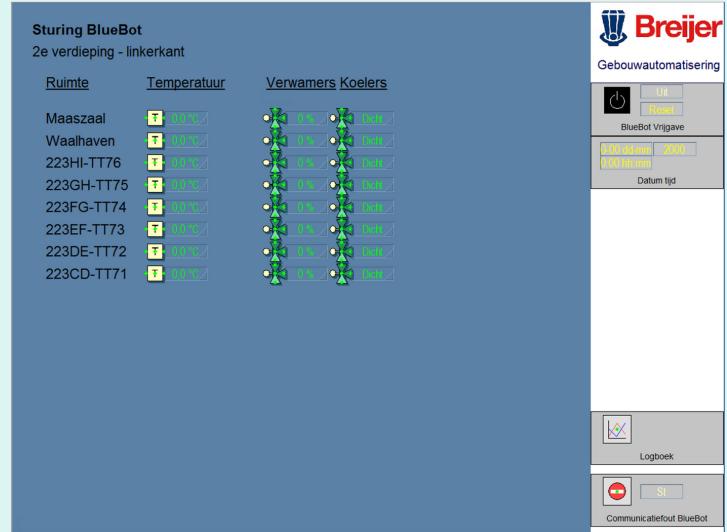
Figuur 1. Voorbeeld schematische tekening GBS

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 heeft vele varianten, waaronder serieel, met RS-232 of RS-485, maar ook een internet variant gebouwd op UDP. 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. Daarbij is het mogelijk om data te schrijven naar het GBS met dit protocol. Dit protocol gaat als basis dienen voor het ontvangen van data en het sturen van het GBS. (Wikipedia, 2018)

Om ervoor te zorgen dat de juiste data beschikbaar wordt gesteld vanuit BACnet, hebben Michel Maassen en ik een priva interface gemaakt, zie Figuur 2. 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 iets bijzonders; 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 temperatuur uit de opwekkers, bijvoorbeeld de CV ketel. Met de laatste strategie kan je een hele hoop energie besparen, omdat de CV-ketel bijvoorbeeld niet aanhoud speciaal voor de vloerverwarming. Door de nieuwe interface is het van belang dat de regelstrategie niet gevolgd wordt, omdat het Machine Learning model dit voortaan gaat bepalen



Figuur 2. Interface Priva BACnet

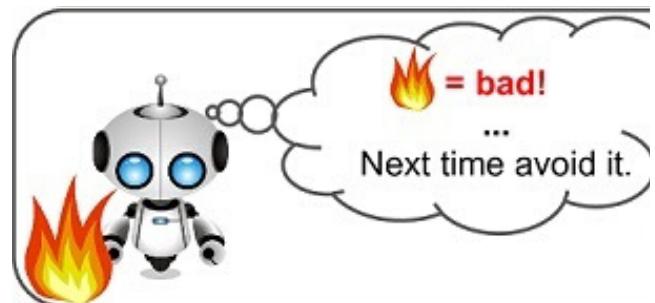
06. REINFORCEMENT LEARNING?

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 te 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 heeft meestal 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

