

MENS VERSUS MACHINE

KAN EEN GENETISCH ALGORITME BETER BELEGGEN DAN EEN MENS?

STACH REDEKER - NOAH VERKAIK - SANDER VERMEULEN



6 VWO WISKUNDE D
ERASMUS COLLEGE ZOETERMEER
BEGELEIDER: M. VAN DE VEN-APPEL
15 DECEMBER 2020

Beeld omslag

Warren Buffet, Amerikaans zakenman en investeerder, voorzitter van Berkshire Hathaway en een van de rijkste personen ter wereld staat tegenover een stereotypische robot. Op de achtergrond is foto van een elektronisch scherm met beursgerelateerde getallen geplaatst.

Voorwoord

Iedereen wil tegenwoordig rijk worden. Wij ook. *The Wolf of Wall Street* is jou, de lezer, allicht bekend en we zouden liegen als we ontkennen dat wij niet door de excentrieke beurshandelaar en meesteroplichter Jordan Belfort, in de film gespeeld door Leonardo DiCaprio, geïnspireerd waren om iets met de aandelenmarkt te doen.

Economie zit niet in onze pakketten, maar dat vormde geen probleem. Een match met wiskunde D werd snel gevormd toen we op een donderdagavond besloten om ook kunstmatige intelligentie in ons onderzoek te verwerken. *AI* is hip en supermodern. Bovendien kwam onze gedeelde passie, computers en programmeren, goed van pas.

Tijdens het maken van ons profielwerkstuk doken we dieper in de zee van *neurale netwerken*, *backpropagation* en *deep learning*. De wiskunde erachter heeft ons verbaasd, verbijsterd en verwonderd. Voor ons is kunstmatige intelligentie nu niet langer een computerprogramma dat invoer op een magische wijze omzet naar een gewenst eindresultaat, maar een systeem van wiskundige vergelijkingen die op een bijzondere manier met elkaar zijn verbonden. We hopen dat jij zal genieten van de eentjes en nulletjes waaruit ons werkstuk is opgebouwd en ons doel is helemaal geslaagd als wij je kunnen fascineren voor de magische wereld achter kunstmatige intelligentie.

Ook willen we graag onze enthousiaste wiskundedocente en begeleider M. van de Ven-Appel bedanken voor haar advies, steun en hulp tijdens het schrijven van ons verslag.

Tot slot: of dit profielwerkstuk een instructie is om een steenrijke multimiljonair te worden, weten wij niet. In elk geval hebben we veel plezier gehad tijdens het onderzoeken, programmeren en schrijven. Uiteindelijk is dat misschien wel het allerbelangrijkste.

Stach Redeker
Sander Vermeulen
Noah Verkaik

15 december 2020, Zoetermeer

Inhoudsopgave

Voorwoord	2
Inleiding.....	8
Onderzoeksraag.....	9
Hypothese.....	11
Opbouw van ons theoriedeel.....	12
Opbouw van ons praktijkdeel.....	12
Topje van de ijsberg.....	14
Dikke vette disclaimer	15
1. (Bijna) alles wat je moet weten over aandelen.....	16
1.1 Initial Public Offering.....	16
1.2 De beurs	16
1.3 Traden of beleggen?.....	17
1.3.1 Flitshandel & FlowTraders	18
1.4 De waarde van aandelen	18
1.5 Indices.....	20
1.6 Volatiliteit & volatiliteitsindices.....	24
1.7 Risico-analyse: aandelen in vergelijking met forex.....	26
2. Analyseren en weten hoe te reageren.....	29
2.1 De fundamentele versus de technische analyse	29
2.2 Interpreteren van grafiekjes.....	29
2.2.1 De onderdelen van een koersgrafiek.....	30
2.2.2 Trends, trendlijnen & patronen	32
2.2.2.1 Bull (opwaartse trend).....	32
2.2.2.2 Bear (neerwaartse trend)	33
2.2.2.3 Helemaal geen trend	35
2.2.2.4 Patronen.....	35
2.2.3 Indicatoren en oscillatoren.....	39
2.2.3.1 Voortschrijdend gemiddelde	39
2.2.3.2 Moving average convergence divergence-indicator (MACD)	42
2.2.3.3 Relatieve Sterkte Index (RSI)	44
2.2.4 Relevantie?	46

2.3 Opties en futures	46
2.4 Verschillende posities	48
3. Geautomatiseerde systemen in de aandelenhandel.....	49
3.1 Conditional orders	49
3.2 Expert advisor (EA).....	50
3.3 Artificiële intelligentie (AI)	51
3.4 Waarom AI op de beurs niet vaak gebruikt wordt, maar wij het wel gaan proberen.....	53
4. Hoe leert een neuraal netwerk?.....	55
4.1 Trainen en trainingsdata.....	55
4.2 Opbouw van een computergestuurd neuraal netwerk.....	56
4.2.1 Input: dit gooien we erin	57
4.2.2 Neuron: de kleinste eenheid van het netwerk.....	58
4.2.2.1 Functies van het gewicht en de bias	60
4.2.3 Activatiefunctie: de invoer verandert in een uitvoer.....	61
4.2.3.1 Het belang van activatiefuncties	62
4.2.4 Enkele laag: samen staan neuronen sterk	64
4.2.5 Tussenlagen: dit is waar de magie gebeurt.....	69
4.2.6 De laatste laag: hier neemt het neurale netwerk zijn beslissing	69
4.2.7 Verliesfunctie & gradient descent: zo leert een neuraal netwerk	71
4.2.8 Backpropagation: een neuraal netwerk leert van zijn fouten.....	74
4.2.8.1 De wiskunde achter backpropagation.....	75
4.3 Samengevat.....	78
5. NEAT: het algoritme dat ons een god maakt.....	79
5.1 Survival of the fittest, maar dan digitaal.....	80
5.2 Hoe werkt die evolutie dan precies?	81
5.3 Een evolutionair algoritme in plaats van gradient descent?	83
6. Programmeerbenodigdheden	85
6.1 Python	85
6.1.1 Libraries.....	85
6.1.1.1 NEAT-python.....	85
6.1.1.2 pickle	86
6.1.1.3 Matplotlib	86
6.1.1.4 xlsxwriter.....	86

6.1.1.5 python-shell.....	86
6.1.1.6 yfinance	86
6.1.1.7 forex_python.....	86
6.1.1.8 utils / gui_utils.....	86
6.1.1.9 pytz.....	86
6.2 Classes en variabelen.....	87
6.3 Loops	87
6.4 Electron	87
7. Praktijk: zo werkt ons product.....	88
7.1 AI-gedeelte	88
7.1.1 Gedefinieerde variabelen	88
7.1.2 Trainingsfase I: de voorbereiding.....	91
7.1.3 Trainingsfase II: het échte trainen.....	91
7.1.4 Testfase	93
7.1.5 Grafieken en data-export.....	93
7.1.6 NEAT-configuratie.....	95
7.2 Het GUI-deel	98
7.2.1 Rood: de balance	99
7.2.2 Oranje: de tijd	99
7.2.3 Geel (1): de vormgeving van de grafiek.....	100
7.2.4 Even tussendoor: de backend.....	100
7.2.5 Lichtblauw: de tickers op een rij	102
7.2.6 Lichtgroen: voeg een ticker toe.....	103
7.2.7 Donkerblauw: naam, waarde, geldeenheid en tijd.....	103
7.2.8 Geel (2): de inhoud van de grafiek.....	107
7.2.9 Paars (1): de knoppen boven de grafiek	108
7.2.10 Paars (2): de backend van de knoppen boven de grafiek.....	108
7.2.11 Roze: indicatoren	110
7.3 Afsluiting.....	112
8. Evaluatie en conclusies.....	113
8.1 AI-versiegeschiedenis.....	113
8.2 Resultaten.....	115
8.2.1 1 januari 2019 - 1 januari 2020 (voor corona).....	116

8.2.1.1 EUR/USD-pair (EURUSD=X), super stabiel	116
8.2.1.2 AEX-index (^AEX), vrij stabiel	117
8.2.1.3 Microsoft (MSFT), bullish	119
8.2.1.4 GameStop (GME), bearish.....	119
8.2.1.5 VIX-index (^VIX), hoge volatiliteit.....	121
8.2.1.6 Just Eat Takeaway (TKWY.AS), hoge volatiliteit.....	122
8.2.2 1 januari 2020 tot 10 december 2020 (met corona)	123
8.2.2.1 EUR/USD-pair (EURUSD=X), super stabiel	124
8.2.2.2 Ahold Delhaize (ADRNY), vrij stabiel.....	124
8.2.2.3 Tesla (TSLA), hevig bullish.....	125
8.2.2.4 Microsoft (MSFT), bullish	127
8.2.2.5 Shell (RYDAF), bearish.....	128
8.2.2.6 Air France-KLM (AF.PA), bearish.....	130
8.2.2.7 VIX-index (^VIX), hoge volatiliteit.....	132
8.2.2.8 Just Eat Takeaway (TKWY.AS), hoge volatiliteit.....	133
8.3 Conclusie.....	134
8.3.1 De hamvraag: is ons programma beter dan een.....	136
8.3.2 Maatschappelijke impact.....	137
8.4 Discussie	137
8.4.1 Terugblik op de hypothese	137
8.4.2 Validiteit en betrouwbaarheid van ons onderzoek	137
8.5 Aanbevelingen voor vervolgonderzoek.....	138
Literatuurlijst.....	140
Appendix I: logboek.....	148
Appendix II: tracksheet.....	156
Appendix III: prestatiegrafiek	160
Appendix IV: compleet overzicht van alle transacties van Stach en een grafiek met zijn rendement over de tijd.....	161
Appendix V: vergrote grafieken hoofdstuk 1	163
Appendix VI: vergrote grafieken hoofdstuk 2	166
Appendix VII: vergrote versie soorten neurale netwerken (towardsdatascience.com).....	176
Appendix VIII: verschillende activatiefuncties met uitleg.....	177
VIII.1 ReLU	177

VIII.2 Leaky ReLU	179
VIII.3 Sigmoid	180
VIII.4 Tanh	182
Appendix IX: screenshots van ons programma	184
Appendix X: Beknopt plan van aanpak PWS	186
X.1 Wie wij zijn	186
X.2 Gekozen onderwerp.....	186
X.3 Waarschijnlijke hoofdvraag.....	186
X.4 Welk vak en welke begeleider	186
X.5 Waarom met z'n drieën	187

Inleiding

Vraag: wat is de overeenkomst tussen de onderstaande foto's?



Afbeelding i.a: een collage van computergegengeneerde portretten. Afbeeldingsbron: samengesteld uit foto's van (This Person Does Not Exist).

Dit lijkt misschien een open deur, maar dat is het niet. Dit zijn geen mensen. Of tenminste, deze personen bestaan niet, en hebben ook nooit bestaan. De gezichten zijn volledig gegeneerd door een computerprogramma. Een computerprogramma dat werkt dankzij Artificial Intelligence (AI).

AI, AI en nog eens AI. Open een willekeurige technieuwswebsite en het begrip vliegt je om de oren. Kunstmatige intelligentie is *trending* en dat is eigenlijk heel begrijpelijk: als computers meer taken

van mensen overnemen, krijgen we het alleen maar gemakkelijker. Er zijn complottheorieën die suggereren dat kunstmatige intelligentie de mensheid uit zal roeien. Aan hen die daarin geloven: wees niet bang, in ons onderzoek beperken we ons tot het ‘nabouwen’ van een menselijk brein met als enige doel betere winstmarges op de beurs te behalen. Om ons ultieme doel, het maken van een zelflerende beleggingsrobot, te bereiken duiken we in dit werkstuk niet alleen in de wereld van de beurs en de beurskoersen, maar ook in de technieken achter *deep learning*, *neurale netwerken* en *backpropagation*.

Onderzoeksraag

We willen onderzoeken of een computerprogramma, dat gebruikmaakt van een *genetisch algoritme*, beter kan beleggen dan een mens. Dat programma willen we zelf gaan schrijven. Uiteindelijk willen we dit programma inzetten om beurskoersen te bekijken en te analyseren. Ons eindproduct kan dan hopelijk volledig autonoom aandelen verhandelen.

Om te onderzoeken of ons computerprogramma beter kan beleggen dan een mens, vergelijken we de resultaten met verschillende spelers in de aandelenmarkt, namelijk:

- a) **De doorgewinterde, zeer succesvolle¹ belegger/trader.** Het is lastig in te schatten hoeveel winst een professionele belegger of trader maakt. Ze staan in elk geval niet te springen om hun salarissen op het internet te gooien. Volgens de Amerikaanse website The Balance is het rendement van een belegger afhankelijk van veel verschillende factoren. Volgens hen is het theoretisch mogelijk om rond de 20% winst per maand te hebben, maar ligt het gemiddelde tussen de 5% en de 15%. Als we aannemen dat een succesvolle belegger 7,5% rendement per maand heeft, komt dit neer op een jaarlijks rendement van **238,18%**. (The Balance, 2020)
- b) **Een beleggingsfonds (ook wel een hedge fund genoemd).** Een beleggingsfonds is een groep beleggers die met geld van hun klanten beleggen. Niet ieder fonds maakt evenveel winst. Er zijn meerdere onderzoeken gedaan naar het gemiddelde rendement van fonds. Een onderzoek van Yale en NYU Stern wijst uit dat over een periode van 6 jaar het gemiddelde rendement 13,6% per jaar is. Dat percentage krijgen investeerders overigens niet uitgekeerd. Beheerders van het beleggingsfonds pakken zo'n 20% van de winst. Onder de streep komt dit dus uit op een rendement van **10,88% per jaar**. (Investopedia, 2020)
- c) **Een AI-beleggingsfonds.** We zullen je later (hoofdstuk 3.4) meer vertellen over deze interessante, maar zeldzame initiatieven. Het AI-fonds NextAlpha claimt een rendement van **40% per jaar** (Kahn J., 2020).
- d) **Diverse indices.** Een index is een soort potje met meerdere aandelen. We geven meer uitleg over indices in hoofdstuk 1. De AEX heeft sinds het ontstaan van de index een gemiddeld rendement van **8,85% per jaar**² (Wikipedia contributors, 2020) en de S&P500 heeft de

¹ Geschat wordt dat 80% (!) van de traders verlies (gemiddeld -36,30%) draait in het eerste jaar. Een groot deel, zo'n 75%, stopt binnen 2 jaar. We spreken hier over de groep traders en beleggers die wel winst maakt, maar onthoud dat dit dus meer uitzondering dan regel is.

² We hebben de koersen van 2020 niet in de berekening opgenomen.

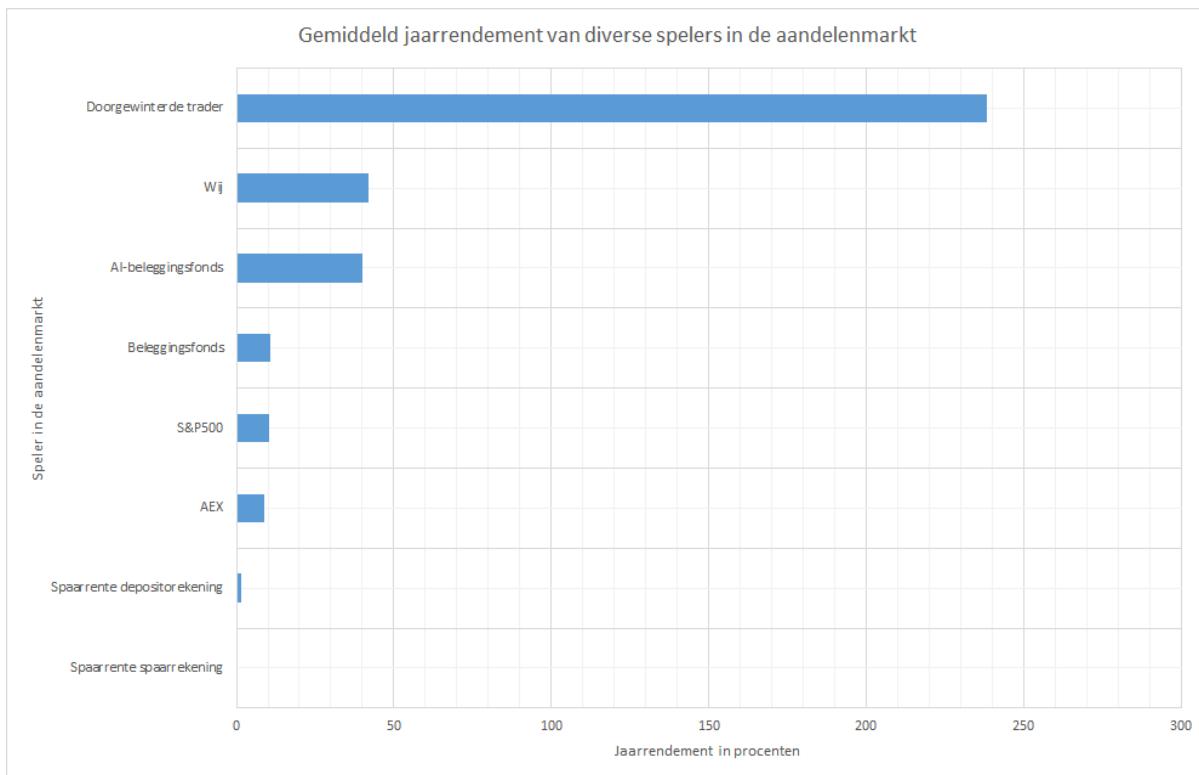
laatste 25 jaar een gemiddeld rendement van **10,22% per jaar**³ (Wikipedia contributors, 2020).

- e) **De spaarrente.** Op dit moment is de hoogste spaarrente op een Nederlandse rekening, de ASN Bank jeugdspaarrekening, **0,55% per jaar**. Als iemand bereid is zijn geld voor langere tijd vast te zetten, belooft de Bigbank depositorekening een rente van **1,50% per jaar**. (ActueleRentestanden, 2020)
- f) **Wij.** Groepslid Stach maakte op 13 juli 2020 een potje in een aandelensimulator aan. Hij startte met € 10.000 en kocht naar eigen inzicht een aantal aandelen. Hij handelde een paar keer, maar hield het grootste deel van de tijd zijn aandelen vast. Een compleet overzicht van al zijn transacties staat afgedrukt in appendix IV, evenals een grafiek met zijn rendement over de tijd. Op 10 december 2020, 151 dagen na het aanmaken van zijn potje, sloot Stach al zijn posities. Hij maakte € 1.733 winst. Dit komt neer op een rendement van 17,33%. Als we (onterecht) de aanname maken dat Stachs rendement lineair zou stijgen, kunnen we zijn verwachte jaarrendement berekenen. Wij komen uit op **41,9% per jaar**⁴. Is dit realistisch? Niet echt, want Stach kocht zijn aandelen tijdens de coronacrisis, precies toen de economie in een zware recessie zat. Toch geeft dit getal wel een redelijke indicatie van de belegprestaties van een mens, want Stach kon niet weten dat de economie zo snel uit de recessie zou klimmen.

We houden in bovenstaande berekeningen geen rekening met inflatie. In onderstaand diagram zijn de verschillende spelers in de aandelenmarkt overzichtelijk weergegeven.

³ Ook hier hebben we de koersen van 2020 niet in de berekening opgenomen.

⁴ Berekening is als volgt: $(1733 \div 151) \cdot 365,25 = € 4191,9 = \text{verwachte jaarwinst}$. $(4191,91 \div 10000) \cdot 100\% = 41,9\% = \text{verwacht jaarrendement}$.



Afbeelding i.b: gemiddeld jaarrendement van diverse spelers in de aandelenmarkt.

Hypothese

We begrijpen dat een programma dat alle patronen van de aandelenmarkt zonder moeite doorziet allicht een utopie is. Immers, als zo'n programma binnen 230 uur door 3 scholieren geschreven kan worden, waarom doet de rest van de wereld dat dan niet ook?

Nou, dat doet de rest van de wereld wel. Er worden verschillende computerprogramma's gebruikt om beurshandel makkelijker te maken. Deze programma's bespreken we in hoofdstuk 3. In datzelfde hoofdstuk leggen we ook uit waarom wij denken dat programma's die specifiek met AI werken, niet vaak voorkomen.

We denken dat het ons gaat lukken om een programma te schrijven dat met behulp van AI beslissingen maakt, maar we vermoeden niet dat ons programma zal zorgen voor veel opschudding in de aandelenwereld. Met name flitshandelaren (hoofdstuk 1.3.1) maken gebruik van krachtigere middelen dan wij tot onze beschikking hebben. Bovendien zal ons programma alleen getallen en grafieken kunnen analyseren, en bijvoorbeeld niet het financiële nieuws kunnen volgen. Dit laatste heet *fundamentele analyse* en bespreken we kort in hoofdstuk 2.1. Fundamentele analyse is een krachtig middel om inzicht te krijgen in de beurs. Een middel dat veel beleggers wel gebruiken, maar ons programma niet.

We verwachten dus geen torenhoge rendementen, maar we hopen een rendement van 20-30% per jaar te behalen. Daarbij komt dat we echt niet een perfect geoptimaliseerd programma hoeven te schrijven, om beter te kunnen beleggen dan een gemiddeld mens. Een mens is immers niet perfect. We denken echter niet dat onze AI beter zal beleggen dan een doorgewinterde trader, want een rendement van meer dan 200% is best aanzienlijk.

Tot slot realiseren we ons dat geluk een behoorlijke rol speelt. Om de geluksfactor zo goed mogelijk uit te sluiten, zullen we ons programma testen met meerdere aandelen. We vertellen meer over ons testmethode in hoofdstuk 8.

Opbouw van ons theoriegedeelte

Ons profielwerkstuk is opgebouwd uit vier delen. In deel één (hoofdstuk 1 en 2) onderzoeken en bespreken we de belangrijkste aspecten van de beurshandel. In deel twee (hoofdstuk 3,4 en 5) duiken we dieper in de wereld van automatisering. Uiteraard besteden we in dit deel vooral aandacht aan het onderwerp van ons profielwerkstuk: kunstmatige intelligentie. Vervolgens leggen we in deel drie (hoofdstuk 6 en 7) uit welke keuzes we op het gebied van programmeren hebben gemaakt. In deel vier (hoofdstuk 8) bespreken we de behaalde resultaten en trekken we de bijbehorende conclusies.

Achtereenvolgens geven we in elk geval antwoorden op onderstaande deelvragen:

- 1) Wat zijn aandelen en hoe zijn ze ontstaan? (hoofdstuk 1.1)
- 2) Wat is een aandelenbeurs? (hoofdstuk 1.2)
- 3) Wat is het verschil tussen traden en beleggen? (hoofdstuk 1.3)
- 4) Hoe wordt de waarde van een aandeel bepaald? (hoofdstuk 1.4)
- 5) Wat zijn indices? (hoofdstuk 1.5)
- 6) Wat is volatiliteit? (hoofdstuk 1.6)
- 7) Hoe kunnen we grafiekjes analyseren? (hoofdstuk 2.2)
- 8) Wat zijn trends, patronen en indicatoren? (hoofdstuk 2.2.2 en hoofdstuk 2.2.3)
- 9) Wat zijn opties en futures? En hoe kunnen we die gebruiken? (hoofdstuk 2.3)
- 10) Welke computersystemen worden op dit moment in de aandelenhandel gebruikt?
(hoofdstuk 3)
- 11) Hoe werkt een neurale netwerk? (hoofdstuk 4)
- 12) Wat is NEAT en waarom gebruiken we het? (hoofdstuk 5)
- 13) Welke programmeertermen zijn belangrijk om te kennen? (hoofdstuk 6)

Opbouw van ons praktijkgedeelte

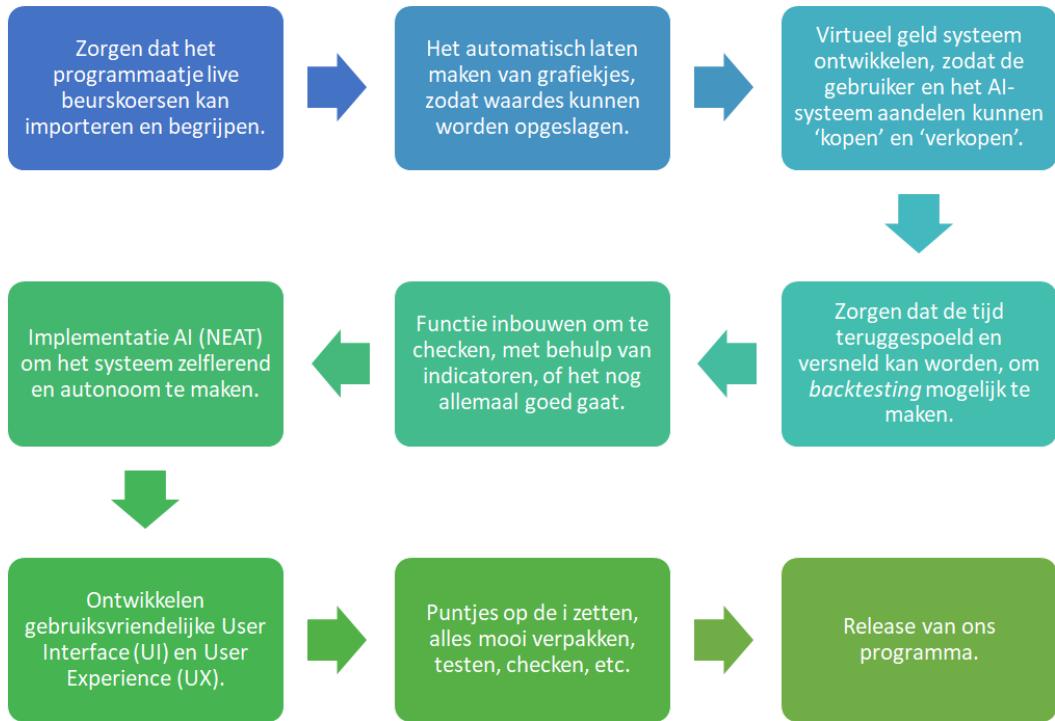
Ons praktijkgedeelte is het schrijven van een programma. Het programma bestaat grofweg uit twee onderdelen: de GUI en de AI.



Afbeelding i.c: het programma bestaat grofweg uit twee onderdelen: de GUI en de AI.