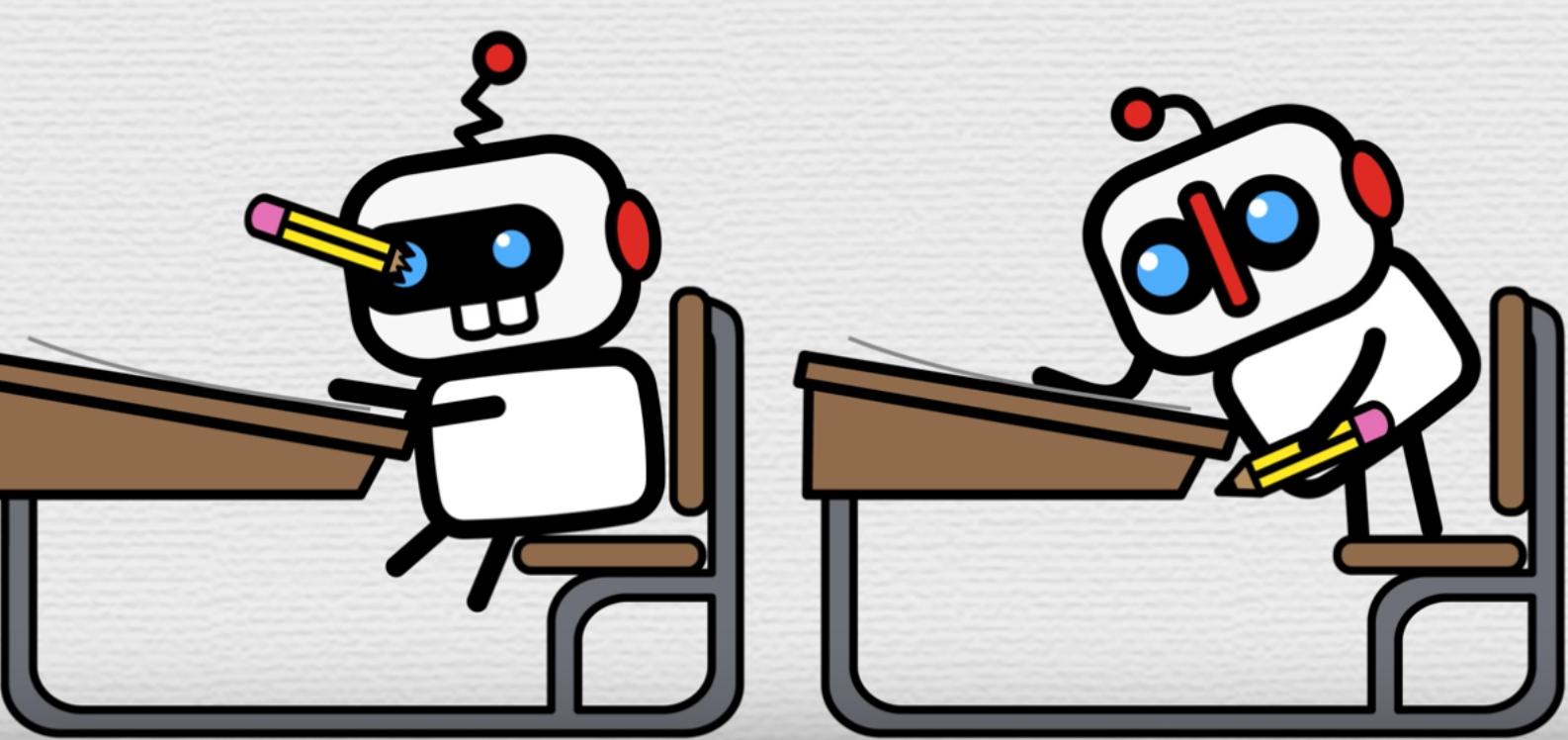
Het doel van de agent is om zoveel mogelijk punten te verzamelen. Om ervoor te zorgen dat de kans hiervoor zo groot mogelijk is, moet er nog een concept geïntroduceerd worden: Exploration. Soms is het goed om een random actie te doen, op deze manier kan de agent nieuwe manieren proberen en kijken of 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 reinforcement learning? Reinforcement learning 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 reinforcement learning al toegepast? Een van de bekendste voorbeelden van reinforcement learning is AlphaGo. Dit is een model dat de beste Go spelers: zoals Fan Hui (in 2015) of Lee Sedol (in 2016).

Maar volgens een onderzoek van towardsdatascience.com 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.





NEURAL NETWORKS

Een Neural Network (NN) is vaag 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 op de matrixen zijn de standaard lineaire vergelijkingen. De weights representeren de multiplicatie (a) en de biases de additie (b) van de formule.

$$y = a*x + b$$

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.

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.