GUI staat voor *graphical user interface*. Hiermee bedoelen we het scherm waar de gebruiker interacties met het programma kan uitvoeren. AI staat voor *artificial intelligence*. Hiermee bedoelen we alle code die het voorspellen van aandelenkoersen mogelijk maakt. Onze focus zal vooral liggen op het stukje over AI, maar we zullen ook enkele interessante elementen uit de GUI toelichten.

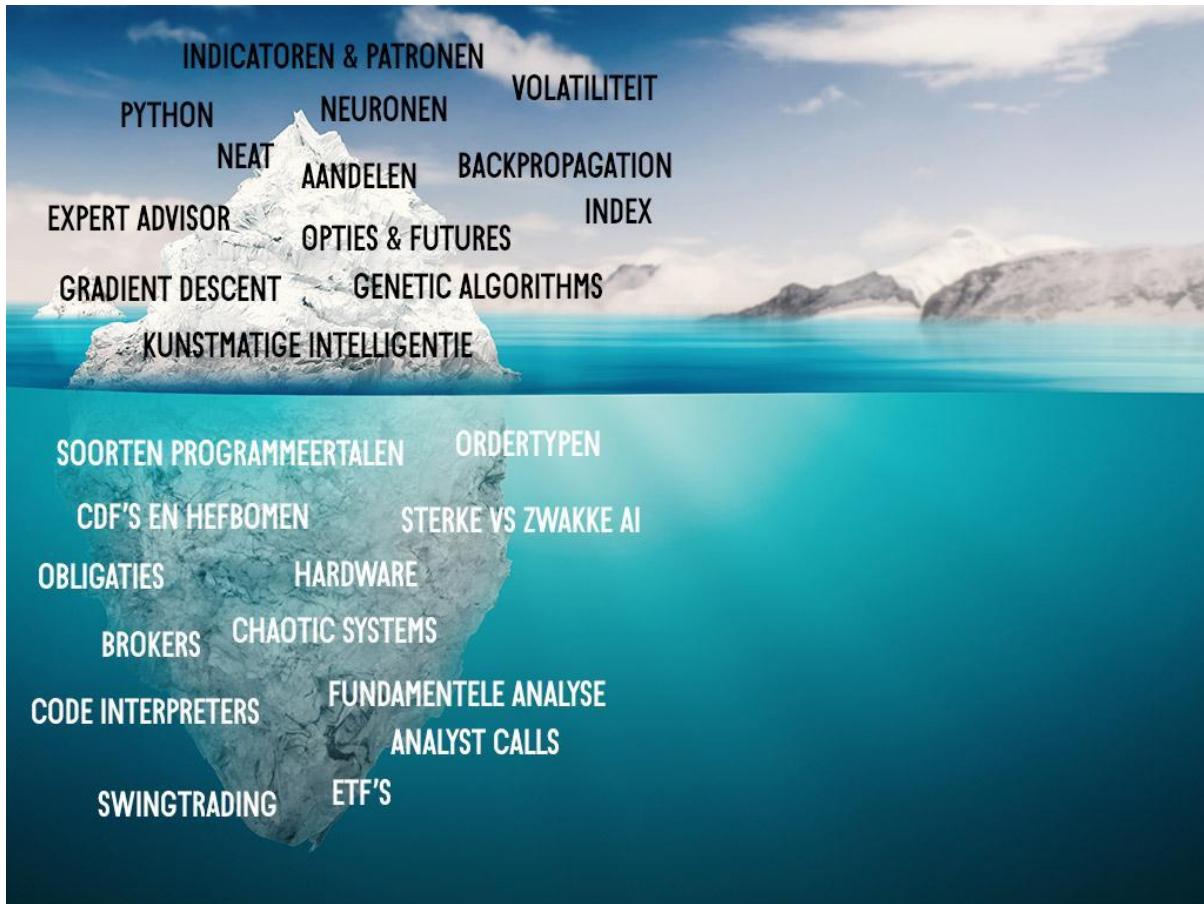
We hebben geprogrammeerd volgens een stappenplan. Dit stappenplan staat weergegeven in afbeelding i.d. In hoofdstuk 7 lichten we onze werkwijze verder toe.



Afbeelding i.d: we hebben geprogrammeerd volgens dit stappenplan.

Topje van de ijsberg

Tijdens ons onderzoek zijn we erachter gekomen dat de wereld van kunstmatige intelligentie, aandelen en programmeren ontzettend groot is. We hebben daarom keuzes gemaakt over wat we wel en niet in ons werkstuk willen bespreken. Wees je bewust dat ons verslag geen compleet overzicht biedt van de aandelen-, AI-, en programmeerwereld. Als er in een hoofdstuk keuzes gemaakt zijn om informatie weg te laten, lichten we deze meestal toe. Afbeelding i.e geeft een aantal begrippen weer die we wel en niet zullen bespreken.



Afbeelding i.e: de begrippen boven het water zullen we in ons profielwerkstuk bespreken. De begrippen onder de zeespiegel zullen we niet behandelen.

Dikke vette disclaimer

De informatie is ons profielwerkstuk is niet te beschouwen als professioneel beleggingsadvies of een vervanging daarvoor. Wij zijn drie enthousiaste scholieren en absoluut geen experts in de aandelenmarkt. Dat een bepaald programma of een bepaalde strategie voor ons werkt of heeft gewerkt, wil niet zeggen dat het ook voor anderen kan werken. Wij geven geen garantie op ons programma en we nemen ook geen verantwoordelijkheid voor eventuele schade die met ons programma wordt gemaakt. Beleggen neemt risico's met zich mee. Je kunt (een deel van) je inzet kwijtraken.

1. (Bijna) alles wat je moet weten over aandelen

In de Gouden Eeuw handelde de Verenigde Oost-Indische Compagnie (VOC) met behulp van meer dan 1700 schepen in goud, porselein, zijde en specerijen. Deze handelsmissies waren prijzige operaties. Om de handelsmissies te bekostigen, investeerden particulieren meer dan 6,4 miljoen gulden (Wikipedia-bijdragers, 2020). In ruil hiervoor kregen de investeerders een deel van de winst. De eerste *aandelen* waren geboren. (Elfenbaum, 2019)

De eerste *effectenbeurzen*, zoals we die vandaag de dag kennen, zijn ontstaan aan het begin van de negentiende eeuw (Wikipedia-bijdragers, 2020). Een (effecten)beurs is een fysieke of digitale plek waar aandelen worden verhandeld. Aandelen zijn waardepapieren die de houder, voor een bepaald percentage, eigenaar maakt van het bedrijf dat de aandelen verstrekkt. Er bestaan verschillende soorten aandelen. Omdat de verschillen tussen de soorten voor ons eindproduct niet van belang zijn, gaan we hier niet verder op in.

1.1 Initial Public Offering

Bedrijven hebben kapitaal nodig voor investeringen en groei. Om dat geld te verkrijgen, kunnen bedrijven leningen afsluiten bij particulieren of bij de bank. Soms is het echter strategisch slimmer om aandelen uit te geven. Bijvoorbeeld als het benodigde bedrag te hoog is, of als een bedrijf te weinig winst maakt om een lening op redelijke termijn terug te betalen. (Ondernemen & Internet, 2019)

Aandelen kunnen beursgenoteerd zijn, maar dat hoeft niet. Als slechts een paar investeerders interesse hebben in het bedrijf, is een beursgang niet direct nodig. Als een bedrijf wel kiest om met aandelen naar de beurs te gaan, bijvoorbeeld om meer investeerders aan te trekken, gebeurt dit via een *Initial Public Offering* (IPO), ook wel primaire emissie genoemd. Het bedrijf bepaalt hoeveel aandelen worden verkocht en wat de prijs per aandeel moet worden. Om meer investeerders te trekken, worden IPO's vaak onder hun geschatte waarde geprijsd (Wikipedia-bijdragers, 2019). Handelaren kunnen vanaf de IPO aandelen kopen en verkopen op de beurs. (BinckBank Nederland, sd)

1.2 De beurs

Een beurs is een plek waar aandelen worden verhandeld. Er bestaan verschillende beurzen. De bekendste is zonder twijfel het Amerikaanse Wall Street. Op Wall Street bevindt zich de *New York Stock Exchange* (NYSE): de grootste aandelenbeurs ter wereld. Alle aandelen van technologiebedrijven worden in de Verenigde Staten via de *National Association of Securities Dealers Automated Quotations* (NASDAQ) verhandeld. De Nederlandse effectenbeurs staat in Amsterdam.

Vroeger, voor de komst van snelle computers, werden aandelen verkocht en gekocht door beleggers die rondliepen in een beursgebouw. Met behulp van *open outcry* – een soort gebarentaal – seinen

handelaren naar elkaar of ze willen kopen of verkopen, en voor welk bedrag. Ondanks dat open outcry overbodig is geworden door computerhandel, is er tot op de dag van vandaag een aantal beurzen waar dit systeem nog steeds wordt gebruikt. (Corporate Finance Institute, 2019)



Afbeelding 1.a met behulp van open outcry kan offline snel en efficiënt worden gehandeld. Afbeeldingsbron: (bizjournals.com).

Tegenwoordig gaat aandelenhandel in de meeste gevallen gewoon met behulp van een computer. Dat brengt nieuwe problemen met zich mee. Computers kunnen veel sneller reageren dan mensen en dat levert oneerlijke situaties op: programma's hebben eerder door dat een koerswijziging plaatsvindt dan een hardwerkende handelaar. De IEX, een beurs in de Verenigde Staten, experimenteert daarom met 61 kilometer aan glasvezelkabel waar dataverkeer van buitenaf verplicht doorheen moet. Zo wordt het verschil tussen supersnelle computer en trage mens een beetje verkleind. (Scott, 2019)

1.3 Traden of beleggen?

Tradent is niet hetzelfde als beleggen. Een belegger koopt aandelen en houdt deze voor een langere tijd vast, terwijl een trader zijn aandelen meestal binnen een dag (en vaak zelfs binnen enkele minuten) weer verkoopt. Beleggers bezitten het liefst veel verschillende soorten aandelen. Zo spreiden ze het risico: als één sector het zwaar heeft, dalen niet per se alle andere aandelen. Voor een trader maakt de diversiteit van zijn/haar aandelenportefeuille minder uit, omdat de aandelen toch niet lang vastgehouden worden. (Epstein & Roze, 2018)

Tot slot investeren beleggers vaak in aandelen waarvan ze denken dat die op de lange termijn in waarde gaan stijgen. Ze hebben dan ook recht op een eventuele *dividenduitkering*. Dat is een

percentage van de winst van het bedrijf waarin de belegger geïnvesteerd heeft (Bos, 2020). Niet ieder bedrijf kiest er echter voor om dividend uit te keren en er zijn weinig regels opgesteld over de hoogte of de frequentie van een dividenduitkering (Wikipedia-bijdragers, 2018). Traders hebben bijna nooit recht op dividend, omdat ze aandelen meteen verkopen en niet voor langere termijn vasthouden.⁵

Ons computerprogramma zal vooral gebaseerd zijn op trading. Daarom besteden we in de rest van ons verslag geen (of heel weinig) aandacht aan dividend en andere strategieën die specifiek op beleggers gericht zijn.

1.3.1 Flitshandel & FlowTraders

Er is een bijzondere categorie traders waar we nog even de nadruk op willen leggen: de *flitshandelaren*. Flitshandelaren kopen en verkopen aandelen in fracties van een seconde. Het grootste en meest bekende flitshandelbedrijf in Nederland heet FlowTraders. Medewerkers van FlowTraders kopen op de beurs soms wel 10.000 aandelen tegelijkertijd. Ze verkopen de aandelen vrijwel direct (in minder dan een seconde) en maken ongeveer één cent winst. Eén cent winst per aandeel klinkt weinig, maar als je alles bij elkaar optelt, levert zo'n transactie $10000 \cdot 0,01 = 100$ euro op. (FlowTraders, sd)

Het zijn natuurlijk geen mensen die deze supersnelle transacties uitvoeren. In een interview met de Financial Study Association Amsterdam zegt medewerker Erik-Jan Nieboer het volgende over flitshandel: "*Algoritmes* spelen hier een steeds grotere rol bij, deze nemen een deel van ons werk over door een prijs aan te bieden, [...] maar er is nog steeds menselijke invloed nodig om alles te controleren en de algoritmes telkens te verbeteren. Ik ben zelf veel in contact met *developers* en *quants*⁶ die wij op de werkvloer hebben. (FSA, 2019)"

Flitshandelaren maken dus gebruik van snelle computers en slimme programma's (algoritmes). De desbetreffende algoritmes waar Erik-Jan Nieboer over praat, gaan we in hoofdstuk 3 behandelen.

1.4 De waarde van aandelen

De waarde van een aandeel is afhankelijk van de waarde van de bijbehorende onderneming. Als een onderneming weinig winst maakt, of weinig eigen vermogen heeft, kan de waarde van een aandeel laag zijn. Een aandeel met een waarde lager dan vijf dollar wordt een *penny stock* genoemd

⁵ Het is voor beginnende traders vaak verleidelijk om vlak voor de dividenduitkering aandelen in het desbetreffende bedrijf te kopen om ze daarna direct weer te verkopen. In de praktijk blijkt dit een zinloze strategie, omdat direct na de dividenduitkering de koers van het bedrijf met de hoogte van de dividend daalt. Dit komt omdat het bedrijf minder geld in kas heeft na het uitkeren van de dividend en de aandelen dan dus ook minder waard zijn.

⁶ *Quants* zijn *quantitative analysts*, ofwel mensen die gebruikmaken van technische analyse. Meer hierover in hoofdstuk 2.

(Securities and Exchange Commission, 2009). Investeren in penny stocks is riskanter dan investeren in aandelen van grotere bedrijven, om de volgende drie redenen:

- 1) De lage koers van een penny stock is een indicatie dat het slecht gaat met een bedrijf. Het bedrijf is niet veel waard en zit soms op het randje van een faillissement. Dit hoeft natuurlijk niet altijd zo te zijn. Als een *start-up* (nieuw bedrijf) aandelen uitgeeft, kan de prijs van een aandeel laag zijn, terwijl het bedrijf wel veel potentieel heeft;
- 2) Penny stocks zijn moeilijk verhandelbaar, omdat ze vaak *over-the-counter*, zonder tussenkomst van een beurs, worden verhandeld;
- 3) Tot slot zorgen *pump-and-dumps scams* voor weinig vertrouwen in goedkope aandelen. Tijdens een pump-and-dump wordt een penny stock door investeerders eerst ingekocht en daarna veelvuldig gepromoot. Hierdoor stijgt de waarde van het aandeel (pump). Daarna verkopen de investeerders al hun aandelen (dump). Hierdoor maken de investeerders in korte tijd veel winst over de ruggen van latere kopers. (reddit.com, 2012)

Niet elk aandeel heeft een lage waarde, integendeel. Aandelen van het Amerikaanse bedrijf Berkshire Hathaway⁷ behoren al jaren tot de duurste ter wereld (Investopedia, 2020). Voor één aandeel wordt rond de 300 duizend dollar betaald⁸. Dat is omgerekend 265 duizend euro⁹. Genoeg om een aardig huis van te kopen. Berkshire Hathaway heeft ongeveer 650.000 aandelen uitgegeven (Yahoo Finance, 2020).

Penny stocks en de aandelen van Berkshire Hathaway zijn natuurlijk uitschieters. Het is lastig om te zeggen hoeveel een aandeel gemiddeld waard is, omdat er simpelweg teveel aandelen bestaan die we dan zouden moeten bekijken. En in de praktijk kunnen prijzen van aandelen hevig fluctueren. Bovendien is de prijs van een aandeel (of een ander financieel product) niet altijd afhankelijk van de waarde van de bijbehorende onderneming. Eigenlijk geldt: aandelen zijn zoveel waard als de prijs waarvoor men ze bereid is te kopen. Dit kan tot situaties leiden waarbij financiële producten hevig in prijs stijgen (of dalen), terwijl ze misschien helemaal niet zoveel (of zo weinig) waard zijn. Dit wordt een *bubbel* genoemd. De *bitcoinbubbel* (afbeelding 1.b) is misschien wel de bekendste, maar ook het Tesla-aandeel (afbeelding 1.c) vertoont kenmerken van een bubbel.

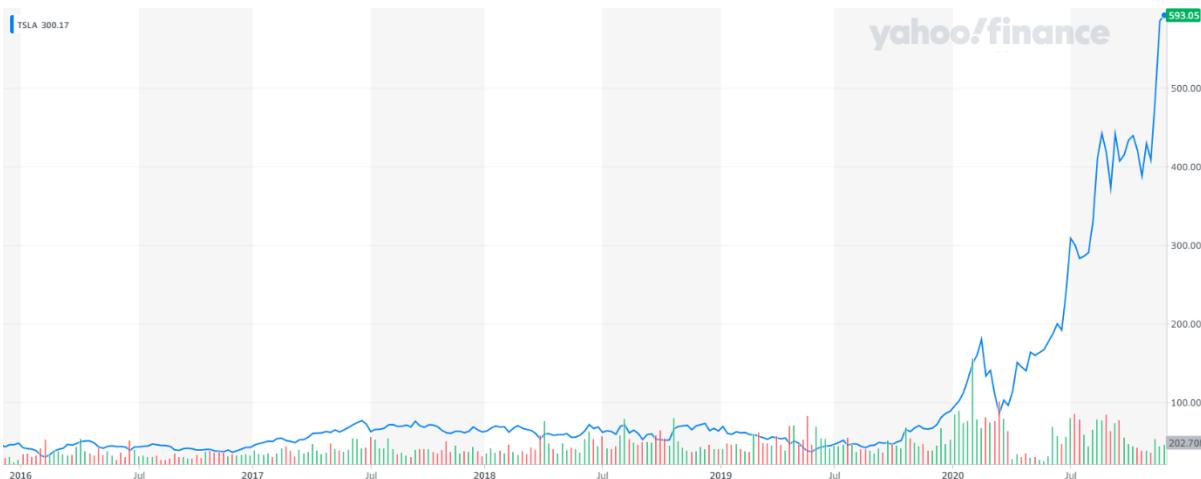
⁷ Niet onbelangrijk: de oprichter van dit bedrijf, Warren Buffet, staat op de cover van ons verslag.

⁸ Een A-klasse aandeel. Er zijn goedkopere B-aandelen.

⁹ De waarde van dit aandeel is zo hoog, omdat er nog nooit een *stock split* (de waarde van een aandeel wordt met een factor n verlaagd, en het aantal aandelen wordt met een factor n verhoogd) heeft plaatsgevonden. Ook keert Berkshire Hathaway geen dividend uit.



Afbeelding 1.b: de koers van de Bitcoin (BTC) van 4 december 2015 tot en met 4 december 2020. De piek van de eerste bubbel lag rond de jaarwisseling 2017/2018. Er is nu (2020) een tweede bubbel zichtbaar. Bron gegevens: (Yahoo Finance, 2020).



Afbeelding 1.c: de koers van Tesla (TSLA) van 4 december 2015 tot en met 4 december 2020. Ook deze grafiek vertoont veel kenmerken van een bubbel. Bron gegevens: (Yahoo Finance, 2020).

1.5 Indices

Een (markt)index is niets meer dan een wiskundig gemiddelde van de koers van een aantal verschillende aandelen. Zo kunnen beleggers en investeerders met één cijfer een beeld krijgen van hoe goed of slecht het met de aandelenmarkt gaat. Dit bespaart tijd en vereenvoudigt het beleggingsproces, omdat er voor handelaren geen noodzaak is om tientallen koersen van verschillende aandelen te monitoren. (Investing With Rose, 2019)

Omdat een index een (complexe) berekening is, en geen ‘echt’ aandeel, wordt de koers van een index niet weergegeven in euro’s of dollars, maar in *punten*.

De S&P 500 is de meest gebruikte index bij beleggers. In deze index zijn de 500 grootste Amerikaanse bedrijven opgenomen. De weging van ieder aandeel is hetzelfde: 0,2 procent (Wikipedia-bijdragers, 2020). Dat leidt tot de volgende berekening:

$$S\&P500 = \frac{\sum(P_i \cdot Q_i)}{D} \quad (1.1)$$

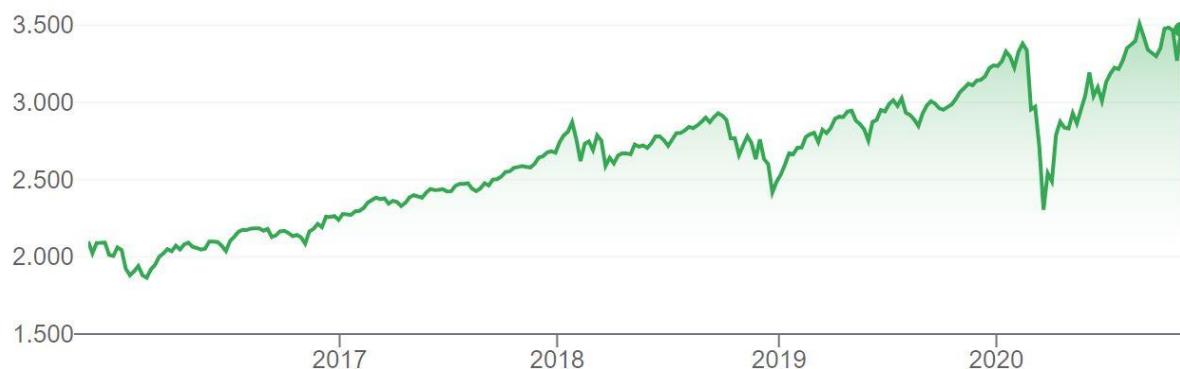
met:

$S\&P500$ = de waarde van de S&P-500-index;

P_i = de prijs van het i -de aandeel in de index;

Q_i = het aantal aandelen van het bedrijf dat hoort bij P_i dat via openbare markten wordt verhandeld;

D = een factor om de waarde van de S&P500 relatief consequent te houden, zodat er bijvoorbeeld geen schommeling plaatsvindt op de dag dat er een nieuw bedrijf aan de index wordt toegevoegd. Op dit moment geldt $D \approx 8,3$ miljard. (Yardeni, Yardeni Research, Inc., & Abbott, 2020)



Afbeelding 1.d: koersgrafiek van de S&P-500 index van 4-11-2015 tot en met 4-11-2020. Gegevens: (Google Finance, 2020).

Een andere bekende, Amerikaanse index is de *Dow Jones Industrial Average*. Deze index maakt gebruik van een *gewogen gemiddelde*: niet ieder aandeel telt even zwaar mee. In deze index zijn 30 bedrijven opgenomen. Het doel van deze index is het creëren van een zo representatief mogelijke weergave van het Amerikaanse bedrijfsleven. De traditionele berekening¹⁰ voor de Dow is als volgt: (Wikipedia-bijdragers, 2020)

¹⁰ Tegenwoordig benadert men de waarde van de Dow ook wel door de koersen van alle aandelen bij elkaar te delen door een factor. Deze factor is per 1 september vastgesteld op $\beta = 0,152$. Deze β is net als de D van de S&P500 bedoeld om schommelingen te voorkomen als gevolg van aandelensplitsingen of nieuwe bedrijven. Wij kiezen voor de traditionele berekening, omdat dan duidelijk wordt dat niet ieder aandeel even zwaar meetelt.

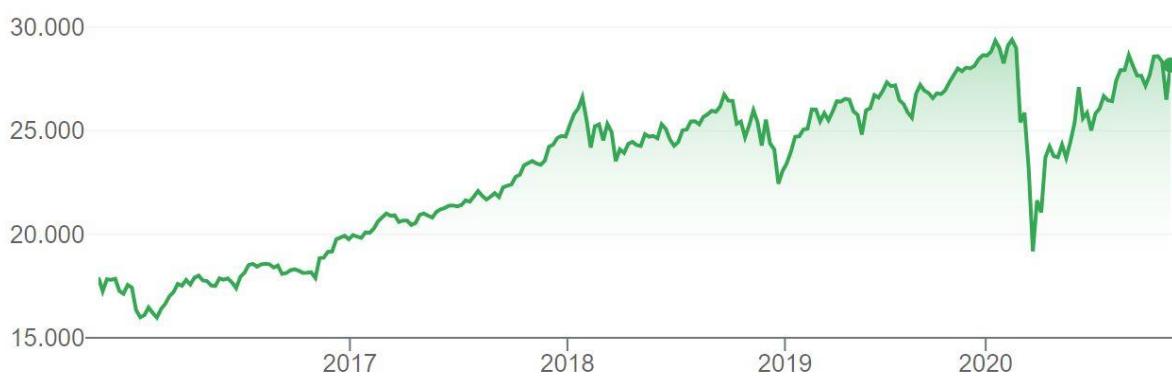
$$Dow = \frac{\sum(P_i \cdot W_i)}{30} \quad (1.2)$$

met:

Dow = de waarde van de Dow-index;

P_i = de prijs van het i -de aandeel in de index;

W_i = een weegfactor die bij het aandeel i hoort. Deze weegfactoren worden per kwartaal bepaald en zorgen ervoor dat aandelen die relatief veel worden verhandeld zwaarder meetellen dan aandelen die minder worden verhandeld.



Afbeelding 1.e: koersgrafiek van de Dow-Jones-index van 4-11-2015 tot en met 4-11-2020. Gegevens: (Google Finance, 2020).

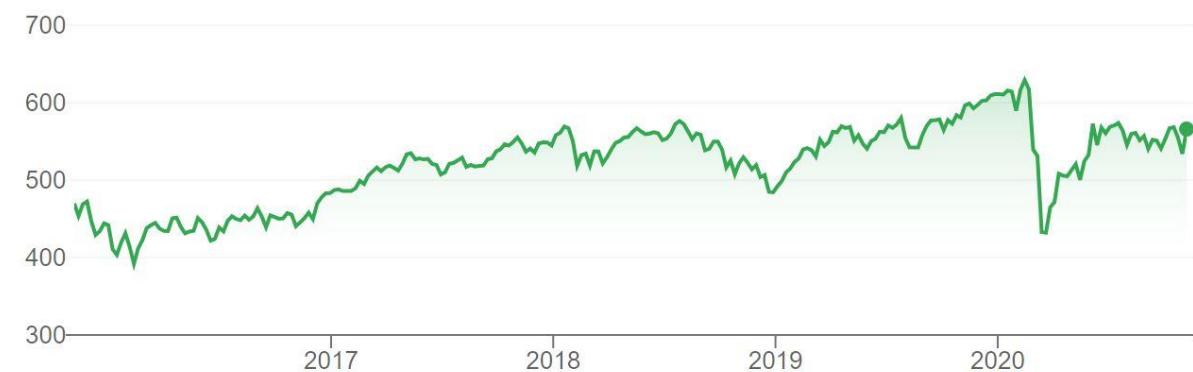
Wat hebben we eigenlijk aan zo'n indexberekening? Een index geeft een representatiever beeld van de economische status van een bepaald land dan een los aandeel. De waarde van een aandeel schommelt meer (dat heet *volatiliteit*, zie hoofdstuk 1.6) dan de waarde van een index en geeft daarom geen goed beeld van de totale economie. De daling van één aandeel hoeft immers niets te betekenen. Een gemiddelde van een aantal belangrijke aandelen kan beleggers daarentegen wel waardevolle informatie verschaffen over de gezondheid van de economie.

Er zijn natuurlijk ook Nederlandse indices. De *Amsterdam Exchange Index (AEX)* is verreweg de bekendste en belangrijkste. De AEX geeft een beeld van de koersontwikkeling van de 25 bedrijven met de hoogste beurswaarde¹¹ op de effectenbeurs in Amsterdam (EuroNext, sd). Een andere bekende Nederlands index is de *Amsterdam Midcap Index (AMX)*. De AMX bevat aandelen van de 25 Nederlandse bedrijven die na de AEX-aandelen de hoogste beurswaarde hebben. Zowel de AEX als de AMX maken gebruik van het gewogen gemiddelde, zoals beschreven bij de berekening van de Dow, om een zo accuraat mogelijke weergave van de markt te vormen. (Wikipedia-bijdragers, 2020)

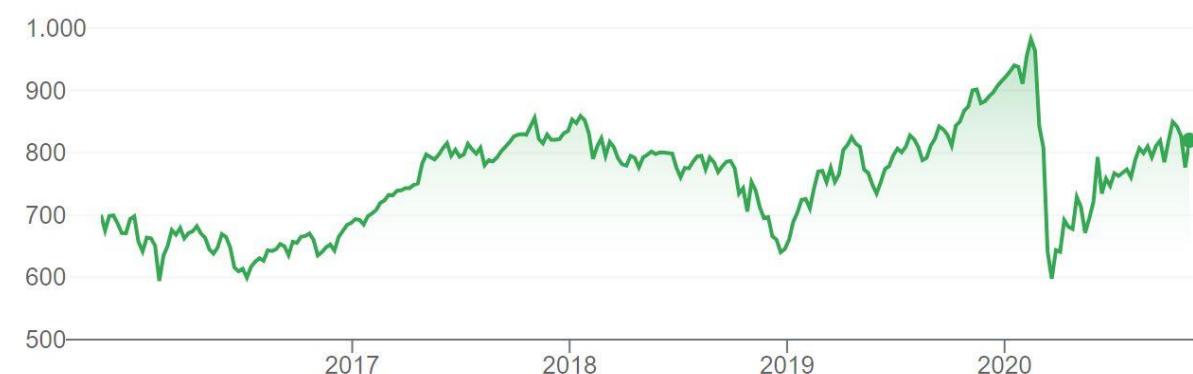
¹¹ *Beurswaarde*: de totale waarde van alle aandelen van een bepaalde bedrijf.

Overigens worden de kleinere bedrijven in Nederland ook niet vergeten. *De Amsterdam Small Cap Index (AScX)* omvat 25 aandelen die na de AEX- en de AMX-aandelen de hoogste beurswaarde hebben. Ook deze index wordt berekend met behulp van het gewogen gemiddelde. (EuroNext, sd)

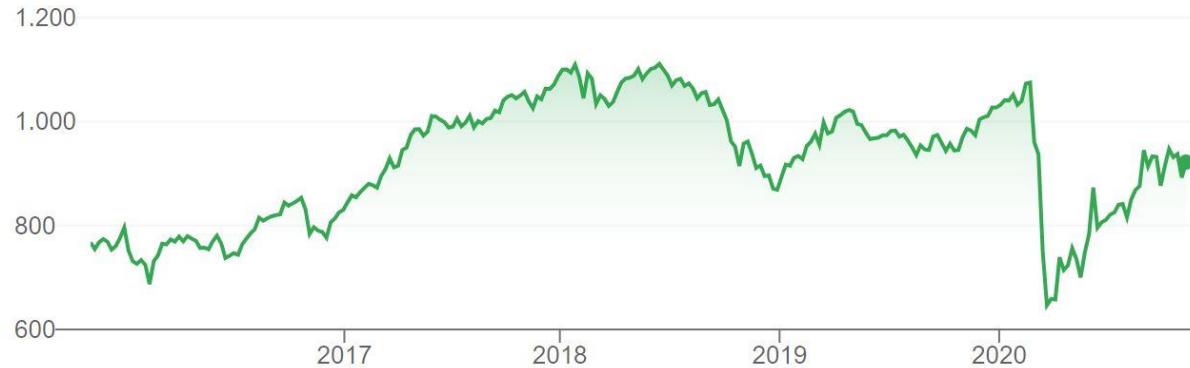
De grafieken van de AEX, AMX en AScX zijn hieronder afgedrukt.



Afbeelding 1.f: koersgrafiek van de AEX-index van 4-11-2015 tot en met 4-11-2020. Gegevens: (Google Finance, 2020).



Afbeelding 1.g: koersgrafiek van de AMX-index van 4-11-2015 tot en met 4-11-2020. Gegevens: (Google Finance, 2020).



Afbeelding 1.h: koersgrafiek van de AscX-index van 4-11-2015 tot en met 4-11-2020. Gegevens: (Google Finance, 2020).

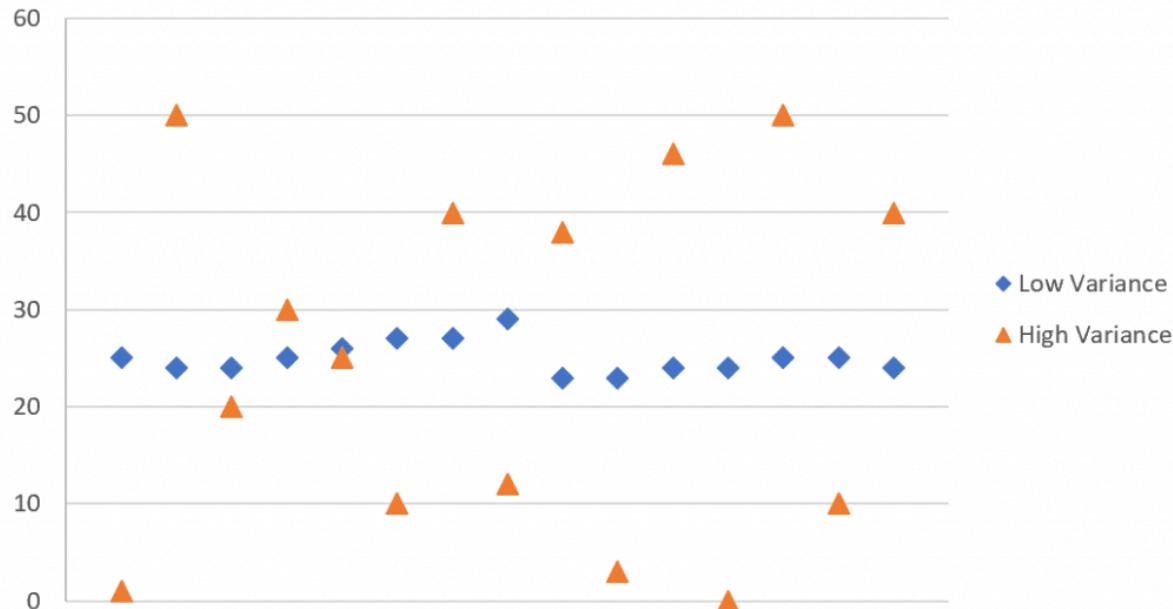
Merk op dat de grafiekjes veel overeenkomsten vertonen. Dit is niet heel gek, want een index geeft per definitie een (representatief) beeld van de totale economie. Omdat de indexen AEX, AMX en AScX gebaseerd zijn op dezelfde economie, die van Nederland, lijken de grafieken erg op elkaar.

1.6 Volatiliteit & volatiliteitsindices

Volatiliteit is een ander woord voor de beweeglijkheid van de koers van een aandeel of index (Encyclo, sd). Indices of aandelen van betrouwbare, grote partijen hebben vaak een lagere volatiliteit dan bijvoorbeeld penny stocks (Wikipedia-bijdragers, 2019). De beweeglijkheid van een index kan worden uitgezet in een andere index: de *volatiliteitsindex*. Zo kennen we bijvoorbeeld de AEXVola en de VIX voor respectievelijk de AEX- en de S&P500-indices. (Beursbrink, 2018)

De waarde van een volatiliteitsindex wordt berekend met behulp van de *standaarddeviatie*. De standaarddeviatie, ook wel standaardafwijking genoemd, geeft aan in welke mate getallen van een reeks zijn verspreid. (Wikipedia-bijdragers, 2020)

Low vs. High Variance (Equal Means)



Grafiek 1.i: De standaarddeviatie van de oranje driehoeken is hoger dan de standaarddeviatie van de blauwe ruiten, maar de gemiddelden van de waarden van de driehoeken en de ruiten zijn gelijk. Afbeeldingsbron: (Langmann).

De gemiddelen van de blauwe en de oranje reeksen in dit assenstelsel zijn hetzelfde, maar de oranje driehoekjes liggen verder uit elkaar dan de blauwe vierkantjes. Bij de oranje reeks is dan sprake van een hogere standaarddeviatie dan bij de blauwe reeks.

$$\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n}} \quad (1.3)$$

Hierbij is

σ de standaarddeviatie en;

x de waarde van een punt en;

\bar{x} het gemiddelde van alle waarden en;

n het totaal aantal waarden.

Met deze kennis kunnen we beter begrijpen hoe de waarde van een volatiliteitsindex tot stand komt.

De berekening voor de VIX is als volgt (Chicago Board Options Exchange, 2019):

$$VIX = 100 \cdot \sigma \quad (1.4)$$

met

$$\sigma^2 = \frac{2}{T} \cdot \sum_i \frac{\Delta K_i}{K_i^2} \cdot e^{RT} \cdot Q(K_i) - \frac{1}{T} \left[\frac{F}{K_0} - 1 \right]^2 \quad (1.5)$$

Bij deze vergelijking leggen we de verschillende variabelen niet uit. Dit komt omdat deze variabelen gebaseerd zijn op opties (zie hoofdstuk 2.3) en veel achtergrondkennis nodig is om deze variabelen te begrijpen. We begrijpen dat de gemiddelde lezer aan een Russische zin misschien wel meer zou hebben. En om eerlijk te zijn: wij snappen ook niet alle *ins en outs* van deze formule.

Het is niet nodig om de formule helemaal te begrijpen om toch een beeld te vormen van wat de VIX-index precies doet. Uit (1.4) blijkt dat de VIX direct afhankelijk is van de standaardafwijking. Een schommelende koers leidt tot een relatief grote standaardafwijking en dus tot een hoge VIX-waarde. Als de koers minder schommelt, daalt de waarde van de VIX-index.

Opvallend is dat de standaarddeviatie¹² σ niet volgens de traditionele wiskundige formule wordt berekend. Dat komt omdat de VIX meer factoren meeweegt dan alleen de positie van bepaalde aandelen (Macroption, sd). Er wordt bijvoorbeeld gekeken naar de verstreken tijd sinds 1 januari en naar de waardes van bepaalde *opties* (wat opties precies zijn, wordt uitgelegd in hoofdstuk 2.3) die aflopen over – vraag ons niet waarom – 16 en 44 dagen. (Investopedia, sd)

Nogmaals: de volledige wiskunde achter de VIX-berekening is te complex om in dit profielwerkstuk te bespreken. Gelukkig hoeft een belegger de waarde van de VIX dan ook niet zelf te berekenen. De Chicago Board Options Exchange (CBOE) regelt de berekeningen. En voor de liefhebbers: op hun site staat een *white paper*, van maar liefst twintig pagina's, met alle details over hun volatiliteitsindex. (Chicago Board Options Exchange, 2019)

1.7 Risico-analyse: aandelen in vergelijking met forex

De zojuist verworven kennis over volatiliteit is cruciaal voor het maken van risicoanalyses. Aandelen met een hoge volatiliteit, waar de koers dus veel van schommelt, zijn riskantere investeringen dan aandelen met een lage volatiliteit. Je kunt immers in één klap veel geld verliezen. Maar schommelende aandelen kunnen ook heel lucratief zijn. Als je aandelen koopt wanneer de prijs laag is, en verkoopt wanneer de prijs hoger is, maak je winst.

¹² Er valt over de discussiëren of de σ hier nog wel standaarddeviatie mag heten. De σ wordt immers niet op de traditionele manier, zoals beschreven in formule 1.5, gedefinieerd. In de whitepaper van de CBOE gebruikt men niet letterlijk het woord 'standaarddeviatie', maar wel de letter sigma. In artikelen over de VIX wordt gewoon over de standaarddeviatie gesproken.

In principe geldt de volgende vuistregel:

Hoe hoger de volatiliteit, hoe meer potentiële winst, maar hoe groter het risico.

Even een voorbeeld. We vergelijken de prijs van een aandeel in Air France-KLM (EPA: AF) met de waarde van de AEX-index (INDEXEURO: AEX). Het grafiekje hieronder toont de geschiedenis van dit aandeel en deze index vanaf 30 oktober 2015 tot 30 oktober 2020.



Figuur 1.j: deze grafiek toont de waarde van de AEX-index (geel) en de prijs van een aandeel Air France-KLM (blauw) over een periode van 5 jaar. De koersen op 30 oktober 2015 noemen we de startprijzen en worden gedefinieerd als 0%. De stapgrootte op de x-as is 1 week. We hebben het percentuele verschil tussen de startprijs en de gemiddelde weekprijs berekend en uitgezet op de y-as. Deze grafiek staat vergroot afgebeeld in appendix V. Bron gegevens: (Yahoo Finance, 2020).

Opvallend is dat de AEX-index een stuk stabieler is dan Air France KLM. Dat is niet heel gek, want schommelingen van de AEX zijn afhankelijk van 25 verschillende bedrijven, terwijl schommelingen in de koers van Air France KLM direct afhankelijk zijn van Air France KLM zelf. Een index is dus meestal een veiligere investering¹³ dan een aandeel, maar doordat de pieken en dalen minder steil zijn, ligt de potentiële winst lager.

Een index is dus een relatieve veilige investering, maar het kan nog veiliger. Op de *forex-markt* (*foreign exchange*) handelt men in valuta's. Ondanks dat er uitzonderingen bestaan, zijn valuta's doorgaans zeer stabiel. Laten we de wisselkoers van euro naar dollar, ook wel het *EUR/USD-pair* genoemd, toevoegen aan onze grafiek:

¹³ Theoretisch gezien is het onmogelijk om te investeren in een index, omdat een index enkel en alleen een rekensommetje is. Als een belegger wil investeren in een index, investeert deze eigenlijk in een *financieel derivaat*. Dat is een 'aandeel' dat exact dezelfde prijs heeft als de waarde van de index.



Figuur 1.k: deze grafiek toont de waarde van de AEX-index (geel), de prijs van een aandeel Air France-KLM (blauw) en de waarde van het EUR/USD-pair (rood) over een periode van 5 jaar. De koersen op 30 oktober 2015 noemen we de startprijzen en worden gedefinieerd als 0%. De stapgrootte op de x-as is 1 week. We hebben het percentuele verschil tussen de startprijs en de gemiddelde weekprijs berekend en uitgezet op de y-as. Deze grafiek staat vergroot afgebeeld in appendix V. Bron gegevens: (Yahoo Finance, 2020).

Uit grafiek 1.k kunnen we concluderen dat de koers van het EUR/USD-pair stabiever is dan de koers van de AEX-index en de koers van een aandeel in Air France-KLM, want de rode lijn is minder volatiel dan de gele en de blauwe. Forex is dus op de lange termijn een zeer stabiele investering. Ook indices zijn relatief stabiel. Wil je als trader snel veel winst maken? Dan moet je kiezen voor losse aandelen. Die hebben de hoogste volatiliteit.

2. Analyseren en weten hoe te reageren

Inmiddels zijn we goed op de hoogte van de geschiedenis van aandelen, hebben de AEX en de S&P-500 geen geheimen meer voor ons en kunnen we iets zeggen over het risico dat aandelen met een hoge volatiliteit met zich meebrengen. We zijn echter nog geen succesvolle traders. Daarom duiken we in dit hoofdstuk nog dieper in de wereld van de aandelen. Dit hoofdstuk bestaat uit twee delen. In het eerste deel bespreken we diverse analysemethoden. In het tweede deel leggen we uit hoe een belegger of trader de analyse kan gebruiken om winst te maken.

2.1 De fundamentele versus de technische analyse

De aandelenwereld onderscheidt grofweg twee analysevormen: *fundamentele* en *technische analyse*. De fundamentele analyse richt zich op de financiële gezondheid van een bedrijf (Bartram, 2017). Analytici, veelal beleggers (en dus geen traders), die voor fundamentele analyse kiezen, kijken bijvoorbeeld naar de haalbaarheid van een bedrijfsplan. Of naar hoe baanbrekend het product is dat wordt geproduceerd. Ook kan fundamentele analyse zich richten op waar we ons nu bevinden in de *economische cyclus*¹⁴.

Traders geven de voorkeur aan technische analyse. Ze zijn niet bezig met het bestuderen van de financiële gegevens en achtergronden van bedrijven, maar kijken naar grafieken, *indicatoren* en *trends* (Epstein & Roze, 2018). Over deze dingen later meer. Er zijn twee vormen van technische analyse: de *visuele methode* en de *kwantitatieve methode* (Wikipedia-bijdragers, 2020). De visuele methode richt zich op patronen en patroonherkenning. De kwantitatieve methode focust op het gebruik van indicatoren. Beide methodes lichten we in dit hoofdstuk nader toe.

Ons programma zal gebaseerd zijn op de technische analyse. Getallen, cijfers en grafieken zijn voor een computer beter begrijpbaar dan een bedrijfsplan. Daarom zullen we in de rest van dit hoofdstuk de nadruk leggen op aspecten van de technische analyse.

2.2 Interpreteren van grafiekjes

Beurzen en grafiekjes zijn onlosmakelijk met elkaar verbonden. Zet de televisie op een financieel kanaal en je wordt gegarandeerd met lijnen, trends en staafdiagrammen geconfronteerd. Sommige mensen vinden grafiekjes eng, ingewikkeld of onbegrijpelijk. Maar grafieken zijn essentieel om de koers van aandelen in de gaten te houden. Grafieken laten zich makkelijker lezen dan reeksen getallen en geven in een oogopslag een totaaloverzicht van het verloop van aandelenkoersen.

¹⁴ Met de economische cyclus doen we op de *conjunctuur* (*fluctuatie*, beweeglijkheid) van het groeipercentage van de economie. Het groeipercentage over de tijd blijft namelijk niet constant, maar schommelt. Deze schommelingen zijn te verklaren door verschillende economische fasen, waaronder *regressie*, *crisis* en *herstel*. De theorie hierachter is te uitgebreid om in ons profielwerkstuk te bespreken. We gaan er dan ook niet verder op in. (Wikipedia-bijdragers, 2020)

2.2.1 De onderdelen van een koersgrafiek

Laten we nogmaals kijken naar de grafiek van een aandeel Air France-KLM. De grafiek hieronder toont de prijs van een aandeel tussen 21 november 2019 en 21 november 2020.



Figuur 2.a: de koers van een aandeel Air France-KLM, bekeken vanaf 21 november 2019 tot en met 21 november 2020. Bron gegevens: (Yahoo Finance, 2020).

De grafiek hierboven wordt een koersgrafiek genoemd. Best logisch, want de koers van het aandeel is in de grafiek afgebeeld. Een koersgrafiek heeft een vaste opbouw:

- Op de horizontale as is de tijd uitgezet;
- Op de verticale as is de prijs uitgezet. De eenheid van de prijs is in dit geval euro, omdat het aandeel op een Europese beurs verhandeld wordt;
- De blauwe lijn geeft de waarde van het aandeel weer. In dit geval is die eind 2019 ongeveer 10 euro en eind 2020 ongeveer 4 euro;
- De groene en rode staven boven de horizontale as geven het *tradingsvolume* weer. De staaf is groen als de *slotkoers* (koers bij sluiten van de beurs) van het aandeel hoger is dan de slotkoers van de dag ervoor. De staaf is rood als de slotkoers lager is dan de dag ervoor. Het tradingsvolume is het totale aantal aandelen dat op die dag is verhandeld. Het tradingsvolume geeft aan hoeveel vraag er is naar een aandeel. Koersveranderingen bij een laag volume betekenen voor technisch analisten minder dan koersveranderingen bij een hoger volume. Immers, bij een laag volume hebben relatief weinig beleggers 'interesse' in dat aandeel. Bij een hoog volume willen juist veel mensen handelen in dat aandeel. Een aandeel waarin veel mensen handelen is voor traders interessanter dan een aandeel waar minder mensen in handelen. Daarom is het tradingsvolume belangrijk.

Er zijn nog twee andere koersgrafieken die we graag in ons profielwerkstuk willen bespreken, omdat ze regelmatig voorkomen.



Figuur 2.b: de koers van een aandeel Air France-KLM, weergegeven in een candlestickgrafiek. Bron gegevens: (Yahoo Finance, 2020).

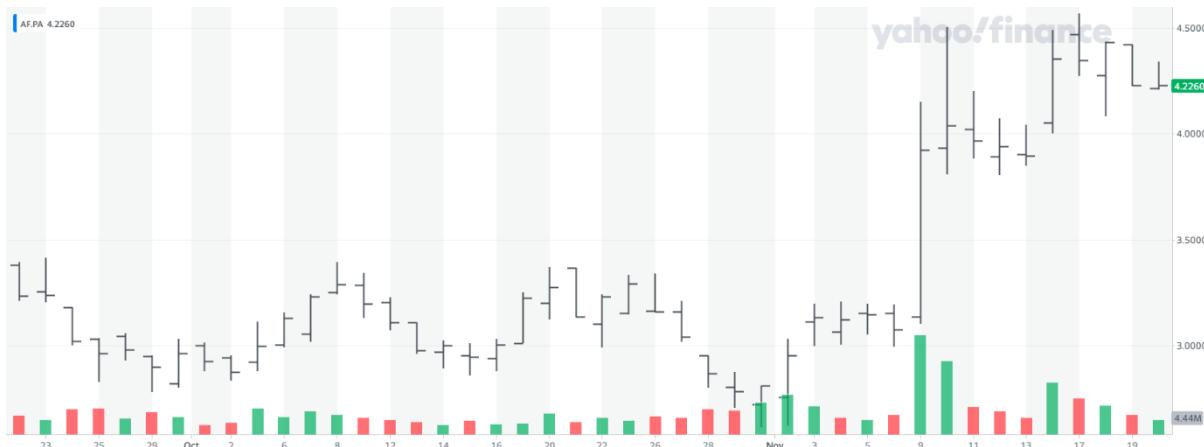
Deze grafiek wordt ook wel een *candlestickgrafiek* genoemd, omdat de staven – met veel fantasie – de vorm van een kaars hebben. Eén candlestick is altijd op de volgende manier opgebouwd:



Afbeelding 2.c: een candlestick. Afbeeldingsbron: (CMC Markets, sd).

- Het onderste puntje (low) is de laagste koers van de gekozen tijdseenheid;
- Het hoogste puntje (high) is de hoogste koers van de gekozen tijdseenheid;
- De koers bij het openen van de beurs is een horizontaal streepje en de slotkoers is ook een horizontaal streepje;
- De kleur is direct afhankelijk van de open- en slotkoers van een aandeel. Als de slotkoers hoger ligt dan de openingskoers is het balkje groen. Het aandeel is die tijdseenheid dan meer waard geworden. Als de slotkoers lager ligt dan de openingskoers is het balkje rood. Het aandeel is die tijdseenheid dan minder waard geworden. De begrippen *bullish* en *bearish* leggen we later uit. (Wikipedia-bijdragers, 2019)

En tot slot: de *barchart* (figuur 2.d).



Figuur 2.d: de koers van een aandeel Air France-KLM, weergegeven in een barchart. Bron gegevens: (Yahoo Finance, 2020).

De barchart is op precies dezelfde manier opgebouwd als de candlestick, maar is niet gekleurd. Het bovenste puntje is de hoogste koers die die tijdseenheid is waargenomen, het onderste puntje de laagste. Het streepje links van een verticale lijn is de openingskoers en het streepje rechts van de verticale lijn is de slotkoers. (Epstein & Roze, 2018)

We hebben in deze paragraaf enkele koersgrafieken bekeken en diverse grafiekonderdelen besproken. Grafieken zijn handige middelen om in één oogopslag het koersverloop van een aandeel te bekijken. In ons programma zal een mogelijkheid worden ingebouwd om grafieken te laten genereren op basis van historische koersdata, maar die grafieken zullen minder geavanceerd zijn dan de grafieken die we kunnen maken met behulp van Yahoo Finance.

2.2.2 Trends, trendlijnen & patronen

Nu we de grafiekjes onder de knie hebben, kunnen we er wiskundige berekeningen en analyses op loslaten. De simpelste analysevorm is het herkennen van trends en patronen. Deze analysemethode wordt ook wel de visuele methode genoemd.

2.2.2.1 Bull (opwaartse trend)

Laten we kijken naar grafiek 2.e. Dit is de koers in dollar van een aandeel in Alphabet, het moederbedrijf van Google.



Figuur 2.e: de koers van een aandeel in Alphabet, van maart 2020 tot en met november 2020. In de grafiek is een trendlijn getekend. Bron gegevens: (Yahoo Finance, 2020).

Als we de laagste dagwaarde van twee dagen vergelijken, kunnen we een trendlijn opstellen.

Er geldt dan:

$$\text{richtingscoëfficiënt } \alpha = \frac{\Delta \text{dagwaarde}}{\Delta \text{dagen}} \quad (2.1)$$

In ons geval (de paarse lijn) komt dat op het volgende neer:

$$\alpha = \frac{1758,57 - 1113,36}{160} = \frac{645,21}{160} = 4,033 \quad (2.2)$$

Deze richtingscoëfficiënt is (sterk) positief. De trendlijn is dus stijgend. Zo'n opwaartse trend wordt ook wel een *bulltrend* genoemd. Het desbetreffende aandeel is dan *bullish*.