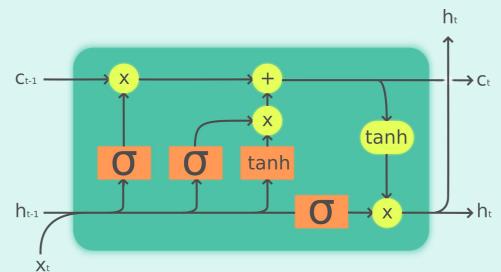
RECURRENT NEURAL NETWORKS

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. Hiervoor hebben wetenschappers een ander variant NN bedacht: Een Recurrent Neural Network (RNN).

Voor deze variant van NN is het verplicht om historische data mee te geven. Hierdoor is het in staat om patronen over de tijd te detecteren. In *Figuur 3*, is een voorbeeld gegeven van een fully recurrent NN. Hier is te zien dat iedere neuron data van de vorige staat, in andere woorden: de historische data, meeneemt in de berekening

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

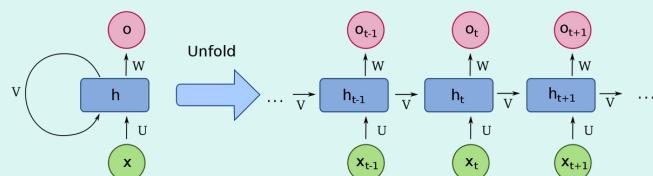
Verder zijn er nog veel meer types RNN: van fully recurrent tot Gated recurrent unit met ieder zijn eigen voor- en nadelen. Maar omdat, volgens het bovenstaande artikel, met een LSTM het meeste succes geboekt wordt, is er voor dit type RNN gekozen voor dit onderzoek.



Legend:

Layer Pointwise op Copy
 ↓ ↗

Figuur 5. Voorbeeld LSTM



Figuur 4. Voorbeeld fully recurrent neural network

De werking van een LSTM-cel is weergegeven in *Figuur 4*. Zo'n soort cel werkt als volgt: Een cel heeft drie inputs, 2x historische en 1x met de actuele data. Verder bevat een cel drie "gates": een Forget-, een input- en een output-gate. Deze drie gaan zijn weergegeven in *Figuur 4* als de lijn die van de bovenste horizontale lijn naar de onderste toe gaan, of andersom.

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.).



TOOLS VOOR RL

Tools voor reinforcement learning. Er zijn ongelofelijk veel tools voor machine learning voor alle 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 6. Logo ML.NET

Brain.js

Deze tool is bibliotheek voor Neural Networks. Het bevat een hoop wiskundige trucjes waardoor deze tool als volledige machine learning tool gezien kan worden. Zo kan je reinforcement learning models maken met enkele regels aan code!

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



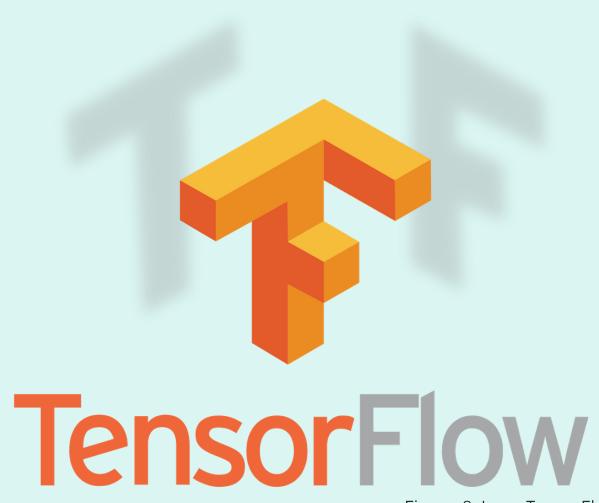
Figuur 7. Logo Brain.js

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 Python, wat gecompileerd kan worden naar C++. 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 development. 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.