2.2.2.2 Bear (neerwaartse trend)

Laten we nu nogmaals kijken naar de koers van een aandeel Air France-KLM in euro's.



Figuur 2.f: de koers van een aandeel Air France-KLM van september 2019 tot en met november 2020. In de grafiek is een trendlijn getekend. Bron gegevens: (Yahoo Finance, 2020).

Ook voor deze grafiek kunnen we de richtingscoëfficiënt van de trendlijn berekenen:

$$\alpha = \frac{4,2260 - 11,0460}{270} = \frac{-6,82}{270} = -0,025 \quad (2.3)$$

Hier ontdekken we meteen een van de limitaties van trendlijnen. Ze geven soms een vertekend beeld: -0,025 lijkt een zeer minimale daling, maar in werkelijkheid daalt de koers over deze periode met maar liefst 62,12%.

Een goed alternatief voor de richtingscoëfficiënt is dus de hellingsgraad, uitgedrukt in procenten:

$$\text{hellingsgraad} = \frac{\text{nieuwe prijs} - \text{oude prijs}}{\text{oude prijs}} \cdot 100\% \quad (2.4)$$

Als we deze formule invullen voor de koers van Air France-KLM, krijgen we de volgende waarde:

$$\frac{4,2260 - 11,0460}{11,0460} \cdot 100\% = -62,12\% \quad (2.5)$$

Dit geeft een representatiever beeld van de grootte van de koersdaling. We kunnen dus concluderen dat de waarde van de richtingscoëfficiënt ons vertelt **of** een aandeel daalt en de waarde van de hellingsgraad ons vertelt **hoe hard** een aandeel daalt.

Een neerwaartse trend wordt ook wel een *beartrend* genoemd. Het aandeel heet dan *bearish*.

2.2.2.3 Helemaal geen trend

Het kan natuurlijk ook voorkomen dat er geen duidelijke trend in een koersverloop zichtbaar is. Dit is het geval in figuur 2.g hieronder.



Figuur 2.g: de koersgrafiek van computerbedrijf Intel (INTC) van eind 2019 tot eind 2020. In de grafiek zijn quadrant lines getekend. Bron gegevens: (Yahoo Finance, 2020).

Het verloop van deze grafiek wordt *trendloos* genoemd. De koers fluctueert rond een bepaalde middellijn, maar vertoont geen kenmerken van een bull- of beartrend. Het is wachten totdat de koers *uitbreekt*. Met een uitbraak wordt een relatief grote stijging of daling bedoeld. (Epstein & Roze, 2018)

Het paarse gebied is een verzameling van *quadrant lines*. De vijf lijnen verdelen een bepaald gebied in vier stukken. De bovenste lijn (ook wel de *weerstand* genoemd) geeft de hoogste koers in een bepaald domein weer. De onderste lijn (ook wel de *ondersteuning* genoemd) is verantwoordelijk voor het aangeven van de laagste koers. De drie lijnen ertussen verdelen het complete paarse gebied in vier gelijke stukken: de kwadranten. (ChartSchool, sd)

Als de onderlinge afstand tussen de quadrant lines groot is, is de standaarddeviatie van de koersen ook groot. Erg geldt: hoe kleiner de onderlinge afstand, hoe kleiner de standaarddeviatie en dus hoe minder volatiel het desbetreffende aandeel is.

2.2.2.4 Patronen

Er zijn een aantal *patronen* die we graag in ons profielwerkstuk willen bespreken. Patronen zijn vormen in koersgrafieken die kunnen duiden op een aankomende bull- of beartrend.

Als eerste: de *cup & handle*. De traditionele vorm van de cup & handle wijst op een aankomende bulltrend (stijgende koers). Dit patroon heeft een vorm die lijkt op een koffiekopje: eerst een lichte, cirkelvormige daling die vervolgens weer iets stijgt (cup) en daarna een neerwaartse trend (handle). Hierna volgt een (vaak hevige) uitbraak. Zie afbeelding 2.h. (Chen, 2020)



Afbeelding 2.h: een voorbeeld van een cup & handle-patroon. Afbeeldingsbron: (Killian, 2019).

Er bestaan allerlei theorieën over waarom dit patroon meestal¹⁵ tot een bulltrend leidt. Er is veel achtergrondkennis van aandelen en de aandelenmarkt nodig om deze theorieën te begrijpen. Hoe het cup & handle-patroon precies tot stand komt, is voor ons eindproduct niet relevant. Daarom gaan we er in ons profielwerkstuk niet dieper op in.

Een *omgekeerde cup & handle* komt ook voor. Dit patroon heeft de vorm van een omgekeerd koffiekopje. Zie afbeelding 2.i. Na deze vorm volgt meestal¹⁶ een daling van het aandeel: een beartrend dus. (Killian, 2019)

¹⁵ Er zijn uitzonderingen bekend.

¹⁶ Ook hier geldt: er zijn uitzonderingen bekend.



Afbeelding 2.i: een voorbeeld van een omgekeerd cup & handle-patroon. Afbeeldingsbron: (Killian, 2019).

Twee andere interessante patronen zijn de *dubbele top* en de *dubbele bodem*. We gebruiken nogmaals de koers van een aandeel Air France-KLM om deze patronen weer te geven. Zie afbeelding 2.j.



Afbeelding 2.j: koersgrafiek van een aandeel in Air France-KLM van november 2015 tot april 2016. Bron gegevens: (Yahoo Finance, 2020).

Als eerste: de dubbele top. Dit patroon is te herkennen – en de naam zegt het eigenlijk al – aan twee ‘toppen’. Als de koers eerst stijgt, daarna daalt en vervolgens weer stijgt, zoals in afbeelding 2.k het

geval is, is er sprake van een dubbele top. Een dubbele top heeft meestal¹⁷ een beartrend als gevolg. Deze beartrend wordt in afbeelding 2.k aangegeven door de meest rechter trendlijn.



Afbeelding 2.k: koersgrafiek van een aandeel in Air France-KLM van november 2015 tot april 2016. Met behulp van trendlijnen is een dubbele top aangegeven. Bron gegevens: (Yahoo Finance, 2020).

Een omgekeerde versie van dit patroon bestaat ook: de dubbele bodem. Als er tweemaal een daling is, ontstaat er daarna meestal een bulltrend. Deze bulltrend geven we in figuur 2.l aan met de meest rechter trendlijn.



Afbeelding 2.l: koersgrafiek van een aandeel in Air France-KLM van november 2015 tot april 2016. Met behulp van trendlijnen is een dubbele bodem aangegeven. Bron gegevens: (Yahoo Finance, 2020).

Ook voor deze patronen geldt dat de economisch achtergrond uitgebreid en ingewikkeld is. We kiezen ervoor om deze niet op te nemen in ons verslag.

¹⁷ Er zijn uitzonderingen.

2.2.3 Indicatoren en oscillatoren

De hierboven besproken trends zijn niet altijd goed te herkennen. Daarnaast kunnen patronen multi-interpretabel zijn. Om deze problemen te verhelpen, zijn kwantitatieve analysemethodes bedacht. In deze paragraaf bespreken we de belangrijkste twee vormen van kwantitatieve analyse: *indicatoren* en *oscillatoren*.

Indicatoren zijn getallen, lijnen of grafiekjes die het interpreteren van een koersgrafiek minder subjectief maken. Dankzij indicatoren kunnen traders, zonder dat hun gevoel een beslissing kan beïnvloeden, een redelijk accurate voorspelling doen over de koers van een aandeel.

Een oscillator is volgens Wikipedia: "Een indicator die gebruikt wordt voor technische analyse en die schommelt boven en onder een middellijn, of tussen twee vastgestelde waardes" (Wikipedia contributors, 2020). Volgens die definitie zijn de MACD en de RSI die we later gaan bespreken, dus oscillatoren. Omdat een oscillator een subtype van een indicator is, noemen wij de MACD en de RSI gewoon indicatoren.

In deze paragraaf vatten we de werking van drie belangrijke indicatoren samen. We beginnen met de meestgebruikte: het voortschrijdend gemiddelde.

2.2.3.1 Voortschrijdend gemiddelde

De eerste indicator die we bespreken is de *simple moving average (SMA)*. In het Nederlands wordt deze indicator een *voortschrijdend gemiddelde* genoemd. Het is een lijn die de richting en de kracht van een trend in een bepaalde periode laat zien. Het voortschrijdend gemiddelde is een gemiddelde van de koersstijgingen en -dalingen. Dankzij het voortschrijdend gemiddelde wordt het identificeren van een trend minder subjectief. De trader moet nog wel zelf aan de bak, want de indicator kan geen verandering voorspellen. (Epstein & Roze, 2018)

In figuur 2.m is de paarse lijn een SMA-indicator bovenop de koers van een aandeel Air France-KLM (blauw).



Figuur 2.m: grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA-indicatorlijn getekend. Bron gegevens: (Yahoo Finance, 2020).

Eén punt op de *SMA-indicatorlijn* wordt volgens deze formule bepaald:

$$SMA_n = \frac{(P_x + P_{x-1} + P_{x-2} + \dots + P_n)}{N} \quad (2.7)$$

Waarbij:

SMA_n = de waarde van de SMA-indicator op dag n ;

P_n = de waarde van het aandeel op dag n ;

P_x = de waarde van het aandeel op x dagen voor dag n ;¹⁸

N = het aantal dagen dat we totaal bekijken.

Uit deze vergelijking komt een punt $P(SMA,n)$ rollen. Als we deze berekening maken voor diverse dagen/momenten, kunnen we een lijn trekken door de verkregen punten. Deze lijn wordt ook wel de SMA-indicatorlijn genoemd en is dus de paarse lijn in figuur 2.m. (Epstein & Roze, 2018)

Er zijn diverse veelgebruikte variaties op de SMA-indicatorlijn. Deze functies hebben geen gigantische meerwaarde, maar we bespreken ze kort omdat ze regelmatig voorkomen. In figuur V staat een *SMA-indicatorlijn met onzekerheidsmarge* afgebeeld.

¹⁸ Een trader kan zelf een keuze maken hoeveel dagen hij voorafgaand aan dag n bekijkt. 20, 50 en 100 dagen zijn het meest gebruikelijk. In onze voorbeeldgrafieken nemen we $x = 50$ dagen.



Figuur 2.n: grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA-indicatorlijn met onzekerheidsmarge getekend. Bron gegevens: (Yahoo Finance, 2020).

De middelste lijn is gelijk aan de SMA-indicatorlijn die we zojuist hebben besproken. De bovenste en onderste lijn zijn nieuw. Zij worden tezamen ook wel een *envelop* genoemd. De waardes van de bovenste en onderste lijn worden als volgt gedefinieerd (StockCharts, sd):

$$\text{Bovenste lijn} = \text{SMA}_n + (\text{SMA}_n \cdot p) \quad (2.8.1)$$

$$\text{Onderste lijn} = \text{SMA}_n - (\text{SMA}_n \cdot p) \quad (2.8.2)$$

Hier geldt:

SMA_n = de waarde van de SMA-indicator op dag n ;

p = de onzekerheidsmarge. In onze voorbeeldgrafiek 2.n hebben we gekozen voor $p = 0,05$.

En tot slot kennen we ook nog een *exponentieel voortschrijdend gemiddelde (EMA)*. Deze indicator wordt berekend volgens onderstaande formule.

$$\text{EMA}_n = (W_n \cdot K) + (\text{EMA}_{n-1} \cdot (1 - K)) \quad (2.9)$$

$$\text{met } K = \frac{2}{N+1}$$

En er geldt:

EMA_n = de waarde van de EMA-indicator op dag n ;

W_n = de (slot)koers van dag n ;

EMA_{n-1} = de waarde van de EMA-indicator op een dag voor n ;

N = het aantal dagen dat we in totaal bekijken.

Als we deze formule uitzetten in een grafiek levert dat voor het aandeel van Air France-KLM de volgende rode lijn op:



Figuur 2.o: grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA- en een EMA-indicatorlijn getekend. Bron gegevens: (Yahoo Finance, 2020).

Er zijn redenen waarom een trader liever een EMA in plaats van een SMA gebruikt. De EMA ligt meestal dichter bij de huidige slotkoers. Hierdoor verandert de waarde van de EMA sneller dan de waarde van een SMA-indicator. Trendveranderingen op de korte termijn worden zo sneller zichtbaar. Een reden om voor SMA in plaats van EMA te kiezen, is omdat de SMA minder gegevens nodig heeft. Daarnaast is de SMA makkelijker te berekenen, al laten veel traders hun grafieken door de computer tekenen. (Epstein & Roze, 2018)

2.2.3.2 Moving average convergence divergence-indicator (MACD)

Een andere veelvoorkomende indicator is de *moving average convergence divergence-indicator*. Omdat dat nogal veel woorden zijn, gebruiken we vanaf nu af aan de officiële afkorting MACD.

De MACD is een aparte grafiek onder de koersgrafiek en wordt op de volgende manier opgesteld:

Voor de *MACD-lijn* (de paarse lijn in figuur 2.o) geldt:

$$MACD = EMA_{12} - EMA_{26} \quad (2.10)$$

met:

$MACD$ = de waarde van de MACD voor een bepaalde dag x ;

EMA_{12} = de waarde van de EMA-indicator op dag x , bekeken ten opzichte van 12 dagen (dus $N = 12$);

EMA_{26} = de waarde van de EMA-indicator op dag x , bekeken ten opzichte van 26 dagen (dus $N = 26$).

En voor de *signaallijn* (de oranje lijn in figuur 2.o) geldt:

$$SIGN = EMA_{MACD_9} \quad (2.11)$$

met:

$SIGN$ = de waarde van de signaallijn voor een bepaalde dag x ;

EMA_{MACD_9} = de waarde van de EMA-indicator op dag x , niet met de koers, maar de zojuist berekende $MACD$ als input, bekeken ten opzichte van 9 dagen (dus $N = 9$).

Deze twee lijnen kunnen we plotten in een aparte grafiek. Figuur 2.p is een voorbeeld van een grafiek met daaronder een MACD-indicator.



Figuur 2.p: grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020 met daaronder een MACD-indicator. Bron gegevens: (Yahoo Finance, 2020).

Een oplettende lezer ziet dat er een histogram in de grafiek van de MACD getekend staat. Het histogram wordt gebruikt om het verschil tussen de MACD- en signaallijn te laten zien.

Voor het histogram geldt:

$$MACD > SIGN \rightarrow \text{positief histogram} \quad (2.12.1)$$

$$MACD < SIGN \rightarrow \text{negatief histogram} \quad (2.12.2)$$

$$MACD = SIGN \rightarrow \text{geen histogram} \quad (2.12.3)$$

Wat hebben we hier allemaal aan? Nou, de MACD biedt enorm veel informatie in een zeer compacte grafiek. Uit figuur 2.p blijkt dat de MACD oscilleert rond de nullijn. Hierom is de MACD een goede indicator voor de richting van een trend.

Om dit te begrijpen, kijken we nogmaals naar de berekening van de MACD:

$$MACD = EMA_{12} - EMA_{26} \quad (2.13)$$

Hoe minder dagen we bekijken met de EMA-indicator, hoe dichter de waarde ervan bij de daadwerkelijke koers zal liggen. We weten dus dat de EMA_{26} ‘minder accuraat’ is dan de EMA_{12} . Bij een dalende koers (bearish) zal de EMA_{12} eerder dalen dan de EMA_{26} . Er geldt dus $EMA_{26} > EMA_{12}$. De uitkomst van de MACD is dan negatief.

Bij een stijgende koers (bullish) is dit precies omgekeerd. Er geldt dan $EMA_{26} < EMA_{12}$. Hieruit volgt dat de uitkomst van de MACD in dat geval positief is.

Samenvattend geldt:

$$\text{Bearish} \rightarrow MACD < 0 \quad (2.14.1)$$

$$\text{Bullish} \rightarrow MACD > 0 \quad (2.14.2)$$

Waar is de signaallijn dan goed voor? In de praktijk gebruiken veel korttermijntraders de signaallijn voor het activeren de van de volgende *conditional orders* (zie hoofdstuk 3.1):

- 1) Koop het aandeel als de MACD-lijn de signaallijn doorkruist en als voor de richtingscoëfficiënt van de MACD-lijn geldt $\alpha > 0$;
- 2) Verkoop het aandeel als de MACD-lijn de signaallijn doorkruist en als voor de richtingscoëfficiënt van de MACD-lijn geldt $\alpha < 0$.

Volgens Lita Epstein, schrijver van meer dan 35 financiële boeken en Grayson Roze, trader en eigenaar van de website StockCharts.com, geeft de signaallijn ten opzichte van de MACD-lijn regelmatig een foutief signaal af. We gaan daarom niet verder in op bovenstaande punten. (Epstein & Roze, 2018)

2.2.3.3 Relatieve Sterkte Index (RSI)

De *Relatieve Sterkte Index*, in het Engels de *Relative Strength Index*, hierna afgekort met *RSI* is de laatste indicator die we bespreken. De RSI is ook een aparte grafiek onder de koersgrafiek en helpt met het bepalen van een koop- of verkoopmoment. Om te begrijpen hoe dat precies werkt, kijken we naar de berekening van de RSI.

De waarde van de RSI-indicator wordt als volgt genoteerd:

$$RSI(x, \alpha) \quad (2.15)$$

met:

x = de periode waarover de RSI wordt berekend in minuten, uren, dagen, maanden, enzovoorts (afhankelijk van de schaal van de grafiek). Meestal is deze periode gelijk aan 14;

α = welk moment in de desbetreffende periodes wordt bekeken, uitgedrukt in een letter, meestal H/L/O/C voor respectievelijk High, Low, Open, Close (hoogste koers, laagste koers, eerst bekende koers, slotkoers). Wij gebruiken in onze berekening de slotkoersen.

En dan nu de berekening voor $RSI(x, C)$:

$$RSI(x, C) = 100 - \frac{100}{(1 + \frac{\beta}{\gamma})} \quad (2.16)$$

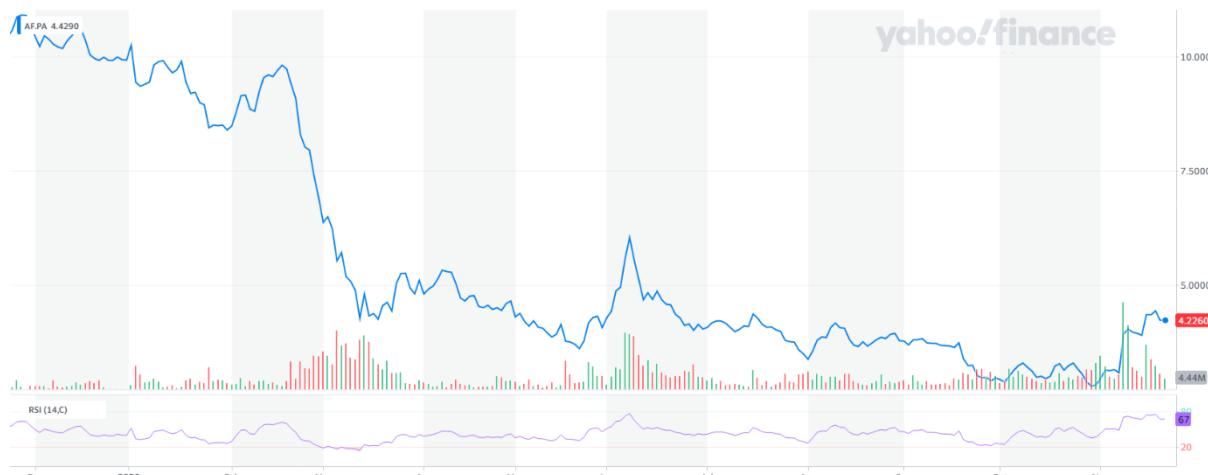
met:

$RSI(x, C)$ = de waarde van de RSI-indicator over x perioden, berekend met de slotkoersen;

β = de gemiddelde stijging. Als de slotkoers (C) hoger is dan de slotkoers van de dag ervoor, dan wordt ΔC opgeteld bij een *stijgingsvariabele* s . Deze stijgingsvariabele wordt gedeeld door de periode x om de waarde van de gemiddelde stijging β te verkrijgen;

γ = de gemiddelde daling. Als de slotkoers (C) lager is dan de slotkoers van de dag ervoor, dan wordt ΔC opgeteld bij een *dalingsvariabele* d . Deze dalingsvariabele wordt gedeeld door de periode x om de waarde van de gemiddelde daling γ te verkrijgen. (Fernando, 2020)

Dit alles resulteert in een lijn die oscilleert tussen de 0 en de 100. Een voorbeeld van de RSI-indicator in actie staat weergegeven in figuur 2.q.



Figuur 2.q: grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020 met daaronder een $RSI(14,C)$ -indicator. Bron gegevens: (Yahoo Finance, 2020).

Een trader kan vervolgens twee fictieve lijnen tekenen, $y = n$ en $y = m$ met $n < m$. Meestal worden voor n en m respectievelijk 30 en 70 gekozen, zoals de bedenker van de RSI in zijn boek *New Concepts in technical trading systems* suggereerde, maar Yahoo Finance kiest voor 20 en 80. (Wilder, 1978)

Als geldt $RSI > m$, dan is er veel vraag naar het desbetreffende aandeel. De koers is dan aan het stijgen. Op het moment dat geldt $RSI < m \wedge RSI' < 0$, dus als de RSI-indicator eerst boven m zat, en dan tot een waarde onder m zakt, is de koers aan het dalen. Dat is voor veel traders het moment om het aandeel te verkopen.

Andersom werkt dit ook. Als geldt $RSI < n$, dan is er weinig vraag naar het aandeel. De koers daalt. Op het moment dat geldt $RSI > n \wedge RSI' > 0$, doorkruist de RSI-indicator de lijn $y = n$. De koers begint dan aan een stijging. Veel traders zien dat als een geschikt moment om aandelen te kopen. (Wikipedia-bijdragers, 2020)

2.2.4 Relevantie?

Waarom is kennis van trends, patronen en indicatoren relevant, terwijl we een computerprogramma voor ons laten beleggen? Het antwoord op die vraag is eigenlijk heel simpel: controle. Later in dit verslag zal blijken dat wij, als mensen, bijzonder weinig inzicht hebben in wat ons programma precies doet. Trends, patronen en indicatoren kunnen ons helpen met het verkrijgen van beter inzicht in de markt én ze stellen ons in staat om zelf redelijk accurate voorspellingen van aandelenkoersen te doen. We kunnen dankzij de trends, patronen en indicatoren dus hypotheses opstellen over wat onze AI waarschijnlijk voor actie(s)¹⁹ zal suggereren. Als ons programma erg afwijkt van de door ons voorspelde trend, kan dat twee dingen betekenen:

- 1) Wij hebben een fout gemaakt in de programmacode, of;
- 2) Onze AI heeft de aandelenmarkt gekraakt en daarom veel meer kennis dan wij.

We hopen natuurlijk dat dat laatste het geval zal zijn.

2.3 Opties en futures

In hoofdstuk 1 hebben we twee verschillende handelingen beschreven die met aandelen uitgevoerd kunnen worden: aan- en verkopen. Het aankopen of verkopen van aandelen zijn niet in enige middelen die een trader in handen heeft om winst te maken. In deze paragraaf introduceren wij *opties* en *futures*.

¹⁹ Lees ook paragraaf 2.4 ‘verschillende posities’.

Een optie geeft een trader of belegger het **recht**, maar niet de plicht, om een aandeel te kopen of te verkopen tegen een vastgezette prijs. De trader hoeft dit aandeel niet in zijn bezit te hebben (bij verkoop), of daadwerkelijk aan te schaffen (bij koop). Het gaat bij een optie alleen maar om de theoretische mogelijkheid dat een aandeel verkocht of gekocht kan worden. (Investopedia, sd)

Er zijn twee soorten opties: *calls* en *puts*. Je koopt een call-optie als je een aandeel op een later moment tegen de prijs van nu wil kopen. Je koopt een put-optie als je een aandeel op een later moment tegen de prijs van nu wil verkopen.

Niet voor ieder aandeel zijn opties beschikbaar. Opties zijn net als aandelen verhandelbaar op beurzen en hebben eigen koersen. Hoe deze koersen tot stand komen, is voor ons niet zinvol om te bespreken. Wanneer een trader zal kiezen voor het kopen van een optie in plaats van een aandeel, bespreken we in paragraaf 2.4, verschillende posities.

In de tabel hieronder geven we een aantal verschillen tussen opties en aandelen aan.

Aandeel	Optie
Een aandeel is een zelfstandig financieel product	Een optie is een derivaat
Aandelen zijn onbeperkt houdbaar	Een optie kan een houdbaarheidsdatum hebben (Investopedia, sd)
Er is een beperkt aantal aandelen op de markt beschikbaar	In theorie kunnen er oneindig veel opties zijn (SpeedTrader, 2019)
Aandeelhouders hebben recht op een stem in vergaderingen, en hebben recht op dividenduitkering	Houders van opties hebben hier geen recht op

Tabel 2.r: een aantal verschillen tussen aandelen en opties.

Er bestaan ook *futures*. Een future(contract) lijkt veel op een optie. Het enige verschil is dat bij een future de koper de **plicht** heeft om een aandeel te kopen tegen een vooraf afgesproken prijs op een vooraf afgesproken datum. Er bestaan niet alleen *aandelenfutures*. Tegenwoordig zijn *grondstoffenfutures* verantwoordelijk voor een groot deel van de futuremarkt. Producten zoals maïs, goud en olie²⁰ worden alvast 'verkocht', voordat ze zijn geoogst, gedolven of aangeboord. (Wikipedia contributors, 2020)

²⁰ Dit zijn slechts drie voorbeelden. In werkelijkheid worden er futures uitgegeven voor bijna alle metalen, brandstoffen en agrarische producten. De enige uitzondering zijn Amerikaanse uien. De *Onion Futures Act* maakt het verboden om te handelen in uien-futures, omdat twee Amerikaanse handelaren in 1955 bijna de hele markt in handen kregen door alle futures op te kopen. (Lambert, 2010)

2.4 Verschillende posities

Als een aandeel een stijgende trend (bullish) laat zien, of patronen ons doen vermoeden dat de koers van het aandeel (in de nabije toekomst) zal stijgen, is het wellicht zinvol om in dat aandeel te investeren. We kunnen dan het aandeel zelf kopen, maar we kunnen ook geld besteden aan callopties. Als een handelaar denkt dat de koers gaat stijgen, en daar dan ook op anticipeert, zeggen we dat de desbetreffende handelaar *long* gaat.