"TensorFlow is an end-to-end open source platform for machine learning" -
Tensorflow.org



Figuur 8. Logo TensorFlow

Keuze

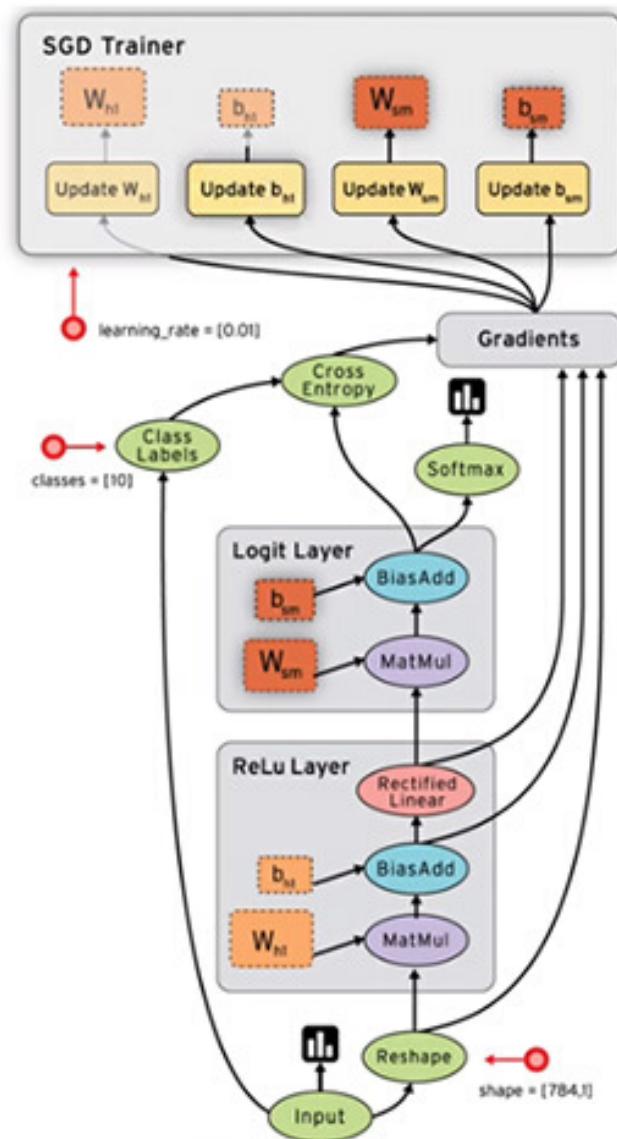
Het is mogelijk om een model op het ene framework te trainen en vervolgens op een ander model te deployen. In theorie heb je ander de vorm van je neural network nodig, en alle weights & biases. Dit kost alleen heel veel extra tijd en energie om dit goed te doen, terwijl we een oplossing hebben waar dit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

Dus omwille van tijd en efficiëntie is er dus gekozen voor het TensorFlow framework

Hoe werkt TensorFlow?

TensorFlow werkt met het principe van tensoren. Het is een algebraïsche 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 *Figuur 6*. 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.



Figuur 9. Voorbeeld TensorFlow



07. CPS?

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 onderzoeksprisen uitgereikt aan inzendingen van de NSF workshop.

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 leidt tot de volgende wiskundige weergaven van de temperatuur in een ruimte:

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

T̄: Nieuwe temperatuur als resultaat van tijdsverandering

a: Standaard warme verlies aan de buitenomgeving

T: Huidige temperatuur

Tr: Minimale temperatuur in de ruimte

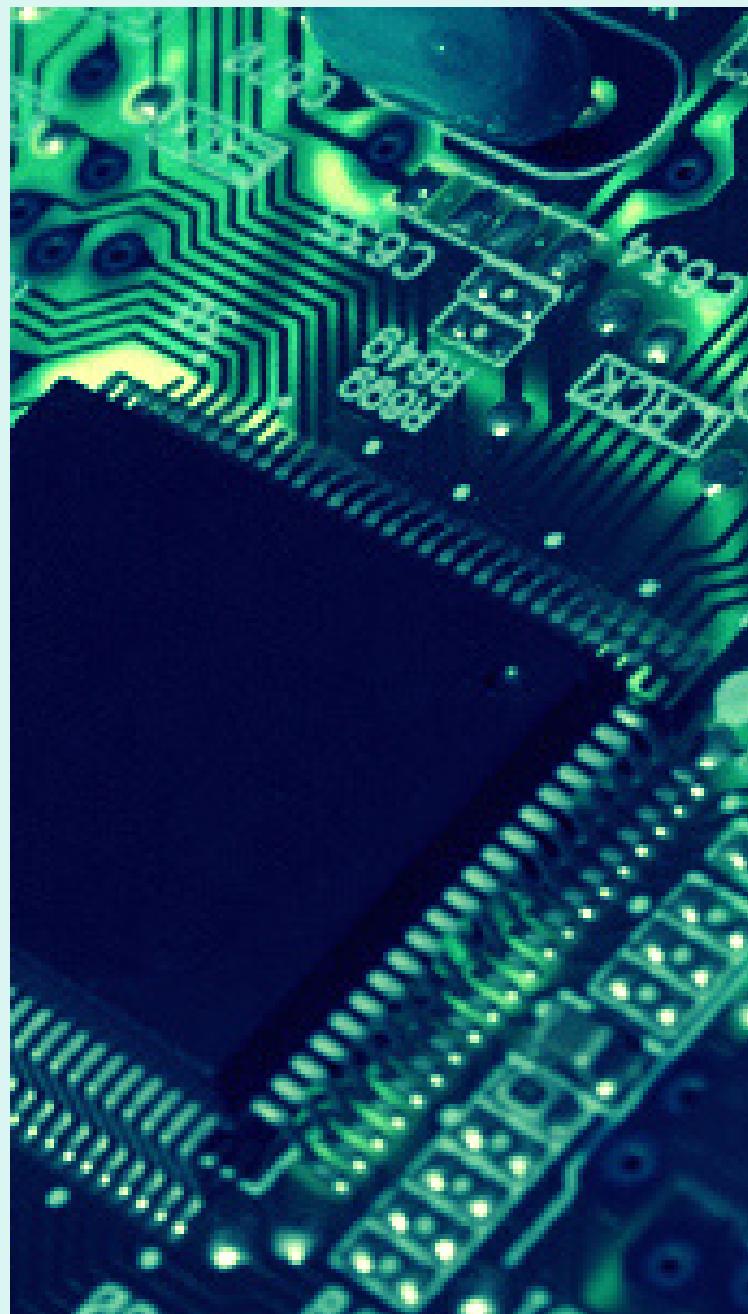
Th: Verandering in temperatuur door het inschakelen van de verwarming (bijvoorbeeld)

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

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

uc: 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 er daarvoor te veel elementen aanwezig zijn. Je moet hierbij denken aan: ramen die open kunnen, deuren die open blijven staan, de ruimte opwarming door de zon. Wel is het zo dat je deze element kan meenemen in de formule. Je dient alleen een benadering te vinden voor deze elementen.





08. **SITUATIE**

In de afbeelding hierboven is het hoofdkantoor van Breijer weergegeven. Dit kantoor is gevestigd op Geyssendorfferweg 5 te Rotterdam. Dit pand is gebouwd in 1959 en bevat 4826 m² aan oppervlakte op de begane grond. Het is in beheer van de Breijer groep zelf, waar in 2006 een Priva HX GBS is geplaatst.

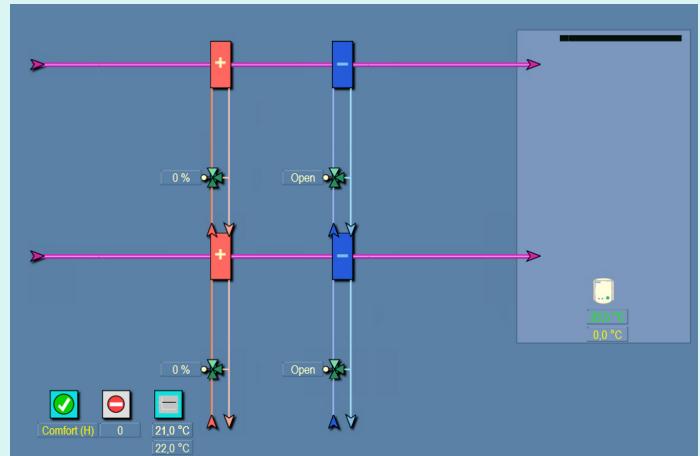
"Onder de vuurtoren schijnt geen licht". Dit is een gezegde waar de Breijer groep heel erg goed in is. Ze zorgen heel goed voor panden van klanten, maar niet haar eigen panden.