Als een aandeel echter een dalende trend (bearish) laat zien, of als we vermoeden dat de koers gaat dalen, heeft het niet veel zin om aandelen aan te schaffen. Immers: die zullen (spoedig) minder waard worden. Hier bieden opties uitkomst. Met behulp van putopties kunnen we toch verdienen aan een bearish aandeel. Als een trader putopties koopt, en dus eigenlijk tegen een koersstijging wedt, gaat een trader *short*.

Long en short gaan worden *posities* genoemd. Nu we wat weten van trends, patronen en indicatoren en wat kennis hebben over opties, hebben we genoeg informatie om te gaan traden op de beurs.



Afbeelding 2.s: standbeeld van de kunstenaar Reinhard Dachlauer. Een bear en een bull staan voor de Frankfurt Stock Exchange. Afbeeldingsbron: (K., 2007).

3. Geautomatiseerde systemen in de aandelenhandel

Aandelen op de beurs worden al lang niet meer verhandeld door mannen in pak die in een grote ruimte door drie verschillende telefoons bevelen schreeuwen. Tegenwoordig worden de meeste transacties uitgevoerd door computers. Dat is niet heel gek. In hoofdstuk 1.3.1 spraken we over flitshandel en FlowTraders. Het is natuurlijk onmogelijk voor een mens om met flitshandelsnelheden aandelen te kopen en verkopen. In dit hoofdstuk gaan we in op de computersystemen die dit mogelijk maken.

Computersystemen die beleggers helpen met handelen, zijn in verschillende soorten en maten beschikbaar. In dit hoofdstuk onderscheiden we drie verschillende soorten programma's: *conditional orders*, *expert advisor* (EA) en *artificiële intelligentie* (AI).

3.1 Conditional orders

De simpelste programma's houden de koers van het aandeel in de gaten en maken aan de hand daarvan beslissingen. In *pseudocode*²¹ zou dat er op de volgende manier uit kunnen zien:

```
Als prijs < vooraf ingestelde waarde dan:  
    verkoop aandeel  
Als prijs > andere vooraf ingestelde waarde dan:  
    koop aandeel
```

De eerste twee programmaregels checken of de prijs van het desbetreffende aandeel lager is dan een vooraf ingestelde waarde. Dit wordt een *stop loss* genoemd: het programma stopt (=stop) het handelen in het aandeel en voorkomt daarmee dat er nog meer geld wordt verloren (=loss). (U.S. Securities and Exchange Commission, sd)

De andere programmaregels controleren of de prijs van het desbetreffende aandeel hoger is dan een vooraf ingestelde waarde. Lekker verwarring, maar ook dit wordt een stop loss genoemd. In dit geval verkoopt het programma het desbetreffende aandeel om zo de winst vast te zetten. (SEC.gov, 2011)

Deze programmaregels worden conditional orders genoemd, omdat aan een bepaalde voorwaarde moet worden voldaan, voordat door het programma een actie wordt uitgevoerd. Wikipedia beschrijft nog ten minste dertien andere soorten (Wikipedia contributors, 2020). Omdat ons eindproduct geen gebruikmaakt van dit soort programmaregels, gaan we hier niet verder op in.

²¹ Pseudocode is programmacode, maar dan geschreven in begrijpelijke taal. Deze codevorm is niet te interpreteren door computers.

3.2 Expert advisor (EA)

“Never change a winning team” is een citaat dat perfect bij de expert-advisorstrategie past. Een EA is een programma dat, aan de hand van een aantal vooraf ingestelde parameters en met behulp van ingewikkelde algoritmes, zo goed mogelijk het handelsgedrag van een expert na probeert te doen. Je zou een EA kunnen beschrijven als een groot aantal conditional orders die samen een beleggingsstrategie vormen. (ig.com, 2019)

Er bestaan verschillende soorten expert-advisorprogramma's. Makers van die programma's publiceren hun codes meestal niet. Immers, als iedereen dezelfde handelsstrategie toepast, is de kans aanwezig dat deze niet meer werkt. Ook de strategieën die een EA kan gebruiken, verschillen. Wikipedia erkent elf verschillende investeerstrategieën (Wikipedia contributors, 2020) en ongetwijfeld zijn er nog veel meer. Makers van een EA kunnen die op hun eigen manieren interpreteren en toepassen. Kortom, iedere expert-advisorcode ziet er anders uit.

Als de programmacodes niet bekend worden gemaakt, hoe komt een investeerder dan aan een werkend programma? Beleggers kunnen hun eigen EA's programmeren (Hallden, 2019), of kopen van beurshandelaren die erin gespecialiseerd zijn. De gemiddelde prijs van een basis expert-advisorprogramma ligt tussen de 50 en de 300 dollar (Fiverr, 2020). Wil je echter een programma van een respectabele effectenmakelaar aanschaffen, bijvoorbeeld de EA van de BJF Trading Group Corporation, een Canadees bedrijf gespecialiseerd in producten en services voor de forexmarkt, dan kost dit je al gauw meer dan 600 dollar (BJF Trading Group, sd).

Heb je dan de garantie dat je vet veel geld binnen gaat halen? Niet echt. Een EA kan weliswaar (veel) meer koersen, getallen en grafieken tegelijkertijd bijhouden dan een mens, maar de strategieën die een EA gebruikt, zijn door mensen bedacht. En mensen zijn nu eenmaal niet perfect.

Bovendien kunnen EA's ook zorgen voor veel chaos. Op 6 mei 2010 kelderde de Dow Jones index binnen enkele minuten met 1000 punten. Daarna herstelde de beurs zich en binnen 20 minuten werd het oude niveau weer bereikt. De U.S. Securities & Exchange Commission en de U.S. Commodity Futures Trading Commission deden vijf maanden lang onderzoek naar deze *flash crash*. Wat bleek: een Brit manipuleerde de beurs en door een fout in een computersysteem verkocht een groot handelshuis per abuis 4 miljard dollar aan aandelen (U.S. Commodity Futures Trading Commission & U.S. Securities & Exchange Commission, 2010). Dit resulteerde in een sneeuwbaleffect: flitshandelaren gingen tegen de lagere koers in grote volumes aandelen aankopen, om ze daarna meteen weer te verkopen. De Brit die de beurs op 6 mei manipuleerde, is vijf jaar later opgepakt. Men schat dat hij tussen 2010 en 2014 bijna 40 miljoen euro met beursmanipulatie heeft verdiend, iets wat mede mogelijk is gemaakt door automatische EA-systeem die direct reageren op bewegingen in de markt. (NOS, 2015)

Ondanks dat EA's nadelen kennen, worden ze toch veelvuldig gebruikt. Schattingen wijzen uit dat zo'n zestig procent van de beleggers een ingenieuze EA-systeem gebruikt om aandelen te

verhandelen (Kijk-redactie, 2012). Daar moet natuurlijk een goede reden voor zijn. We zetten de vier grootste voordelen van expert advisor op een rijtje:

- 1) **Tijdwinst.** Een EA heeft de mogelijkheid om meer dan honderd markten tegelijkertijd in de gaten houden. Een computer kan geprogrammeerd zijn om automatisch markten met hoge volatiliteit te zoeken, daar valt snelle winst te bepalen. Voor mensen is het praktisch onmogelijk om analyses van deze grootte uit te voeren.
- 2) **Toegankelijkheid.** Een expert advisor is doorgaans simpeler te verkrijgen en te gebruiken dan een AI. De gebruiker hoeft geen neuraal netwerk te ‘trainen’, maar kan gewoon een kant-en-klaar programma downloaden.
- 3) **Emotieloos.** De grootste vijand van een belegger, is de belegger zelf. Het is een bekend gezegde dat 90 procent van de handelaren verlies draait, en onderzoek wijst uit dat dat percentage misschien nog wel hoger ligt (Sesay, 2018). Een van de belangrijkste redenen dat dit percentage zo hoog ligt, is emotie. Veel beleggers raken gestrest als een aandeel onder de aankoopprijs zakt. Dit leidt tot aan- en verkopen gebaseerd op emotie, en niet op ratio. Een EA kent geen emotie en haalt deze factor dus volledig uit het beleggingsproces.
- 4) **Backtesting.** Diverse EA-programma’s bieden de mogelijkheid tot backtesting. Dat wil zeggen: nieuwe strategieën testen met behulp van historische data (Wikipedia-bijdragers, 2020). Het computersysteem ‘reist’ een bepaald aantal jaren terug in de tijd en start daar met het uitproberen van de strategie. Als een beleggingsstrategie in het verleden zinvol lijkt te zijn geweest, is de kans aanwezig dat de deze in de toekomst ook zal werken.

3.3 Artificiële intelligentie (AI)

Een programma dat gebruik maakt van artificiële intelligentie, ook wel *kunstmatige intelligentie* genoemd, kan volledig autonoom beslissingen maken en uitvoeren. Waar bij een EA programmaregels precies worden gevolgd, past een AI-programma zichzelf automatisch aan, om een zo optimaal mogelijk resultaat te bereiken. (Poole, Goebel, & Mackworth, 1998)

Als we praten over kunstmatige intelligentie, bedoelen we eigenlijk *machine learning*. Met machine learning wordt bedoeld: het gebruik van algoritmes en statistische modellen om een machine een bepaalde taak uit te laten voeren, zonder dat er van tevoren expliciete regels zijn vastgesteld (Blauw, 2019).

Even een voorbeeld. Stel we willen een computerprogramma schrijven dat foto's van olifanten kan herkennen.



Afbeelding 3.a: een olifant. Afbeeldingsbron: (Pixabay, 2020).

Een expert-advisoroplossing (in pseudocode) zou het volgende kunnen zijn:

```
Zoek op de foto naar een grijze vlek met vier poten en een staart.
```

```
Gevonden? dan:
```

```
    foto is olifant
```

```
Niet gevonden? dan:
```

```
    foto is geen olifant
```

Dit is echter niet heel efficiënt. Een grijze muis heeft dezelfde kenmerken, en zal dus ook voor een olifant aangezien worden. Daarnaast kan het natuurlijk ook voorkomen dat de staart of een van de poten van een échte olifant helemaal niet zichtbaar zijn op een foto. Het programma zal dan onterecht melden dat er geen olifant zichtbaar is.

Machine learning stelt ons in dit geval in staat om duizenden afbeeldingen van olifanten tegelijkertijd in een computerprogramma te stoppen. Vervolgens leert het systeem zichzelf wat een olifant precies is. Foto's worden niet langer aan één, twee of vijf criteria getoetst, maar soms wel aan honderden tegelijkertijd. De computer bepaalt zelf welke criteria dit precies zijn. In het geval van de olifant zal de AI waarschijnlijk kijken naar kenmerkende lichaamsvormen en kleuren. Dit leer- en toetsingsproces gebeurt met een *neuraal netwerk*. In hoofdstuk vier bespreken we de technische en wiskundige werking van zo'n netwerk. (3Blue1Brown, 2017)



Figuur 3.b: Machine learning stelt ons in dit geval in staat om duizenden afbeeldingen van olifanten tegelijkertijd in een computerprogramma te stoppen.

3.4 Waarom AI op de beurs niet vaak gebruikt wordt, maar wij het wel gaan proberen

Hoewel kunstmatige intelligentie al sinds 1956 (Russell, Norvig, & Canny, 2003) bestaat en een langlopend onderzoek met AI op de beurs tussen 1992 en 2015 een rendement van 681 procent boekte (European Journal of Operational Research, Krauss, Do, & Huck, 2017), lijkt artificiële intelligentie niet echt door te breken in de aandelenwereld (Barlow, 2019). Waarom gebeurt dat eigenlijk niet? Nu we iets meer weten over AI, kunnen we een aantal beargumenteerde hypotheses opstellen. We zetten de belangrijkste hieronder op een rij.

- 1) **De mens verliest veel controle.** Wij, mensen, zijn *control freaks*. Het liefst willen we zoveel mogelijk processen in onze eigen hand houden. Al helemaal de processen waarbij onze zuurverdiende spaarcenten een risico zouden kunnen lopen. Een programma gebaseerd op kunstmatige intelligentie biedt weinig mogelijkheden voor menselijk ingrijpen: de computer leert, de computer beslist.
- 2) **AI vereist kracht en is niet extreem snel.** Het kost relatief veel computerkracht om een programma gebaseerd op kunstmatige intelligentie te draaien. Omdat een AI computerintensief is, zijn programma's niet zo snel als men dat in de aandelenwereld het liefst zou zien. Zeker bij flitshandel worden posities geopend en gesloten binnen een fractie van een seconde. De AI heeft dan nog niet voldoende tijd gehad om een goede berekening te maken.
- 3) **AI is misschien helemaal niet nodig.** Eigenlijk vervangt AI een menselijk brein. Op dit moment zijn beleggers en traders prima in staat zelfstandig beslissingen te nemen. Waar is dan een AI voor nodig?
- 4) **AI vereist veel invoer om accuraat te worden.** In het geval van het herkennen van een olifant zijn dat duizenden afbeeldingen van olifanten, en in het geval van het voorspellen

van de beurs zijn dat jaren aan historische koersdata²². Die data is niet altijd goed beschikbaar, volledig of betrouwbaar. Bovendien kunnen we vraagtekens zetten bij de representativiteit van historische koersdata: als een bepaalde gebeurtenis in het verleden tot een bepaalde uitkomst heeft geleid, hebben we dan de garantie dat een vergelijkbare gebeurtenis in de toekomst dezelfde uitkomst oplevert?

Ondanks dit alles is er hoop voor kunstmatige intelligentie op de beurs. Er zijn enkele interessante initiatieven die we graag in ons werkstuk willen bespreken.

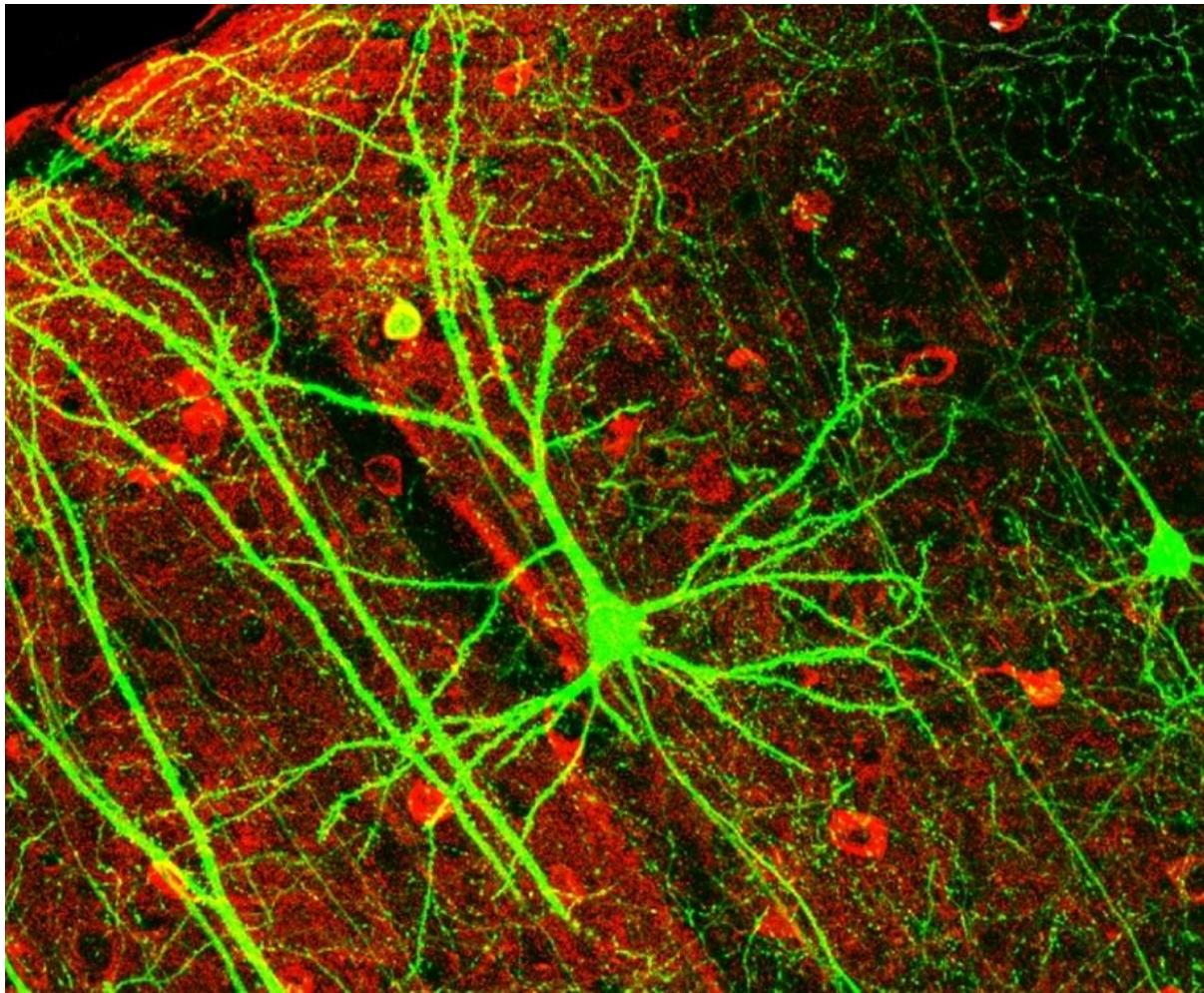
- 1) **AI-beleggingsfondsen.** In de inleiding kwamen we ze al tegen: beleggingsfondsen die draaien op kunstmatige intelligentie. Er zijn niet veel van dit soort hedge funds, en de meeste vallen bovendien onder de categorie *computer-assisted* (Outside Insight, 2018). Dat wil zeggen dat een AI weliswaar beslissingen maakt, maar ze niet volledig autonoom kan uitvoeren. IBM experimenteert met volledig autonome beleggingsfondsen en het AI-fonds NextAlpha claimt al een rendement van 40% per jaar (Kahn J., 2020).
- 2) **Tekstherkennung.** Er wordt onderzoek gedaan naar de inzet van tekstherkennings-AI-robots voor het voorspellen van beurskoersen. Zo promoveerde Frederik Hogenboom in 2014 aan de Erasmus Universiteit met zijn proefschrift getiteld *Automated Detection of Financial Events in News Text*. Hij deed onderzoek naar een algoritme dat met behulp van AI financiële nieuwsberichten analyseerde en aan de hand van die nieuwsberichten een koersvoorspelling deed. Dit heeft natuurlijk vooral te maken met fundamentele analyse (lees ook hoofdstuk 2.1), en wat minder met de technische analyse waar wij op focussen. (Blaauwoer, 2015)

Er zijn dus ontwikkelingen op het gebied van computersystemen in de aandelenhandel die ons ervan overtuigen dat het bouwen van een belegrobot mogelijk en rendabel is. Bovendien wordt AI ieder jaar sterker en volgen verschillende generaties computers elkaar in rap tempo op. Kortom: als een automatische belegger op basis van kunstmatige intelligentie met de technologie van vandaag nog niet mogelijk is, is het goed mogelijk dat dit met de techniek van morgen wel kan. Wij gaan in elk geval ons best doen.

²² Historische koersdata: (aandelen)koersen uit het verleden.

4. Hoe leert een neuraal netwerk?

Als we leren voor een toets wiskunde, Frans, of maatschappijleer zijn onze hersenen bezig met een fascinerend biologisch proces. Ons brein slaat informatie op doordat de hersencellen, *neuronen*, elektrische impulsen naar elkaar sturen. Na elk stroompje wordt de verbinding tussen de hersencellen sterker. Op een gegeven moment is de verbinding sterk genoeg. De leerstof is dan onderdeel geworden van het brein. (Lauro & Jesan, 2003)



Afbeelding 4.a: microscopisch beeld van enkele neuronen in ons brein. De groene bolletjes zijn hersencellen en de groene lijnen zijn de verbindingen tussen deze cellen. Afbeeldingsbron: (wikiwand.com).

4.1 Trainen en trainingsdata

De neuronen van een kind sturen al lang voor de geboorte elektrische impulsen naar elkaar. Hierdoor leert het kind bijvoorbeeld de stem van de moeder herkennen. Volautomatisch (McElroy, 2013). Bij een computergestuurd neuraal netwerk is het helaas niet zo'n feest. Een neuraal netwerk is geboren als een onbeschreven blad, een *tabula rasa*, en moet door de programmeur *getraind* worden. (Jong, 2019)

Laten we teruggaan naar de olifant uit hoofdstuk 3. Stel dat we daadwerkelijk een programma willen schrijven dat met behulp van kunstmatige intelligentie een olifant kan herkennen. In dat geval hebben we veel olifantenfoto's nodig. Die olifantenfoto's kunnen we, één voor één, aan ons programma 'voeren'. Dit 'voeren' wordt ook wel 'trainen' genoemd.

In de rest van dit hoofdstuk bespreken we de weg die één olifantenfoto aflegt. In werkelijkheid wordt het procedé dat in paragraaf 4.2.1 tot en met paragraaf 4.2.8 beschreven staat, duizenden keren herhaald, omdat alle olifantenfoto's individueel door het neurale netwerk worden geleid. Pas na duizenden foto's is het neurale netwerk accuraat genoeg om een voorspelling te doen over of er op een foto een olifant staat afgebeeld. De duizenden olifantenfoto's die we in het neurale netwerk stoppen voordat het een correcte voorspelling kan doen, worden *trainingsdata* genoemd.

Voor alle trainingsdata moet gelden:

$$P(\text{afgebeeld object} = \text{olifant}) = 100\% \quad (4.1)$$

Dit wil zeggen dat op alle trainingsfoto's een olifant moet staan. Anders kan het neurale netwerk nooit leren wat wel en geen olifant is.

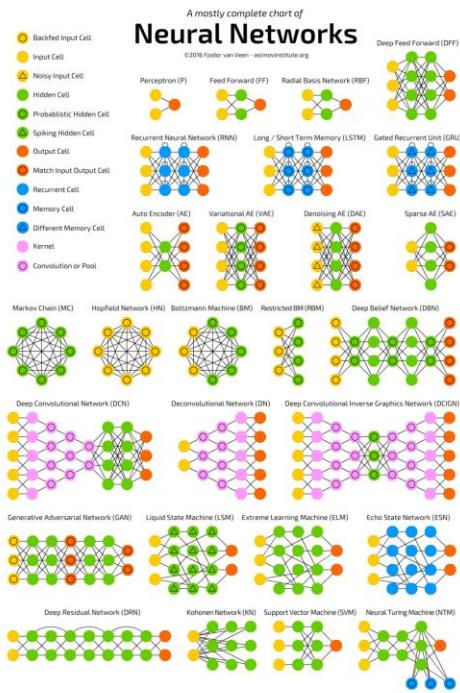
Na het invoeren van de trainingsdata kan het neurale netwerk, als het goed is, een olifant herkennen. We kunnen dit controleren door *testdata* in te voeren. De *testdataset* bestaat uit foto's van zowel olifanten als andere objecten. Het percentage correct voorgespelde foto's vertelt ons iets over de accuraatheid *AP* van het netwerk. (Hui, 2020)

$$AP = \frac{\text{aantal correct voorgespelde foto's}}{\text{totale grootte testdataset}} * 100\% \quad (4.2)$$

Hoe groter de waarde van *AP*, hoe beter het neurale netwerk een olifant van een niet-olifant kan onderscheiden.

4.2 Opbouw van een computergestuurd neuraal netwerk

Hoe ziet zo'n neuraal netwerk er dan precies uit? Nou, een *computergestuurd neuraal netwerk* functioneert eigenlijk precies hetzelfde als menselijke hersenen.



In de figuur hiernaast staan 27 neurale netwerken afgebeeld. Dit is geen complete lijst, maar wel een mooie verzameling van netwerkstructuren die regelmatig worden gebruikt. De figuur staat vergroot afgebeeld in appendix VII.

We snappen heel goed dat dit er overweldigend uitziet. En om eerlijk te zijn: wij begrijpen ook echt niet alles wat op deze afbeelding staat. Voor nu is het belangrijk dat je inziet dat een computergestuurd neurale netwerk overeenkomsten vertoond met de neuronen in onze hersenen. Zowel het computernetwerk als de menselijke variant bestaan uit 'bolletjes' en 'bruggetjes' tussen de bolletjes.

Dan is het nu tijd om een neurale netwerk stap voor stap te ontleden.

Figuur 4.b: een overzicht van 27 neurale netwerken.

Een vergrote versie is te vinden in appendix VII.

Afbeeldingsbron: (towardsdatascience.com).

4.2.1 Input: dit gooien we erin

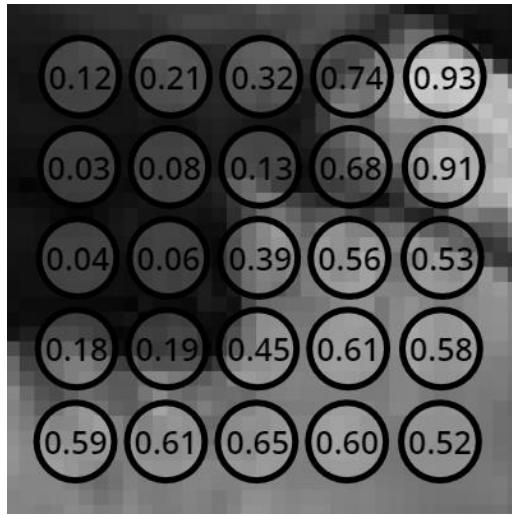
Ieder trainingsproces begin bij de *input*. In de trainingsfase is dat bij ons een plaatje van een olifant. Laten we de foto eerst versimpelen, zodat de computer minder data te verwerken heeft. We verwijderen alle *metadata*²³ en zetten de foto om naar zwart-wit²⁴. Het resultaat hiervan is zichtbaar in figuur 4.c.1.

²³ Metadata: alle niet-zichtbare informatie, zoals locatie-, camera- en datumgegevens.