Dit is zeer goed te zien in de kantoren van dit pand. De 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 SQL database te krijgen. Deze gegevens zijn op basis van 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 SQL server. Aan de hand van deze gegevens is er een analyse gemaakt met Power BI.



Figuur 10. Plattegrond begane grond
Geyssendorfferweg 5 te Rotterdam



Figuur 11. Schematische tekening installatie meldkamer Breijer

21.4°C

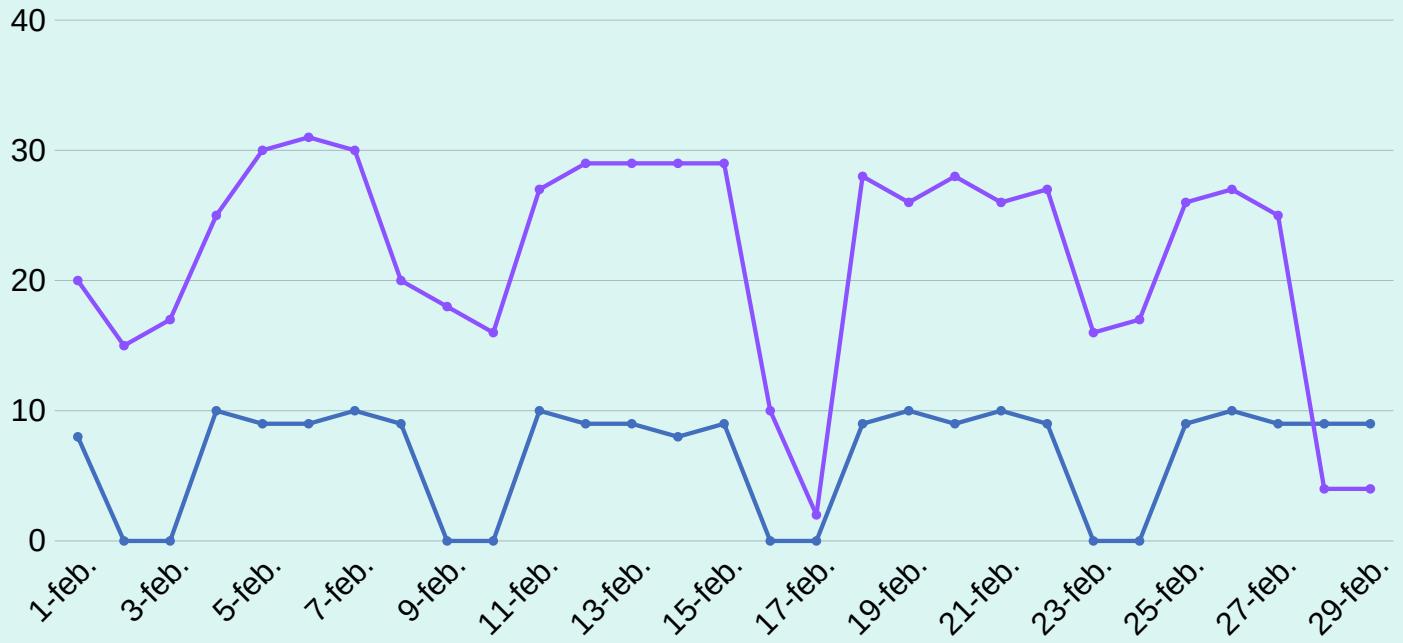
De gemiddelde temperatuur van alle ruimtes

373 KW.H

De gemiddelde verbruik per dag per ruimte

€4.088

De gemiddelde prijs voor temperatuur bewaking per jaar per ruimte



Figuur 12. Verbruik kantoren maand Februari 2020 in MWh. Verwarmen (paars) & koelen (blauw)

In Figuur 9 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.

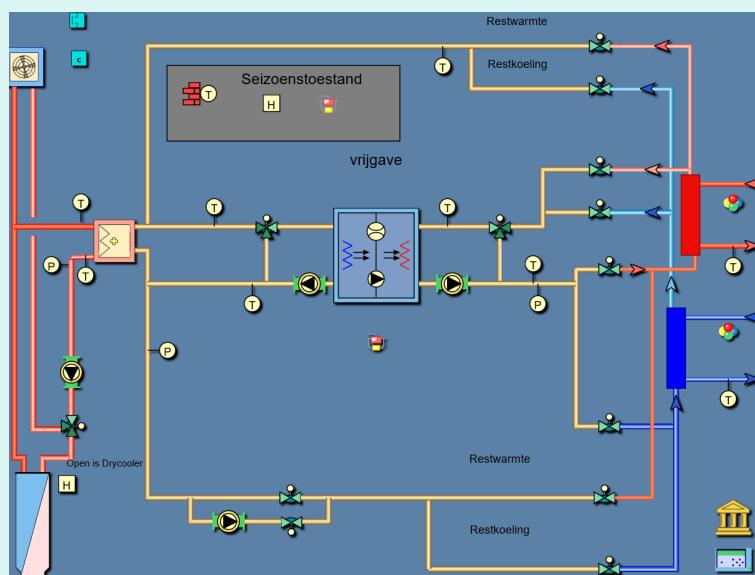
Op een "top"-dag, 6 februari bijvoorbeeld, werd in deze periode 40 MWh aan energie verbruikt, puur en enkel aan klimaat / temperatuur bewaking in de kantoren. In het pand zijn 69 kantoren. Dit zou betekenen dat op een "top"-dag gemiddeld gezien ieder kantoor 579.7 KWh zou verbruiken.

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.

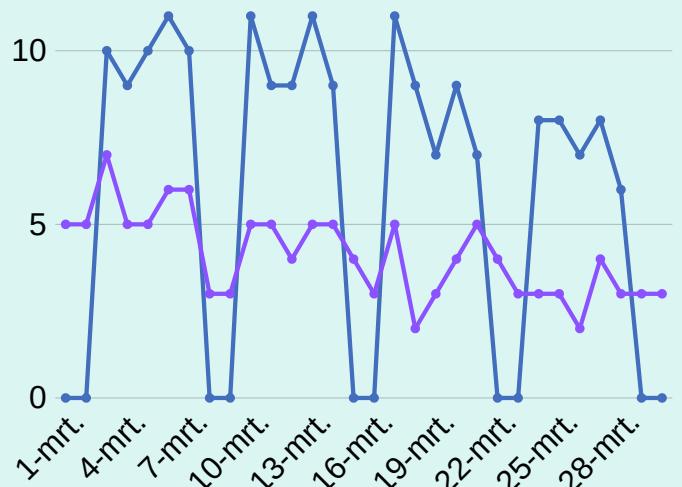
Opwekking is het onderdeel wat verantwoordelijk is voor de opwekking van warmte of koud. Je kunt hier denken aan een warmtepomp, een CV-ketel, koelelement, etc. **Transport** is het onderdeel wat verantwoordelijk is voor de verplaatsing van deze opgewekte energie. Dit zijn bijvoorbeeld je leidingen in een gebouw. Als laatste heb je dus de **afgifte**. Dit zorgt ervoor dat je de opgewekte energie in een ruimte krijgt. Je kunt hierbij denken aan je radiatoren, luchtschachten.

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



Figuur 13. Schematische weergaven energieopwekking pand Rotterdam

15



Figuur 14. Verbruik kantoren maand Maart 2020 in MW.h. Verwarmen (paars) & 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 10* 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.

Deze aanpassing 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 te meetbaar. Waarbij het verbruik per kantoor per dag in de maand februari op 373 KWh lag, lag het in de maand maart op 153 KWh. Hierdoor dalen de jaarlijkse lasten per kantoor voor temperatuur bewaking naar 4.59 euro. Een daling van maar liefst 57.3%!