²⁴ Door dit te doen zijn de *RGB-waarden* (hoeveel rood, groen en blauw pigment aanwezig is) van iedere *pixel* (een klein onderdeel van een foto) niet meer relevant. Dit zorgt voor een aanzienlijke vermindering van de hoeveelheid data.



Figuur 4.c.1 (links): de olifant van in zwart-wit.



Figuur 4.c.2 (rechts): een ingezoomd gedeelte van de zwart-witte olifant in het linker figuur.

Figuur 4.c.2 is een klein deel van de olifantenfoto. Merk op dat iedere pixel een bepaalde waarde tussen 0,00 en 1,00 heeft. Deze waarde is afhankelijk van hoe donker een bepaalde pixel is.

In ons geval geldt:

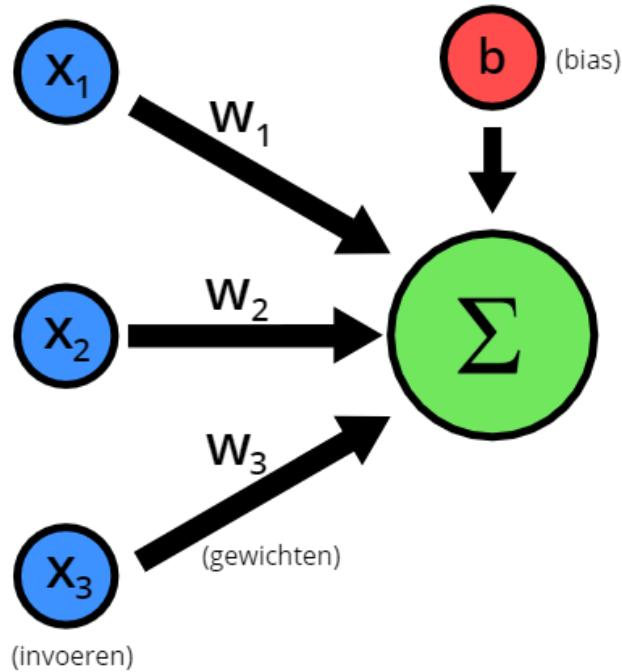
$$100\% \text{ zwart} \text{ en } 0\% \text{ wit} \rightarrow 0,00 \quad (4.3.1)$$

$$100\% \text{ wit} \text{ en } 0\% \text{ zwart} \rightarrow 1,00 \quad (4.3.2)$$

De kans dat een pixel helemaal zwart of helemaal wit is, is natuurlijk klein. In de praktijk zijn pixels vaak grijs. De waarde zit dan ergens tussen 0,00 en 1,00. Het is essentieel om een foto naar *pixelwaardes* om te zetten, omdat een neuraal netwerk pixelwaardes kan ‘begrijpen’ en een gewone foto niet.

4.2.2 Neuron: de kleinste eenheid van het netwerk

Een kunstmatig neuron is simpelweg een object dat een nummer vast kan houden (3Blue1Brown, 2017). Dit getal is bijna geheel afhankelijk van de invoeren (X_1 tot en met X_n) die het neuron krijgt. Meestal komen deze invoeren van andere neuronen. Elke invoerstroom heeft zijn eigen *gewicht* (W_1 tot en met W_n): een factor die bepaald hoe zwaar de invoerstroom meetelt. Tot slot heeft elk neuron ook een *bias*. Dat is een extra getal dat bij de waarde van het neuron wordt opgeteld. (Malik, 2019)



Figuur 4.d: een versimpeld schematisch overzicht van een neuron.

In figuur 4.d zien we een voorbeeld van verschillende invoeren, met elk hun eigen gewicht.

De waarde van een neuron is dus direct afhankelijk van de invoeren en gewichten. Deze waarde wordt berekend door de invoeren die hij binnenkrijgt te vermenigvuldigen met hun bijbehorende gewichten. (Skalski, 2020)

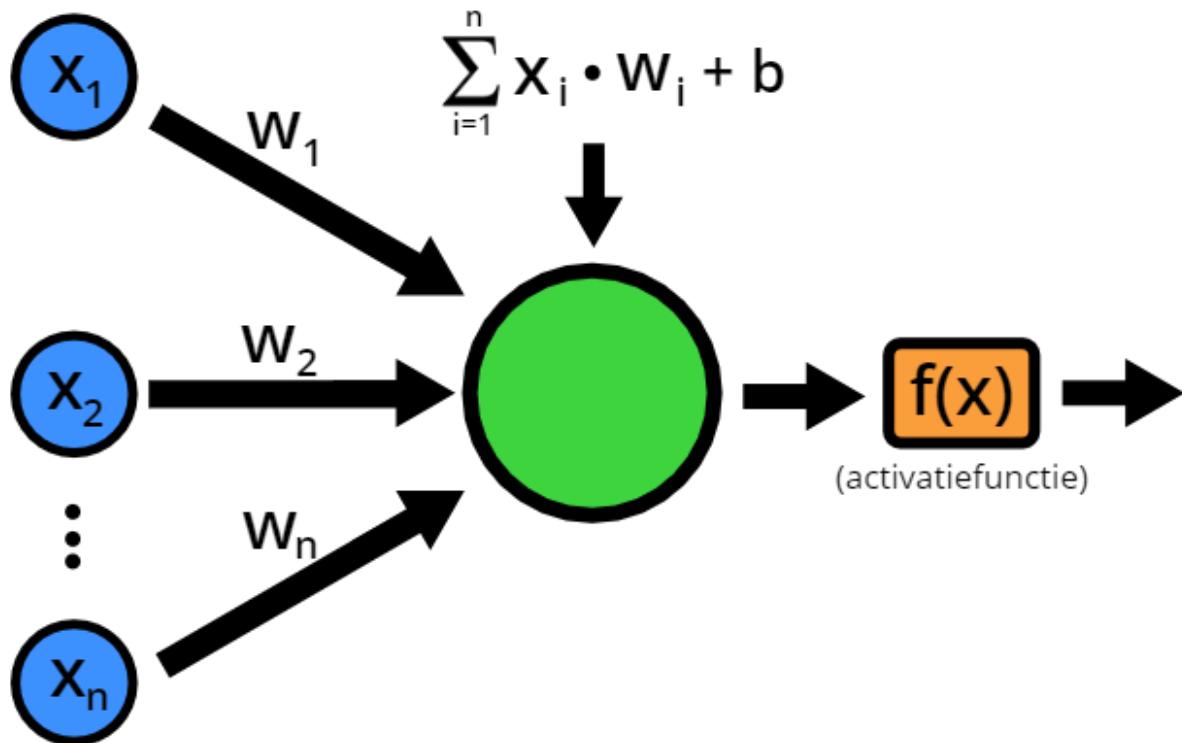
$$\text{waarde neuron} = w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_n \cdot x_n \quad (4.4)$$

Dit is vrij veel schrijfwerk. Daarom korten we het af op deze manier:

$$w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_n \cdot x_n = \sum_{i=1}^n w_i \cdot x_i \quad (4.5)$$

Na deze berekening wordt de bias van het neuron ook nog toegevoegd. Het resultaat van dit alles wordt opgeslagen in het neuron zelf. Tenslotte geeft het neuron zijn waarde door aan een *activatiefunctie* $f(x)$, maar daarover later in meer in paragraaf 4.2.3.

In onderstaand figuur hebben we alle berekeningen rondom een neuron gevisualiseerd.



Figuur 4.e: een schematisch overzicht van alle waarden en berekeningen rond een enkel neuron.

4.2.2.1 Functies van het gewicht en de bias

Het doel van het neuron is de waarde van de activatiefunctie $f(x)$ zo hoog mogelijk²⁵ te krijgen²⁶. Hoe hoger de waarde van $f(x)$, hoe dichter het neuron bij de ‘juiste’ uitkomst is. Om een hoge uitkomst van de activatiefunctie te krijgen, moet het neuron op zoek naar een specifieke combinatie van de gewichten W_1 tot en met W_n . Het neuron maakt dus zelf beslissingen over de waarden van W_1 tot en met W_n .

De bias is een extra variabele die het neuron kan gebruiken om dichterbij een hoge $f(x)$ -waarde te komen. Dit leggen we hieronder uit met een voorbeeld.

²⁵ Hoe ‘hoog’ dit precies is, hangt af van het type activatiefunctie. Dit leggen we in paragraaf 4.2.3 uit.

²⁶ Of zo laag mogelijk. Als een neuron getraind is op een bepaalde input x , en hij x als input krijgt, dan moet de waarde van de activatiefunctie zo hoog mogelijk zijn. Als een neuron echter is getraind op een inputwaarde y , en hij x als input krijgt, dan moet de waarde van het neuron zo laag mogelijk zijn. In de rest van deze paragraaf nemen we aan dat het neuron getraind is op de input die hij heeft ontvangen, dus dat de activatiefunctie zo hoog mogelijk moet zijn.

Stel dat geldt:

$$\sum_{i=1}^n (w_i \cdot x_i) = 0,99 \quad (4.6)$$

en de gewenste waarde is:

$$f(x) = 1,00 \quad (4.7)$$

Dan zal het neuron kiezen voor een bias $b = 0,01$ om aan de gewenste waarde van 1,00 te voldoen.

Het is overigens geen slecht ding als de waarde van de activatiefunctie lager is dan de gewenste waarde. Dit hoeft namelijk niet te betekenen dat het neuron slecht is ingesteld. Het betekent alleen dat dit invoer die het neuron binnenkrijgt, niet overeenkomt met waar het neuron op is getraind. Dit soort situaties komen bijvoorbeeld voor als we het neurale netwerk een testdataset²⁷ als input geven.

4.2.3 Activatiefunctie: de invoer verandert in een uitvoer

De activatiefunctie zijn we al een paar keer eerder tegengekomen, maar waar dient zo'n formule eigenlijk voor? Simpelweg bepaalt deze functie wat het neuron gaat doorgeven aan de neuronen in de volgende laag (Sharma, 2019). Allereerst is het goed om te weten dat er verschillende soorten activatiefuncties bestaan, elk met voor- en nadelen. Om deze paragraaf beknopt en duidelijk te houden, bespreken we de details van enkele belangrijke activatiefuncties in appendix VIII.

Laten we de werking van activatiefuncties uitleggen met een voorbeeld. In dit voorbeeld gebruiken we de *sigmoidfunctie* (appendix VIII.3). Deze functie wordt als volgt gedefinieerd:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.8)$$

En voor deze functie geldt:

$$x = \text{waarde neuron} = \sum_{i=1}^n (w_i \cdot x_i) + b \quad (4.9)$$

waarbij:

w_i = het gewicht dat bij een bepaalde invoer hoort

x_i = de waarde van de invoer

b = de waarde van de bias

²⁷ Zoals eerder besproken in paragraaf 4.1: een testdataset bestaat uit afbeeldingen van zowel olifanten als niet-olifanten.

Tijd om een paar getalletjes in te vullen. Stel dat $\sum_{i=1}^n w_i \cdot x_i = 0,99$ en $b = 0,01$.

Daaruit volgt dat $x = 1$.

Dit substitueren in $f(x)$ levert het volgende op:

$$f(1) = \frac{1}{1 + e^{-1}} \approx 0,7311 \quad (4.10)$$

Hieruit volgt dat de waarde 0,7311 wordt doorgegeven aan het volgende neuron.

4.2.3.1 Het belang van activatiefuncties

Waarom is een activatiefunctie eigenlijk nodig? Waarom nemen we niet gewoon direct de waarde van $\sum_{i=1}^n w_i \cdot x_i + b$ om door te geven aan het volgende neuron? Daar zijn twee redenen voor (Jain, 2020):

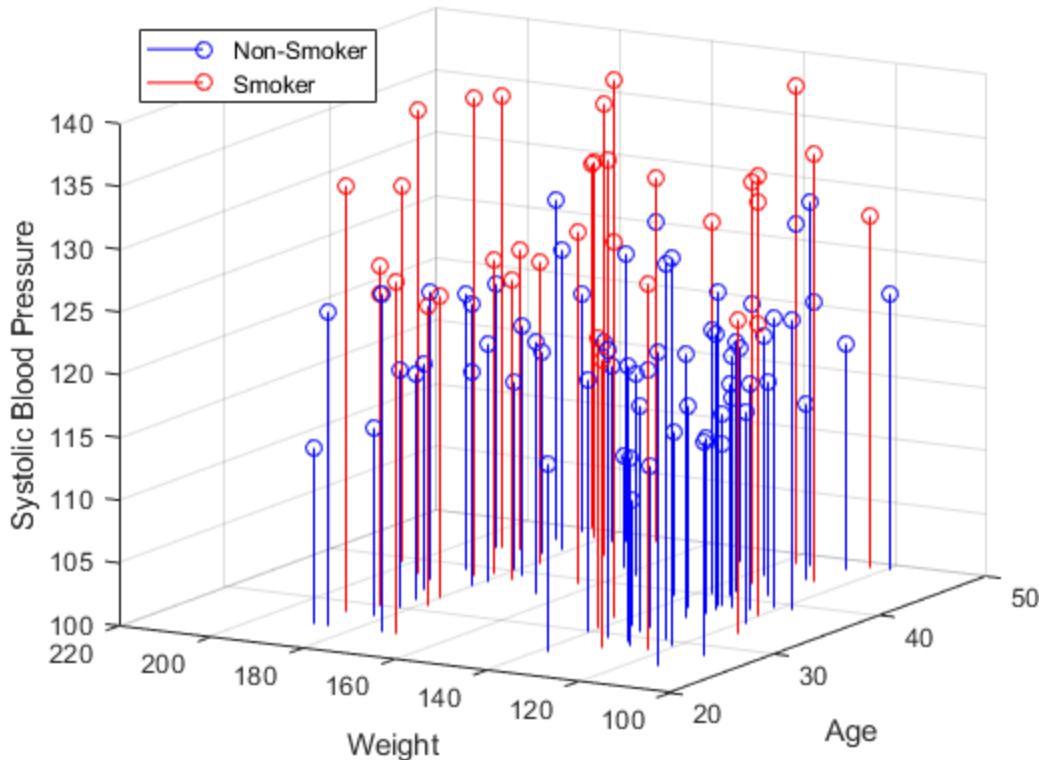
- 1) De activatiefunctie voorkomt gigantische getallen. In het geval dat een neuraal netwerk miljoenen parameters heeft, is de kans op een grote waarde voor $\sum_{i=1}^n w_i \cdot x_i + b$ zeer reëel. Een (te) grote waarde van deze sommatie proberen we te allen tijde te vermijden, omdat neurale netwerken (en de computers waarop ze worden gedraaid) hiermee veel moeite hebben. Een functie met een of meerdere horizontale asymptoten is daarom ideaal als activatiefunctie.

Voorbeeld: voor $f(x) = \frac{1}{1+e^{-x}}$ gelden de volgende limieten:

$$\lim_{x \rightarrow -\infty} f(x) = 0 \text{ en } \lim_{x \rightarrow \infty} f(x) = 1$$

Dus met twee horizontale asymptoten $y = 0$ en $y = 1$ is deze functie zeer geschikt, omdat er geen gigantische waarden uit voortkomen.

- 2) Belangrijker is nog dat activatiefuncties niet-lineaire berekeningen forceren. Dat leggen we uit aan de hand van een grafiek over de leeftijd, het gewicht en de bloeddruk van rokers en niet-rokers.



Grafiek 4.f: een driedimensionale puntenwolk met waardes voor het gewicht, de leeftijd en bloeddruk van rokers en niet-rokers. Afbeeldingsbron: (Jain, 2020).

Een trendlijn²⁸ die rekening houdt met zowel *Weight*, als *Age* en *Systolic Blood Pressure* is een rechte lijn door de driedimensionale ruimte die de punten in de grafiek weliswaar benadert, maar die nooit het exacte verband tussen de drie variabelen kan blootleggen. Dit komt omdat de correlatie tussen gewicht, leeftijd en bloeddruk in dit geval niet lineair is. Het tekenen van een lineaire trendlijn heeft dan niet zoveel zin.

In een neuraal netwerk is hetzelfde waar. Veel dingen die men met kunstmatige intelligentie probeert te berekenen, zijn niet te vangen in een lineair verband. In de formule $\sum_{i=1}^n w_i \cdot x_i + b$ kunnen we $w_i \cdot x_i + b$ zien als een rechte lijn in de vorm $y = ax + b$. De som van een n aantal lineaire functies levert nog steeds een rechte lijn op.

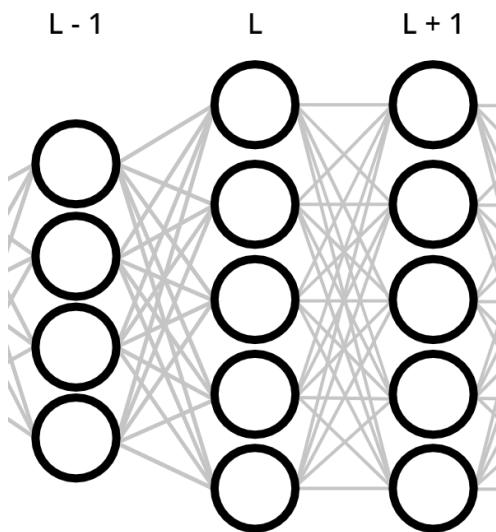
De activatiefuncties zetten de lineaire verbanden om naar complexere functies. Dit maakt het herkennen van complexere patronen door een neuraal netwerk mogelijk.

²⁸ In de wereld van neurale netwerken wordt een trendlijn een *lineair classifier* genoemd. Een lineair classifier is een lijn die probeert te voorspellen welke waarde van y bij een bepaalde waarde x hoort. Eigenlijk analyseert een lineair classifier aanwezige trends en probeert deze samen te vatten in één functie, kortom: een trendlijn. Er zitten kleine verschillen tussen een trendlijn en een lineair classifier, maar de achterliggende principes zijn hetzelfde.

4.2.4 Enkele laag: samen staan neuronen sterk

We hebben het tot nu toe nog alleen gehad over enkele neuronen. Het wordt interessant als we, net zoals in een menselijk brein, meerdere neuronen laten samenwerken. Als neuronen onafhankelijk, maar wel precies tegelijkertijd, berekeningen uitvoeren, spreken we van een *laag*. Met onafhankelijk bedoelen we dat de neuronen elkaar niet beïnvloeden. Zo hoeft niet ieder neuron hetzelfde gewicht en dezelfde bias te hebben.

De invoer die één neuron in een laag ontvangt, is meestal een combinatie van de uitvoeren van alle neuronen in de vorige laag²⁹. In afbeelding 4.g is dit schematisch weergegeven. De grijze lijnen staan voor verbindingen tussen de neuronen. Via deze verbindingen worden waarden doorgegeven.



Figuur 4.g: een voorbeeld van neuronen die in meerdere lagen zijn opgesteld.

We introduceren hier meteen een nieuwe notatie voor lagen. De laag waar we over praten, noemen we L . De laag ervoor en erna zijn respectievelijk $L - 1$ en $L + 1$. (3Blue1Brown, 2017)

Overigens hoort de letter L officieel een kleine letter te zijn, maar in de wiskunde achter neurale netwerken wordt de letter i ook veelvuldig gebruikt. Om onduidelijkheid te voorkomen gebruiken we de hoofdlettervariant.

Aan een laag kunnen we enkele interessante berekeningen koppelen. Bijvoorbeeld formule 4.11. Met deze formule berekenen wij de waarde van een bepaald neuron i in laag L . (Nielsen, 2019)

²⁹ Er bestaan neurale netwerken waarvoor dit niet geldt. In die netwerken zijn niet alle neuronen met elkaar verbonden. Deze netwerken komen zelden voor en hebben zeer specifieke functies. Hier gaan we in ons profielwerkstuk niet verder op in.

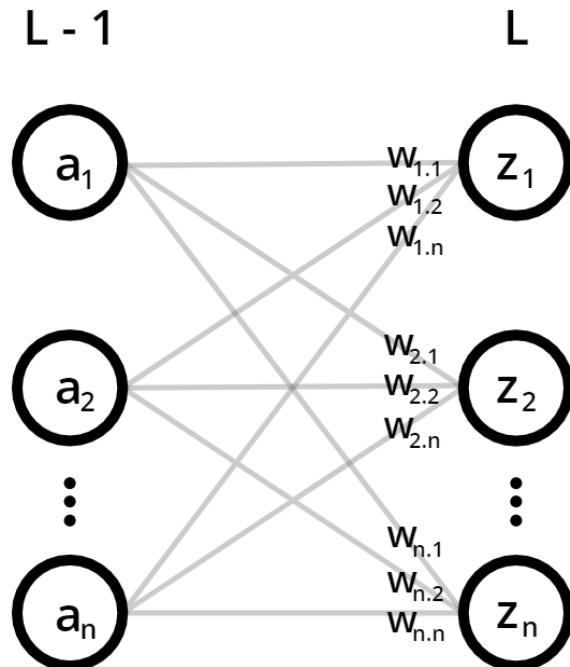
$$z_i^L = \sum_{k=1}^n (w_k^L \cdot a_k^{(L-1)}) + b_i^L \quad (4.11)$$

Laten we deze formule even stap voor stap bekijken. (Skalski, 2020)

De b_i^L staat voor de bias. De L erboven geeft aan bij welke laag de bias hoort, en de i zegt iets over het neuron waaraan de bias is gebonden.

Nu het grootste deel van de formule: de sommatie $\sum_{k=1}^n (w_k^L \cdot a_k^{(L-1)})$. Deze sommatie werkt hetzelfde als bij een enkel neuron. De w staat voor het gewicht, en de a staat voor de uitkomst van de activatiefunctie. De L geeft aan over welke laag we het hebben. Merk op dat we in deze sommatie geen i meer gebruiken, maar een k . Dit komt omdat we de i al eerder als iets anders hebben gedefinieerd. Tot slot gebruiken we $L - 1$ en niet L in de macht van de uitkomst van de activatiefunctie. Dit is eigenlijk heel logisch, omdat we de uitkomst van de activatiefuncties van de vorige laag willen hebben. a_k^L is immers de waarde van de activatiefunctie van deze laag, en die hebben we nog niet berekend (en kunnen we dus ook nooit invullen).

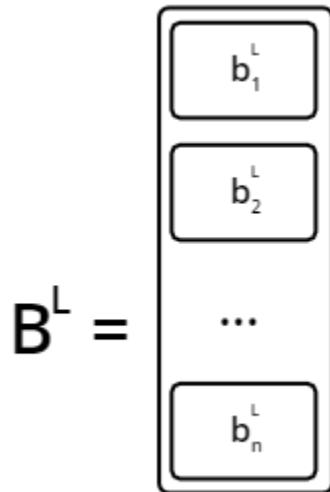
Uit formule 4.11 blijkt dat er voor elk neuron dezelfde berekeningen worden gedaan. Het zou voor een computer erg inefficiënt zijn om dezelfde berekeningen miljoenen keren achter elkaar te doen. Daarom worden deze berekeningen vaak compact opgeschreven met behulp van vectoren en matrices. Om te begrijpen hoe dit precies werkt, kijken we naar een versimpeld neuraal netwerk (figuur 4.h).



Figuur 4.h: een schematisch neuraal netwerk van 2 lagen, met aangegeven gewichten per stroom.

Figuur 4.h is een netwerk met twee lagen en n neuronen. Iedere verbinding heeft een bepaald gewicht $W_{n,x}$. De bias b staat niet in dit plaatje, maar daar moeten we natuurlijk wel rekening mee houden.

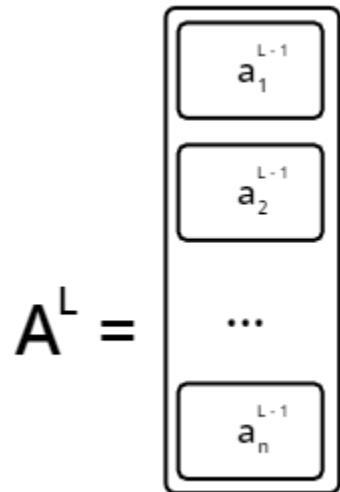
Nu we het toch over de bias hebben, laten we die meteen in een n -dimensionale vector zetten. Zo'n vector zou er als volgt uitzien: (Ross, 2018)



Figuur 4.i: biases voor een bepaalde laag in een vector.

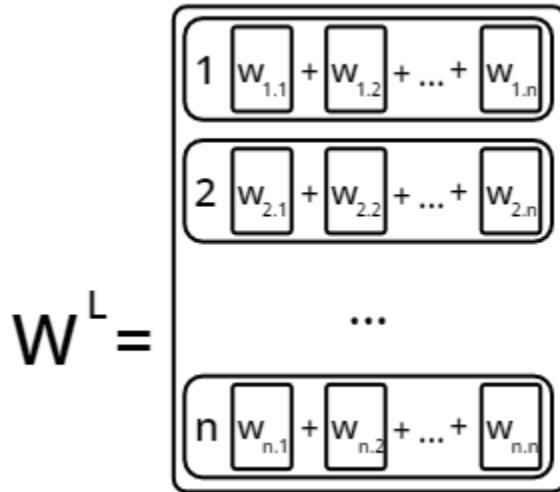
In deze figuur staat B^L voor de vector van biases en staan b_n^L voor een individuele bias van een bepaald neuron. Deze vector geldt voor de berekeningen van één laag.

Hetzelfde kunnen we doen voor de uitkomst van de activatiefuncties van de $L - 1$. Dat ziet er misschien gecompliceerder uit, maar werkt eigenlijk op dezelfde manier.



Figuur 4.j: activatiefunctieresultaten voor een bepaalde laag in een vector.

En tot slot de gewichten. Deze gewichten drukken we niet uit in een vector, maar in een matrix, omdat we hier meer dan één waarde hebben per rij. Zie figuur 4.k hieronder. In deze figuur zijn in elke rij de gewichten die een bepaald neuron verwerkt, weergegeven. Het nummer n voor de rij geeft aan over welk neuron we praten. $W_{1,n}$ zijn de gewichten die bij de inputs van het desbetreffende neuron horen.



Figuur 4.k: gewichten gestapeld in een matrix.