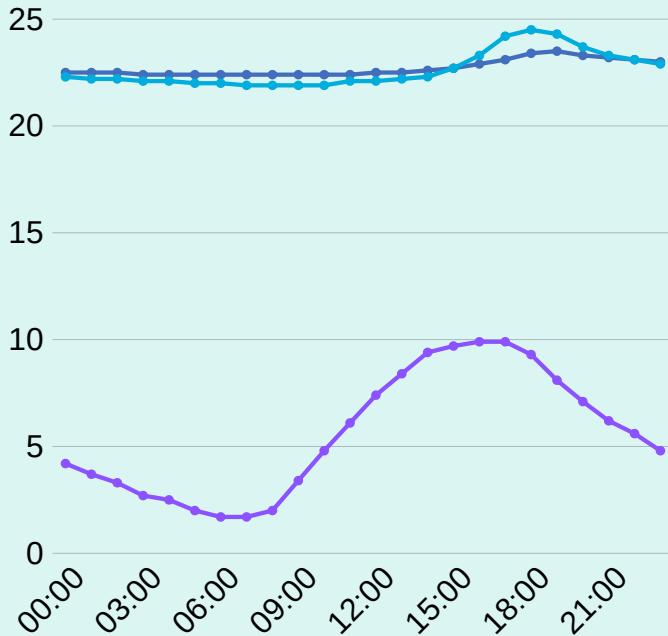
Het verdere verloop van de aanpassing is te zien in *Figuur 11*. Uit *Figuur 11*, hebben Michel Maassen en ik geconcludeerd dat het systeem beter functioneert en dat de rest van de winsten te behalen zijn in een betere / slimmere warmte / koud vraag vanuit de kantoren.

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



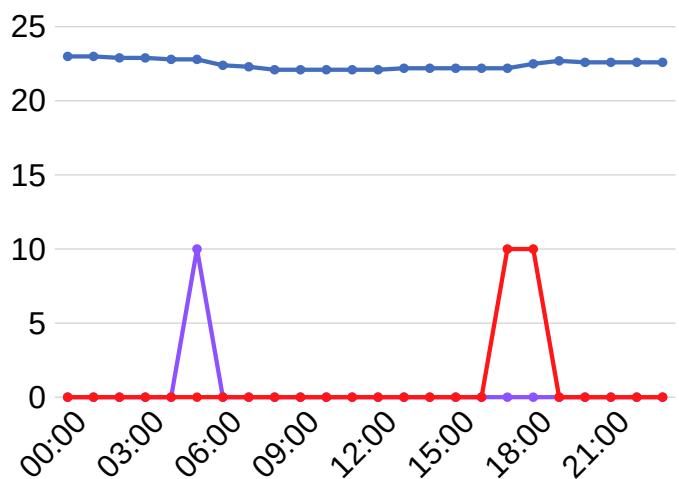
Figuur 16. Plattegrond tweede verdieping pand Rotterdam, rode cirkel kantoor BC-IT, blauwe cirkel meetingruimte Maaszaal

De volgende stap in de analyse was de vraag van energie (warmte of koude) vergelijken met het verloop van de temperatuur. In *Figuur 12* is dit 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. De "passieve koeling" (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. Dit verschil tussen de ruimtes is te wijten aan de verschillende plekken. De Maaszaal licht aan de buitenrand van het pand, hierdoor is het gevoeliger voor veranderingen in buitentemperatuur (Zie Figuur 13).

- Om ruimtes te koelen heb je relatief weinig tijd nodig, dit is te zien in *Figuur 14*. 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 17. Temperatuur kantoor BC-IT (blauw) vs sturing koeling per 10%-punten (paars) vs sturing verwarmen per 10%-punten (rood) op 23-03-2020

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/practical-dev/image/fetch/s--UO68-pH3--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https://cdn-images-1.medium.com/max/1024/0*JXOjhOJMSXfFuCvv.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
- fdeloche. (z.d.). Fully recurrent neural network [Foto]. Geraadpleegd van 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>