Eerder in deze paragraaf hadden we al aangetoond dat voor de waarde van een neuron i in laag L de volgende formule geldt:

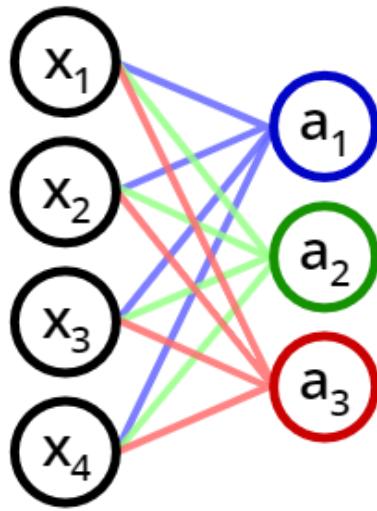
$$z_i^L = \sum_{k=1}^n (w_k^L \cdot a_k^{(L-1)}) + b_i^L \quad (4.12)$$

Nu we $\sum_{k=1}^n w_k^L$ hebben geschreven als matrix W^L , $\sum_{k=1}^n a_k^{(L-1)}$ hebben gedefinieerd als vector A^L en b_i^L hebben geschreven als vector B^L , kunnen we formule 4.12 versimpelen. Substitutie van W^L , A^L en B^L in formule 4.12 levert het volgende op:

$$z^L = W^L \cdot A^L + B^L \quad (4.13)$$

Merk op dat hier z_i^L is veranderd in z^L . Dit is omdat we dankzij deze vectoren en matrices ons niet meer bezig hoeven te houden met ieder individueel neuron, maar alle neuronen in een laag tegelijkertijd kunnen bekijken.

We geven tot slot nog een voorbeeld om de besproken informatie in deze paragraaf te verduidelijken en samen te vatten. In dit voorbeeld geldt dat de uitkomsten van de activatiefuncties van de vorige laag worden geschreven als X_n .



Figuur 4.l: een neurale netwerk. In dit netwerk staan x_n voor de uitkomsten van de activatiefuncties van laag $L - 1$ en a_n staan voor de waarden die het neuron vasthoudt.

We werken met kleuren om de berekeningen hieronder duidelijker te maken. Als we dit netwerk invullen in de formule $z^L = W^L \cdot A^L + B^L$ levert dat het volgende op: (Jordan, 2018)

$$\begin{bmatrix} W_1 + W_2 + W_3 + W_4 \\ W_1 + W_2 + W_3 + W_4 \\ W_1 + W_2 + W_3 + W_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix}$$

Figuur 4.m: een formule die het neurale netwerk in figuur 4.l weergeeft.

En dit kunnen we herschrijven tot onderstaande matrix.

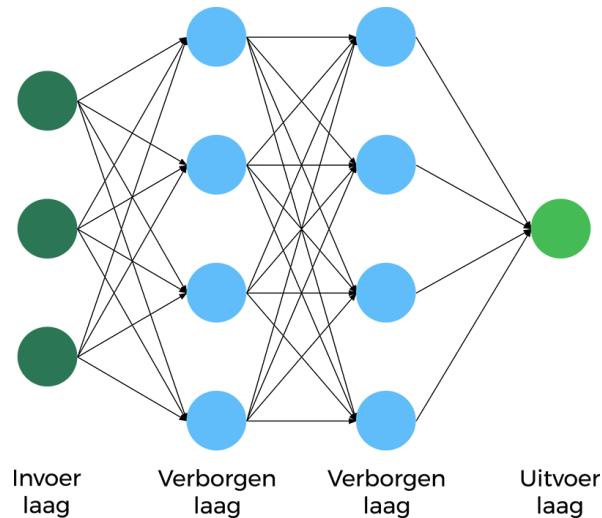
$$\begin{bmatrix} W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4 + b \\ W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4 + b \\ W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4 + b \end{bmatrix}$$

Figuur 4.n: de berekening die uiteindelijk wordt uitgevoerd.

Deze matrix is door een computer gemakkelijk te interpreteren.

4.2.5 Tussenlagen: dit is waar de magie gebeurt

De lagen die tussen de invoer- en de uitvoerlaag zitten, worden *tussenlagen* of *verborgen lagen* genoemd. Dit is waar het ‘leren’ van het netwerk gebeurt; hier worden ‘voorspellingen’ gecreëerd. Deze lagen worden ‘verborgen’ genoemd, omdat het erg moeilijk is om te achterhalen wat het neurale netwerk hier precies doet.



Figuur 4.0: de tussenlagen, ook wel verborgen lagen genoemd, worden in dit figuur gevisualiseerd door de lichtblauwe bolletjes. In de meeste neurale netwerken bevindt het grootste deel van de neuronen zich in zo’n verborgen laag. Afbeeldingsbron: (Hessen).

Hoeveel tussenlagen heeft een netwerk dan nodig? Dat hangt van de accuraatheid af. De hoeveelheid tussenlagen heeft direct te maken met hoe complex we een netwerk willen maken. Een netwerk met maar één tussenlaag is efficiënt en snel, maar niet accuraat. Het is gebruikelijk om twee of drie tussenlagen te hebben, maar dit kunnen er prima meer zijn. Bijvoorbeeld bij veel verschillende invoeren, of netwerken die gebouwd zijn voor ingewikkelde taken.

4.2.6 De laatste laag: hier neemt het neurale netwerk zijn beslissing

De laatste laag, ook wel de *uitvoerlaag* genoemd, telt zoveel neuronen als dat er oplossingen zijn. In het geval van een neurale netwerk dat olifanten herkent, zijn er twee eindoplossingen:

$$A: \text{de foto is een olifant} \quad (4.14.1)$$

$$B: \text{de foto is geen olifant} \quad (4.14.2)$$

De laatste laag bevat dus twee neuronen.

Als we een netwerk willen schrijven dat een beurskoers van een bepaald aandeel voorspelt, kan bijvoorbeeld de volgende *uitkomstenruimte* gelden:

$$A: \text{de koers zal dalen} \quad (4.15.1)$$

$$B: \text{de koers zal gelijk blijven} \quad (4.15.2)$$

$$C: \text{de koers zal stijgen} \quad (4.15.3)$$

In dit geval bestaat de uitvoerlaag uit drie neuronen.

We werken verder met dit laatste voorbeeld. In de uitvoerlaag worden de uitkomsten van alle berekeningen in het neurale netwerk ‘samengevat’ en neemt het neurale netwerk zijn definitieve beslissing. Afbeelding 4.p hieronder toont drie waardes die uit de activatiefuncties³⁰ van de neuronen kwamen. De waardes in het bovenste, middelste en onderste bolletje zijn overeenkomstig met de kansen op respectievelijk gebeurtenis *A*, *B* en *C*.



Afbeelding 4.p: drie neuronen in de laatste laag van een netwerk met bijbehorende waardes.

Hieruit volgt dat³¹:

$$P(A) = P(\text{koers daalt}) = 0,87 \quad (4.16.1)$$

$$P(B) = P(\text{koers gelijk}) = 0,57 \quad (4.16.2)$$

$$P(C) = P(\text{koers stijgt}) = 0,22 \quad (4.16.3)$$

Tot slot kiest het netwerk het neuron met de hoogste waarde uit. De uitkomst waar dit neuron symbol voor staat, wordt gedefinieerd als de uitkomst van het complete neurale netwerk. (Rodrigues, 2019)

³⁰Voor dit voorbeeld gebruiken wij een activatiefunctie die waardes tussen 0 en 1 als uitkomst kan hebben. Dat is verreweg het meest gebruikelijk. Er zijn uitzonderingen bekend.

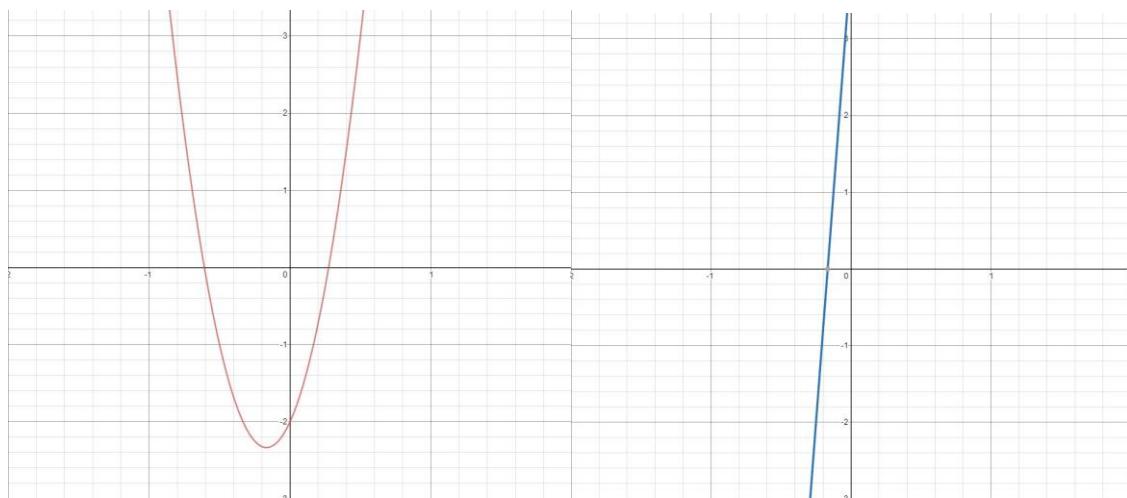
³¹Merk op dat de som van alle ‘kansen’ niet gelijk hoeft te zijn aan 1. Eigenlijk praten we hier over hoe zeker het neurale netwerk van de desbetreffende uitkomst is.

4.2.7 Verliesfunctie & gradient descent: zo leert een neurale netwerk

De verliesfunctie is een functie die de waardes van alle gewichten, biases en invoeren met elkaar combineert. Het is een soort ingebouwde controlestap om te checken hoe dicht de uitkomst van het neurale netwerk bij de daadwerkelijke uitkomst ligt. In een perfecte situatie is de uitkomst van deze functie gelijk aan 0. Immers, het verschil tussen de gevonden uitkomst en de daadwerkelijke uitkomst bij een perfecte voorspelling is 0. Hoe hoger het getal dat uit de verliesfunctie komt, hoe verder het neurale netwerk van de gewenste oplossing afzit. (ML Glossary documentation, sd)

We willen in ons neurale netwerk streven naar een zo laag mogelijke uitkomst voor de verliesfunctie. Eigenlijk zijn we dus op zoek naar het minimum van deze functie. In het geval van een tweedimensionale functie is dit natuurlijk makkelijk: we bepalen de afgeleide en stellen deze gelijk aan 0.

Even een voorbeeld. Stel dat de verliesfunctie $V(x)$ gelijk is aan $12x^2 + 4x - 2$.



Afbeelding 4.q.1: grafiek van $V(x)$. Gemaakt met: (Desmos.com, 2020).

Afbeelding 4.q.2: grafiek van $V'(x)$. Gemaakt met: (Desmos.com, 2020).

Met behulp van algebra kunnen we nu het minimum (het dal) van deze parabool bepalen.

Van de functie $V(x) = 12x^2 + 4x - 2$ is $V'(x) = 24x + 4$ de afgeleide.

Voor het minimum geldt dan:

$$V'(x) = 0 \quad (4.17)$$

dus

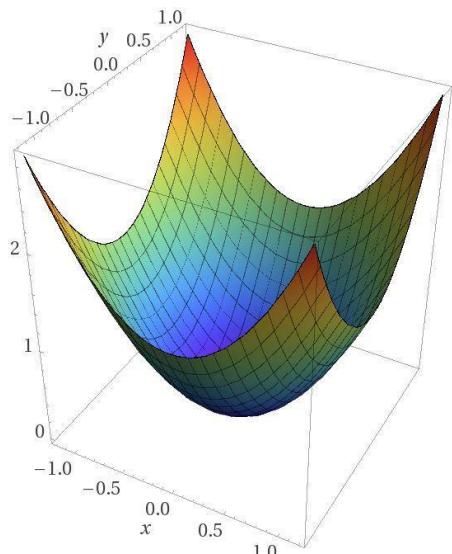
$$24x + 4 = 0 \quad (4.18)$$

geeft

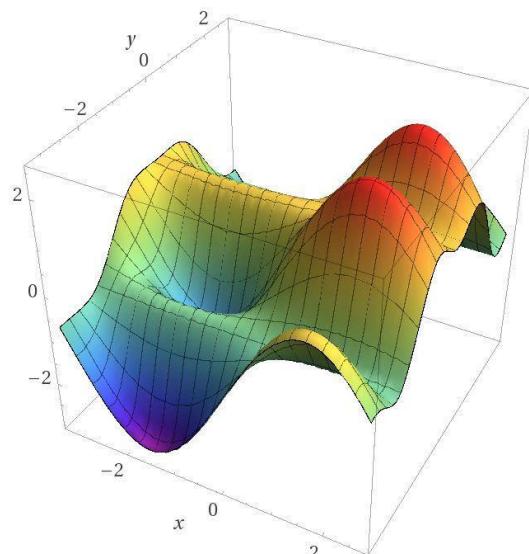
$$24x = -4 \quad (4.19.1)$$

$$x = -\frac{1}{6} \quad (4.19.2)$$

Helaas is het in de meeste gevallen niet zo simpel. Een neuraal netwerk heeft veel verschillende variabelen: zo'n duizend (!) gewichten, neuronen en biases. Al die variabelen zijn niet simpelweg te vangen in één tweedimensionale formule. In de praktijk zijn dit ingewikkelde functies in veel meer dimensies. Omdat grafieken in n -dimensionale ruimtes lastig te visualiseren zijn, versimpelen wij hieronder een verliesfunctie tot drie dimensies.



Computed by WolframAlpha



Computed by WolframAlpha

Figuur 4.r.1 (links): een willekeurige paraboloïde als representatie voor de verliesfunctie. (Paperspace.com).

Figuur 4.r.2 (rechts) een willekeurige driedimensionale vorm als representatie voor de verliesfunctie. (Paperspace.com).

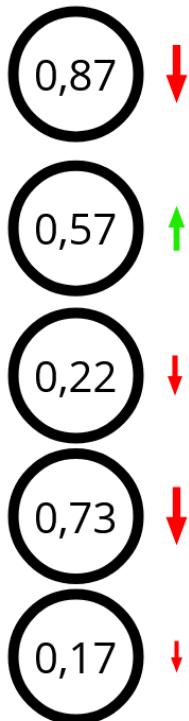
Figuur 4.r.1 is een *paraboloïde*: een soort 3D-parabool. Als we het minimum van deze functie willen vinden, kunnen we ervoor kiezen om op een willekeurige plek een digitaal balletje te plaatsen. Door de 'zwaartekracht' rolt het balletje naar het diepste punt. Op het moment dat het balletje niet meer beweegt, bevindt het zich in het minimum van de grafiek. Het wiskundige algoritme³² achter zo'n rollend balletje, wordt *gradient descent* genoemd.

³² Een algoritme is een serie van instructies, een soort recept, die wordt gebruikt om een bepaald probleem op te lossen of uit te werken.

In figuur 4.r.2 is andere driedimensionale vorm weergegeven. Deze vorm, met diverse pieken en dalen is een realistischer beeld van de verliesfunctie dan een perfecte paraboloïde. In dit geval is het moeilijker om de kleinste waarde voor de verliesfunctie te bepalen met behulp van het rollende balletje. De kans is immers aanwezig dat het balletje in een *lokaal minimum* rolt. Dat is een relatief 'ondiep' dal. Het gradient-descentalgoritme stelt op dat moment dat het absolute minimum is gevonden, omdat het balletje niet meer verder rol, terwijl dit misschien helemaal niet zo is (want er is een kans dat er een dieper dal bestaat). Dit is een van de limitaties van gradient descent. (Ruder, 2020)

4.2.8 Backpropagation: een neuraal netwerk leert van zijn fouten

Backpropagation is de officiële benaming voor het gradient-descentalgoritme (het rollende balletje) dat we hierboven bespraken. Backpropagation helpt het neurale netwerk om dichter bij het juiste antwoord te komen, door de waarde van de verliesfunctie te minimaliseren.



Figuur 4.s: voorbeeld van een aantal neuronen met bijbehorende waardes is de laatste laag. De pijlen geven aan ‘welke kant het getal op moet’ voor het gewenste resultaat.

Laten we backpropagation eerst uitleggen zonder formules.

Neem aan dat het netwerk nog niet goed getraind is. Als we het netwerk een invoer geven waar we de uitvoer al van kennen, kunnen we de uitvoer die het netwerk geeft bijstellen tot een perfect resultaat. In het geval van figuur 4.s willen we dat het tweede neuron de waarde ‘1,00’ krijgt, en de rest van de neuronen de waarde ‘0,00’.

Kijk mee naar figuur a. We willen de 0,57 naar 1,00 krijgen. Er zijn verschillende manieren om dat te doen. We zouden de bias kunnen verhogen, het gewicht kunnen veranderen of de waarden van de activatiefuncties van de neuronen in de lagen voor het tweede neuron kunnen aanpassen. In de praktijk worden al deze opties met elkaar gecombineerd.

Vervolgens willen we de waarden van de andere neuronen omlaag halen. Dit is ingewikkelder, want als we de waarden van de activatiefuncties in de vorige lagen aanpassen, wordt de waarde van het tweede neuron ook weer veranderd. Backpropagation neemt daarom een gemiddelde van alle gewenste aanpassingen om zo een compromis te bereiken.

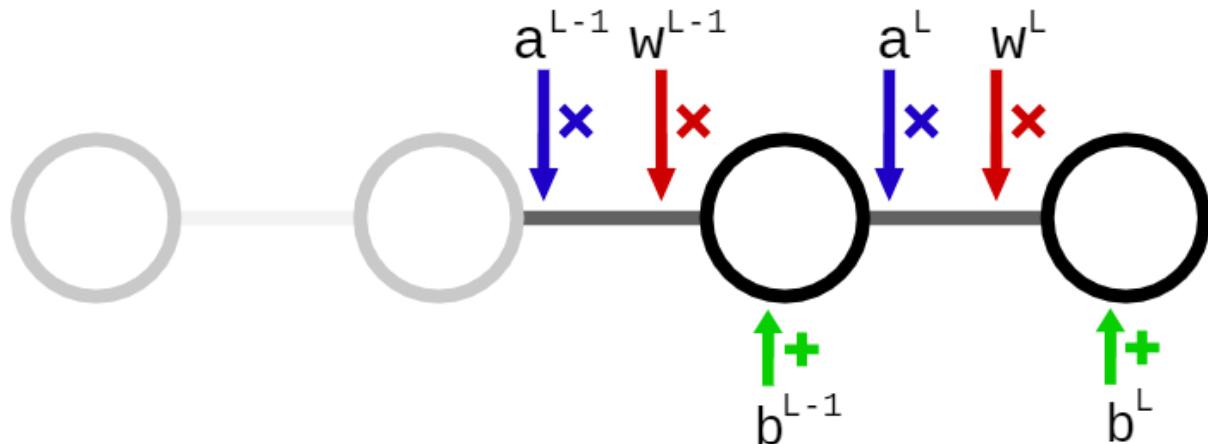
Het hierbovenstaande procedé wordt daarna herhaald voor de andere lagen van het netwerk, zodat ook die lagen beter getraind worden. (3Blue1Brown, 2017)

4.2.8.1 De wiskunde achter backpropagation

Om de wiskunde achter backpropagation simpeler uit te leggen, gaan we kijken naar een neuraal netwerk met maar één neuron per laag (zie figuur 4.t). Laten we kijken naar de laatste twee neuronen. Beide hebben een gewicht en bias die het neuron voor zijn activatiefunctie gebruikt (zie figuur 4.u). Voor de gewichten nemen we hier g^L en voor de biases b^L , waar L voor de laag staat waar het gewicht bij hoort. Ook hebben we per neuron een activatiefunctie. Deze definiëren we als a^L . We focussen in de berekeningen hieronder op het laatste neuron.



Figuur 4.t: voorbeeld van een simpel neuraal netwerk dat bestaat uit vier neuronen.



Figuur 4.u: voorbeeld van de toepassing van het gewicht, de bias en de activatiefunctie in een neuraal netwerk.

In dit netwerk is de waarde van een bepaald neuron a^L gelijk aan:

$$a^L = \sigma(s^L) \quad (4.20)$$

Waarbij geldt:

$$s^L = w^L \cdot a^{L-1} + b^L \quad (4.21)$$

In deze twee formules zitten een hoop variabelen. De som van de gewichten definiëren we als w^L , de activatiefunctie van het desbetreffende neuron als σ , de bias als b^L en het product dat uit de activatiefuncties van de laag ervoor komt als a^{L-1} . Nu we deze letters hebben gedefinieerd, kunnen we ermee rekenen.

We hebben het regelmatig over een verliesfunctie gehad, maar nog nooit er een laten zien. Nu hebben we zo'n verliesfunctie voor het eerst nodig. Het 'verlies' van een netwerk wordt gedefinieerd als:³³

$$C_0 = \frac{1}{2}(a^L - y)^2 \quad (4.22)$$

Waarbij 'y' staat voor de waarde waar wij naar streven. Wij willen weten hoeveel het verlies wordt aangepast als het gewicht verandert. Dit doen we door een verhouding te nemen, waarbij we de verliesfunctie boven het gewicht zetten. Daaruit volgt: (Schuermann, 2016)

$$\frac{\partial C_0}{\partial w^L} \quad (4.23)$$

Wij gebruiken hier de ∂ , ook wel de *partiële afgeleide* genoemd. Deze staat bij w , het gewicht, voor een zeer kleine positieve of negatieve verandering. De ∂ bij de C duidt resulterende verandering van die functie aan, als we dus het gewicht aanpassen.

De verliesfunctie C is samengesteld uit een aantal andere functies. Deze kunnen we substitueren in de zojuist verkregen formule. Als wij dan met dat resultaat de kettingregel toepassen komen wij op de onderstaande formule uit (Silva, 2020). We geven eerst het resultaat, en laten vervolgens zien hoe het resultaat is opgebouwd.

$$\frac{\partial C_0}{\partial w^L} = \frac{\partial s^L}{\partial w^L} \frac{\partial a^L}{\partial s^L} \frac{\partial C_0}{\partial a^L} \quad (4.24)$$

Laten we beginnen met de verhouding $\frac{\partial C_0}{\partial a^L}$. Wij weten wat de verliesfunctie en de activatiefunctie zijn. Als wij nu de verliesfunctie substitueren en doorrekenen, krijgen we de volgende formule:

$$\begin{aligned} C_0 &= (a^L - y)^2 \rightarrow \\ \frac{\partial C_0}{\partial a^L} &= \frac{\partial \frac{1}{2}(a^L - y)^2}{\partial a^L} = \frac{\partial 2(a^L - y) \cdot a^L}{\partial a^L} = 2(a^L - y) \end{aligned} \quad (4.25)$$

³³ Er bestaan ook andere verliesfuncties, maar deze wordt het vaakst gebruikt.

Dezelfde methode gebruiken we voor de verhouding $\frac{\partial a^L}{\partial s^L}$. Deze wordt ook wel op een simpelere manier genoteerd:

$$a^L = \sigma(s^L) \rightarrow \frac{\partial a^L}{\partial s^L} = \sigma'(s^L) \quad (4.26)$$

Tenslotte kunnen we ook de verhouding $\frac{\partial s^L}{\partial w^L}$ omzetten.

$$s^L = w^L \cdot a^{L-1} + b^L \rightarrow \frac{\partial s^L}{\partial w^L} = \frac{\partial w^L \cdot a^{L-1} + b^L}{\partial w^L} = a^{L-1} \quad (4.27)$$

En samenvoegen van alle formules geeft:

$$\frac{\partial C_0}{\partial w^L} = \frac{\partial s^L}{\partial w^L} \frac{\partial a^L}{\partial s^L} \frac{\partial C_0}{\partial a^L} = a^{L-1} \cdot \sigma'(s^L) \cdot 2(a^L - y) \quad (4.28)$$

Dit is de functie voor een specifiek neuron. Als we de waardes van een hele reeks willen weten, moeten we het gemiddelde nemen. Dit is omdat wij, in een netwerk met meerdere neuronen, afhankelijk zijn van meerdere waardes en de uitkomst anders exponentieel zal toenemen. Uiteindelijk krijgen we dus de volgende formule:

$$\frac{\partial C}{\partial w^L} = \frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^L} \quad (4.29)$$

Deze formule lijkt ingewikkeld, en dat is 'ie ook, maar in een echt neuraal netwerk wordt het nog veel ingewikkelder. Hier berekenen wij alleen nog maar de verhouding tussen de verliesfunctie en het gewicht. Daarbovenop zijn wij in dit geval slechts één laag diep gegaan, terwijl neurale netwerken soms wel honderd lagen diep zijn. Toch is de uitwerking hierboven voldoende om een idee te krijgen van hoe backpropagation werkt.

Laten we het nog iets complexer maken. Als we in plaats van één neuron, drie neuronen bekijken, kunnen we de formule als volgt herschrijven: (3Blue1Brown, 2017)

$$\begin{aligned} s^L &= w^L \cdot a^{L-1} + b^L \rightarrow \\ s^L &= w_0^L \cdot a_0^{L-1} + w_1^L \cdot a_1^{L-1} + w_2^L \cdot a_2^{L-1} + b^L \end{aligned} \quad (4.30)$$

En dit leidt tot de matrix die we al eerder hebben aangetoond:

$$s^L = W^L \cdot A^L + B^L \quad (4.31)$$

Hiermee kunnen we uiteindelijk de helling van de verliesfunctie bepalen. Dat kan ons neurale netwerk helpen met het berekenen van resultaten met een hogere accuraatheid.

4.3 Samengevat

Een neuraal netwerk bestaat uit een aantal verschillende onderdelen die nauw met elkaar samenwerken. Een kunstmatig neuraal netwerk vertoont overeenkomsten met een menselijk brein. Door middel van diverse formules en in meerdere stappen doet een neuraal netwerk een voorspelling. Twee algoritmes, gradient descent en backpropagation, zorgen ervoor dat de uitkomst na verloop van tijd zo nauwkeurig mogelijk wordt.

Onthoud tot slot dat we in het grootste deel van dit hoofdstuk gepraat hebben over de weg die slechts één input (dus één foto van een olifant) aflegt. Het hierbovenstaande procedé wordt in de praktijk duizenden keren herhaald, alvorens een neuraal netwerk daadwerkelijk in staat is om een olifant van een niet-olifant te onderscheiden. Een hoop gedoe dus.

5. NEAT: het algoritme dat ons een god maakt

Nu we de basiskennis over aandelenhandel en kunstmatige intelligentie onder de knie hebben, kunnen we beginnen met het vormen van een eerste concept voor ons computerprogramma. Ook leggen we uit waarom we voor een evolutionair algoritme kiezen en niet voor een traditioneel neuraal netwerk. Verder gaan we in op hoe het evolutionaire algoritme NEAT werkt.

Allereerst is het goed om te beseffen dat de beurs zich niet in een magische formule laat vangen. De koers van een aandeel is afhankelijk van diverse factoren. Sommige factoren zijn aardig te voorspellen met behulp van trends en indicatoren, maar andere juist helemaal niet. Deze onvoorspelbare factoren of gebeurtenissen worden *black swans* (zwarte zwanen) genoemd. De Covid-19-pandemie is bijvoorbeeld zo'n zwarte zwaan³⁴, maar ook het ontslag van een Chief Executive Officer (CEO) of Chief Financial Officer (CFO) kan tot een plotselinge koersdaling leiden. (Taleb, 2008)

Ten tweede wordt de prijs van een aandeel deels bepaald door menselijke gevoelens. Toen Elon Musk, de CEO van het autobedrijf Tesla, in 2018 een joint rookte tijdens een online podcast, kelderde de koers van het Tesla-aandeel met 9%. Dit kwam doordat een aantal investeerders het vertrouwen verloor in Musk. Aandelen van Tesla werden massaal verkocht. (Eisenstein, 2018)



Afbeelding 5.a: Elon Musk rookt een joint tijdens een online podcast. Afbeeldingsbron: (PowerfulJRE, 2019).

³⁴ Volgens de officiële *black swan theory* is een pandemie geen zwarte, maar een witte zwaan, omdat men met grote zekerheid kan zeggen dat een pandemie ooit zal plaatsvinden. Het verschil is voor ons eindproduct niet van belang. Daarom gaan we er niet dieper op in.

En tot slot is de prijs van een aandeel direct beïnvloedbaar door menselijke fouten. Uit meerdere wetenschappelijke onderzoeken blijkt dat mensen risico's slecht kunnen inschatten (Forsythe, 2019), (Horowitz, 2013) en (Schneier, 2017). De meeste aandelen worden weliswaar wel via computers verhandeld, maar feit is dat de mens in de meeste gevallen 'op de knop drukt' of de conditionele programmaregels heeft geschreven.

We zijn voor ons programma eigenlijk op zoek naar een algoritme dat zowel met voorspelbare als met onvoorspelbare omstandigheden kan gaan. Het algoritme moet trends kunnen zien en begrijpen, maar moet ook snel kunnen anticiperen op zwarte zwanen en menselijke fouten. Wij denken dat het NEAT-algoritme het best passend is voor onze casus. In dit hoofdstuk leggen we je uit waarom.

5.1 Survival of the fittest, maar dan digitaal

NeuroEvolution of Augmenting Topologies (NEAT) is een *genetisch algoritme*³⁵. Dat wil zeggen: een algoritme gebaseerd op *genetica* (erfelijkheidsleer) en evolutie. Dit specifieke AI-algoritme gaat niet op zoek naar het (diepste) dal van een n -dimensionale grafiek (zoals een 'klassieke' AI, bijvoorbeeld die uit hoofdstuk 4, dat doet), maar maakt gebruik van wijscheden uit de evolutie. Hoe dat dan precies werkt, leggen we nu uit.

Op dag 1 worden 300 virtuele, fictieve beleggers geboren. Iedere belegger krijgt toegang tot 5000 euro. Die 5000 euro mag naar het eigen 'inzicht' van de belegger worden besteed aan forex, aandelen of indices. Aan het einde van de dag worden alle openstaande posities gesloten (en dus alle aandelen die in het bezit van beleggers zijn, verkocht). De virtuele belegger met de meeste winst wordt gekroond tot winnaar.

In de nacht worden de 299 niet-winnaars geëlimineerd. Vervolgens plant de winnaar zich voort. 299 keer, om precies te zijn. Tijdens deze voortplanting vinden *cross-overs* (recombinatie) en *mutaties* plaats. Daarover later meer. Bij het ochtendgloren staan weer 300 beleggers aan de startstreep: 1 vader of moeder, en 299 kinderen. Deze 300 beleggers krijgen weer elk 5000 euro, mogen dit weer uitgeven aan forex, aandelen en indices en aan het einde van de dag wordt weer een winnaar gekozen. Het hierbovenstaande procedé blijft zich steeds herhalen. Eigenlijk spelen we hier Darwins evolutietheorie na.

Na tientallen, honderden of misschien wel duizenden dagen hebben we dan (hopelijk) een superbelegger gecreëerd. Maar wij hebben geen zin om zo lang te wachten. Daarom maken we gebruik van een principe genaamd *backtesting*. In hoofdstuk 3.2 hebben we backtesting al besproken. Kortgezegd trainen we de AI met behulp van historische data (Wikipedia-bijdragers, 2020). We laten de kunstmatige beleggers afreizen naar een tijdstip in het verleden en geven de aandelenkoersen op dat moment als input. Vervolgens laten we de tijd sneller dan normaal verstrijken. Dit tijdreizen, of 'spoelen' in de tijd, is mogelijk omdat we koerswaarden uit het (verre)

³⁵ Ook wel een *evolutionair algoritme* genoemd.

verleden gebruiken. Die waardes zijn voor ons allang bekend en kunnen we dus met een hoog tempo ‘voeren’ aan onze AI. Kortom, NEAT stelt ons dus in staat om in een paar uur tot een paar dagen een heuse belegrobot te fokken.

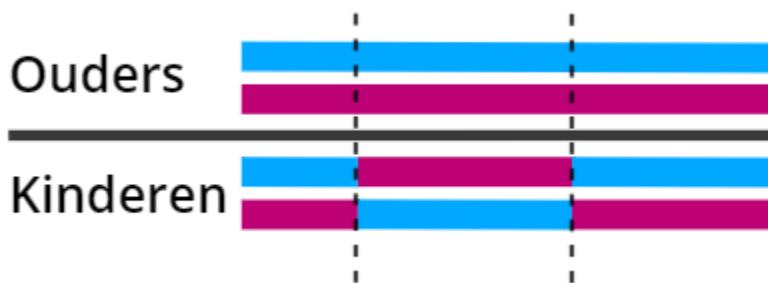
5.2 Hoe werkt die evolutie dan precies?

Laten we eerst even wat begrippen introduceren. De virtuele beleggers noemen we vanaf nu *creaties*. Ons netwerk gaat veel verschillende soorten creaties maken, die allemaal met elkaar zullen *concurreren*. Aan het begin (generatie 1) weet het netwerk nog niks over deze creaties. Daarom gebruikt het netwerk *fitness* (Code Bullet, 2018). Dat is een score die iedere creatie krijgt toegewezen. Hoe hoger de fitness, hoe beter de creatie zijn taak uitvoert (en in dit geval is dat handelen in aandelen).

Ons netwerk begint met precies 300 creaties. Vervolgens test het netwerk iedere creatie, kijkt welke eigenschappen de beste zijn, en kent tot slot een *fitnesswaarde* toe. Dit proces heet *selectie*. (Wikipedia-bijdragers, 2020)

Voor de selectie gebruikt het netwerk een *fitnessfunctie* (Wikipedia-bijdragers, 2020). Deze functie berekent de fitness. Als een bepaalde creatie meer fitness heeft dan een door de maker ingestelde waarde³⁶, wordt deze creatie³⁷ gekozen voor verdere evolutie. De rest van de creaties worden verwijderd. (Wikipedia-bijdragers, 2020)

De volgende stap is *recombinatie*. Dit is, net zoals selectie, een genetisch proces. Het proces is versimpeld afgebeeld in afbeelding 5.b.



Figuur 5.b: voorbeeld recombinatie met twee deellijnen. De ouders zijn de creaties van de oude generatie, de kinderen zijn de creaties van de nieuwe generatie.

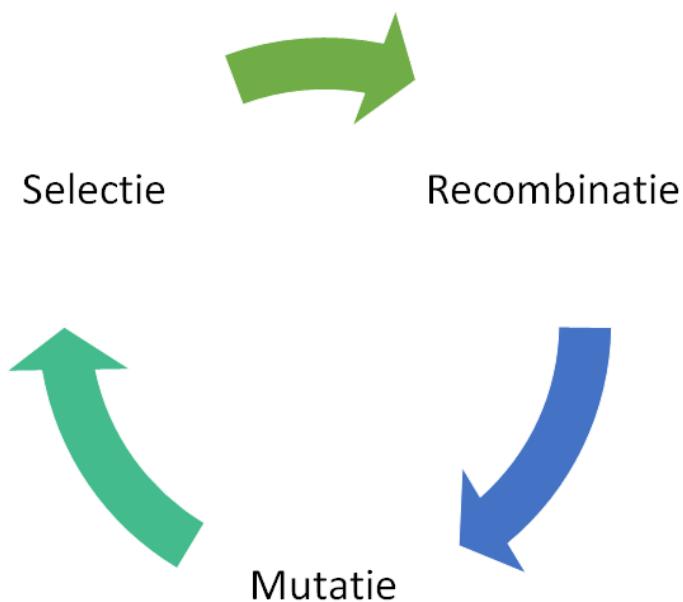
Kort gezegd neemt dit algoritme een deel van de ene ouder en een deel van de andere ouder, en probeert de beste eigenschappen uit beide te krijgen. Dit wordt gedaan door bepaalde delen van het digitale DNA te mixen of om te wisselen.

³⁶Als voor de eerste generatie ($n = 1$) deze waarde gelijk is aan p , dan is voor de tweede generatie deze waarde gelijk aan $2p$. Er geldt: *drempelwaarde* = np .

³⁷De maker kan ook ervoor kiezen om pas te stoppen als x creaties de drempelwaarde hebben bereikt.

En tot slot zorgt het neurale netwerk voor mutaties. Deze mutaties kunnen nieuwe neuronen zijn, nieuwe verbindingen tussen neuronen of andere waardes voor biases. Deze mutaties kunnen ook verbindingen of biases verwijderen.

Het netwerk checkt 'de volgende dag' of de recombinaties en mutaties een verhoging van de fitness met zich meebrengen. Als dat het geval is, ondergaan de nieuwe creaties dezelfde gebeurtenissen als hun ouders. Is dat niet het geval, dan worden de nieuwe creaties vernietigd.



Afbeelding 5.c: dit procedé wordt herhaald totdat er aan een stopconditie wordt voldaan.

Er zijn verschillende stopcondities mogelijk:

- De fitness heeft een bepaalde waarde bereikt;
- Een bepaalde generatie is bereikt;
- De maximaal aanvaardbare rekentijd is overschreden;
- De fitness van creaties stijgt niet meer of nadert een (n-dimensionale) asymptoot;
- Het script is handmatig afgebroken;
- Een combinatie van de bovenstaande dingen. (Wikipedia-bijdragers, 2020)

Een genetisch netwerk gebaseerd op NEAT werkt op veel manieren als elk ander neurale netwerk. De invoeren, uitvoeren, gewichten, biases en verbindingen werken bijvoorbeeld op dezelfde manier en in een genetisch netwerk worden ook activatiefuncties gebruikt. Het enige verschil is de manier waarop beide netwerken leren: een evolutionair algoritme gebruikt fitness en een fitnessfunctie, terwijl een traditioneel neurale netwerk gradient descent gebruikt.

Tot slot is het interessant om te vermelden dat bij gelijke inputs de outputs van een NEAT-netwerk verschillend kunnen zijn. Dit heeft te maken met het feit dat mutaties willekeurig worden

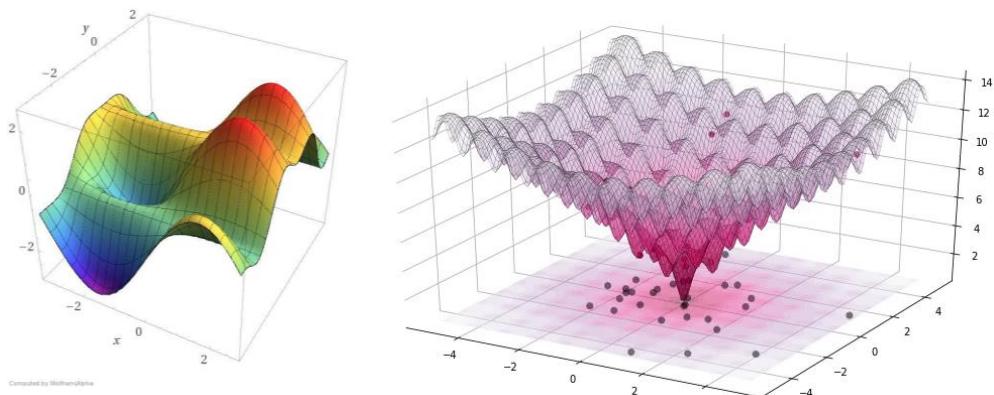
aangebracht. Een traditioneel neuraal netwerk maakt gebruik van pure wiskunde. Hierdoor zal iedere input altijd dezelfde output geven.

5.3 Een evolutionair algoritme in plaats van gradient descent?

Veel traditionele neurale netwerken maken gebruik van gradient descent in plaats van evolutionaire algoritmes. Dit komt waarschijnlijk omdat gradient descent een relatief simpele en efficiënte methode is om een probleem op te lossen. Het inbouwen van evolutie en genetica maakt het programma weliswaar ingewikkelder, maar stelt ons in staat om problemen aan te pakken waar geen complete wiskundige oplossing voor is. Dat is de voornaamste reden waarom wij kiezen voor een evolutionair algoritme in plaats van gradient descent. (Hulstaert, 2018)

Het is overigens goed om te weten dat NEAT en het standaard gradient descent algoritme veel overeenkomsten hebben. Beide methodes maken gebruik van een neuraal netwerk zoals beschreven in hoofdstuk 4. Bovendien vallen beide methodes onder de categorie *mathematical optimization* (*wiskundige optimalisatie*) omdat er wiskundige berekeningen worden gebruikt om een bepaald maximum (of minimum) van een functie te bepalen. Bij een netwerk gebaseerd op gradient descent is dit een verliesfunctie, en bij NEAT is dat een fitnessfunctie.

De manier waarop het absolute minimum of maximum wordt gevonden, is wel anders. Bij NEAT is de kans op vastzitten in een lokaal minimum kleiner dan bij gradient descent. Dit leggen we uit met behulp van twee grafiekjes. Grafiek 5.d toont een visuele weergave van een versimpelde verliesfunctie in een neuraal netwerk gebaseerd op gradient descent. De kans op vastraken in een lokaal minimum is hier aanwezig. Grafiek 5.e toont een versimpelde fitnessfunctie. Merk op dat hier meer lokale minima zijn. Maar door recombinatie en mutaties is de kans veel kleiner dat het netwerk vast komt te zitten in zo'n lokaal minima. Het feit dat bij een evolutionair neuraal netwerk de kans op vastzitten in een lokaal minimum kleiner is dan bij een traditioneel netwerk, is voor ons een andere belangrijke reden om voor evolutie en genetica te kiezen. (Hulstaert, 2018)



Grafiek 5.d (links): visuele weergave van een verliesfunctie in een netwerk gebaseerd op gradient descent.

Afbeeldingsbron: ([Paperspace.com](#)).

Grafiek 5.e (rechts): visuele weergave van een fitnessfunctie in een netwerk gebaseerd op evolutie.

Afbeeldingsbron: ([Pablormier](#)).

De derde reden waarom wij voor NEAT kiezen, is perfectie. We hoeven geen perfect programma te schrijven, om toch winst te maken op de beurs, want de beurs is niet perfect. Aandelenkoersen zijn niet perfect, en mensen die in aandelen handelen zijn helemaal niet perfect. En natuurlijk, een programma dat een oneindig hoog rendement heeft, iedere positie precies op het juiste moment opent en sluit en met alle mogelijke factoren rekening houdt, zou fantastisch zijn. Maar laten we realistisch blijven. Het is niet waarschijnlijk dat wij zo'n programma kunnen maken, omdat zo'n perfect programma misschien wel helemaal niet kan bestaan. Voor ons is een benadering van een perfect programma, een benadering van het maximum van een fitnessfunctie, al genoeg. Dat levert immers ook winst op. NEAT kan ons uitstekend van zo'n benadering voorzien. (Stanley, 2018)

En tot slot: toeval en geluk. Handelen op de beurs is niet alleen maar een kwestie van het analyseren van wiskundige patronen. Geluk speelt soms ook een rol. Dankzij recombinaties en mutaties bevat ons programma ook een geluks- en toevalsfactor en kan het programma de realiteit dus beter imiteren.

6. Programmeerbenodigheden

Het is tijd om ons programma te schrijven. We zijn niet van plan alle theorie achter het programmeren in ons verslag op te nemen. Dit heeft twee redenen:

- 1) **Als we het echt goed willen uitleggen, komen we pagina's tekort.** De wereld van programmeren is ontzettend groot en simpelweg niet in een paar kantjes te vatten;
- 2) **Het is niet extreem interessant.** Een *Von Neumann-cyclus* in een *central processing unit* leest, decodeert en voert alle *bits* en *bytes* uit. Leuk feitje voor de computernerd, maar als we dingen zoals dit zouden bespreken, schieten we het doel van ons profielwerkstuk voorbij.

Toch zullen we niet meteen de code presenteren. In dit hoofdstuk geven we een beknopt overzicht van enkele programmeertermen die in onze programmacode en -uitleg veelvuldig zullen voorkomen.

6.1 Python

Python is een programmeertaal. Dankzij Python kunnen we onze ideeën omzetten naar instructies die een computer kan begrijpen. We hebben voor Python gekozen, omdat Python relatief eenvoudig is en in al onze behoeftes kan voorzien.

6.1.1 Libraries

Een *library* is een verzameling van voorgeprogrammeerde taken (Wikipedia-bijdragers, 2018). Neem bijvoorbeeld het weergeven van de correcte tijd. Dit is voor een programmeur een redelijk ingewikkelde opdracht: men moet onder andere rekening houden met *tijdzones*, zomer- en wintertijd en *schrikkelsecondes* (Computerphile, 2013). Gelukkig heeft iemand in het verleden dit probleem ook gehad en hier een library voor gemaakt. Die library kunnen we gebruiken, zodat we zelf de ingewikkelde code niet meer hoeven te schrijven.

Libraries zijn er in veel soorten en maten. Soms gebruiken we een library voor simpele dingen (maak een grafiekje), soms voor ingewikkelde problemen (zorg dat het neurale netwerk de gegevens correct kan interpreteren). Hieronder staat een overzicht van de libraries die we gebruiken.

6.1.1.1 NEAT-python

Deze library geeft ons toegang tot NEAT. We gebruiken de library ook om onze AI te trainen en om gegevens van onze AI op te vragen.

6.1.1.2 pickle

Deze library gebruiken we om de winnende belegger op te slaan, zodat we deze later weer kunnen gebruiken. *Pickle* is een simpele plugin om bestanden op te slaan en in te laden, zonder dat we moeilijke tussenstappen hoeven te zetten.

6.1.1.3 Matplotlib

Deze library geeft ons de mogelijkheid om eenvoudige grafieken te genereren. De grafieken die *Matplotlib* heeft gemaakt, tonen we onder andere in hoofdstuk 8.1, versiegeschiedenis.

6.1.1.4 xlsxwriter

De grafieken die *Matplotlib* maakt, zijn voor ons niet uitgebreid genoeg. Daarom laten we ook alle data exporteren naar Excel. *Xlsxwriter* regelt dit voor ons. De excelbestanden kunnen we vervolgens gebruiken om complexere grafieken te maken.

6.1.1.5 python-shell

Python-shell zorgt ervoor dat we meerdere programmeertalen met elkaar kunnen combineren. Onze AI is gemaakt in Python en onze GUI is gemaakt in een andere programmeertaal, namelijk *JavaScript*. Normaal gesproken werken deze talen niet samen, maar *python-shell* maakt dit wel mogelijk.

6.1.1.6 yfinance

Geeft ons toegang tot alle (koers)data van Yahoo Finance.

6.1.1.7 forex_python

Forex_python helpt ons met het omrekenen van verschillende muntenheden. Ons programma werkt met aandelenmarkten over de hele wereld. Niet iedere aandelenmarkt gebruikt dezelfde muntenheid. *Forex_python* haalt de laatste wisselkoersen op en maakt dus internationale handel mogelijk.

6.1.1.8 utils / gui_utils

Deze library hebben we speciaal voor ons programma ontwikkeld. *Utils* (voor de AI) en *gui_utils* (voor de GUI) bevatten functies die we vaak gebruiken. Door een library van deze functies te maken, hoeven we die functies niet in elk bestand te kopiëren. Dit maakt het programmeren makkelijker en overzichtelijker.

6.1.1.9 pytz

Deze library helpt ons met het verwerken van tijd en tijdzones, want, zoals we in het begin van dit hoofdstuk beschreven, dat is een flinke klus.

6.2 Classes en variabelen

Classes zijn een soort verzamelingen van een of meerdere objecten. In classes kunnen *variabelen* zitten, maar ook functies. Variabelen zijn waardes die door het programma gelezen en veranderd kunnen worden.

6.3 Loops

Een *loop* is een stuk code dat zichzelf opnieuw uitvoert als het einde van het stuk code is bereikt. Er kunnen extra variabelen aan een loop gebonden zijn om te zorgen dat deze niet oneindig lang blijft doorgaan.

6.4 Electron

Electron is de fundering van onze GUI. Deze speciale, zeer grote³⁸ 'library', ook wel een *framework* genoemd, maakt van onze codebestanden³⁹ een interactieve, visuele representatie. Dit is een service van ons naar de gebruiker. Iemand die onze AI wil gebruiken, heeft dankzij Electron geen verstand nodig van programmeren. (GitHub, sd)

³⁸ Zo'n 200.000 coderegels.

³⁹ Voornamelijk .py (Python), .js (JavaScript), .html (HTML) en .css (CSS).

7. Praktijk: zo werkt ons product

In dit hoofdstuk lichten we belangrijke elementen uit onze code toe. Dit is absoluut geen compleet overzicht. De volledige programmacode hebben we gepubliceerd op onze GitHub-pagina⁴⁰.

Code is voor ons een middel om de kennis uit hoofdstuk 2, hoofdstuk 4 en hoofdstuk 5 om te zetten naar een werkend product. Het is de lezer niet noodzakelijk om alle code tot op de letter te begrijpen. Dit hoofdstuk is vooral bedoeld om een globaal idee te geven van hoe we het programmeren hebben aangepakt.

Wij leggen onze code uit in twee delen, zoals besproken in de inleiding: het AI-gedeelte en het GUI-gedeelte.

7.1 AI-gedeelte

Het stuk over AI bespreken we in vijf delen. Eerst definiëren we enkele variabelen. Daarna bespreken we twee trainingsfases en vertellen we wat er in de testfase gebeurt. Tot slot laten we zien hoe we data uit ons programma kunnen exporteren.

7.1.1 Gedefinieerde variabelen

Allereerst definiëren we een *traderclass*. De traderclass bevat informatie die elke keer dat onze AI wordt gebruikt, opgeslagen moet worden. In dit geval slaat de class de hoeveelheid geld op die de virtuele beleggers bezitten.

```
class Trader:  
    def __init__(self, balance):  
        self.balance = balance  
        self.equities = dict()  
  
    def get_total_balance(self):  
        return self.balance
```

Dan hebben we ook nog een *VirtualMarket-class*. Deze class gebruiken we voor backtesting. De VirtualMarket emuleert een digitale aandelenmarkt en zorgt ervoor dat we terug in de tijd kunnen ‘spoelen’.

De init-functie haalt data op van Yahoo Finance. We downloaden data van een aantal dagen voor de *startdate* tot en met een aantal dagen na de *enddate*. Zo komen we nooit data tekort. De AI zal data tot 31 dagen voor de startdate namelijk ook gebruiken in zijn berekeningen.

⁴⁰ Onze GitHub-pagina is te vinden op: <https://github.com/Toastwer/PWS>

De *while-loop* onderaan haalt foutieve gegevens die we van Yahoo Finance krijgen, weg. De loop vervangt de missende waardes door de waardes van de dag ervoor.

```
class VirtualMarket:
    def __init__(self, startdate, enddate, ticker="MSFT"):
        self.market_info = stocks.download(
            tickers=ticker,
            start=(startdate - timedelta(days=32)),
            end=(enddate + timedelta(days=2)),
            interval="1d")

    [...]

    last_valid = 0
    for i, price in enumerate(self.market_info.get("Close")):
        if price == math.nan or price is None:
            self.market_info.get("Close")[i] = last_valid
            continue
        last_valid = price
```

Vervolgens zoeken we de koerswaardes van een bepaalde dag \square op. Deze krijgen we terug en slaan we op zodat we de gegevens later kunnen gebruiken. De *loop* in het midden bestaat om eventuele foutieve waardes die door de eerste check zijn geglipt, toch nog te verwijderen.

```
def get_day_value(self, d):
    current = None
    current_date = d
    while current is None or current == math.nan:
        current = self.market_info.get("Close").get(
            "{0}-{1}-{2}".format(current_date.year,
        current_date.month, current_date.day))
        current_date -= timedelta(days=1)

    return current
```

Het laatste deel van de VirtualMarket-class zijn een aantal functies om koerswaarden op te vragen die de markt op een bepaalde dag heeft. Zo kunnen we naar de datum van de virtuele dag zelf vragen, maar we willen ook een aantal andere dingen weten, zoals de datum van de dag ervoor en erna.

```
def get_yesterday(self):
    return self.today - timedelta(days=1)

def get_tomorrow(self):
    return self.today + timedelta(days=1)

def increment_day(self):
    self.today += timedelta(days=1)

def decrease_day(self):
    self.today -= timedelta(days=1)

def get_ticker(self):
    return self.ticker

def reset(self):
    self.today = self.startdate
```

Tot slot hebben we nog vier andere variabelen nodig. Een hiervan is START_BALANCE : een constante waarde die aangeeft met hoeveel geld de AI begint. In dit geval is dat 5000 euro.

In de tweede regel slaan we de huidige datum op. De market is de huidige markt waarmee de AI werkt of traint. De markets (meervoud, dus wat anders dan market) worden enkel gebruikt bij het trainen. Dat is een lijst van alle verschillende markten waar de AI op getraind zal worden. We hebben namelijk de mogelijkheid ingebouwd om de AI te trainen op meerdere aandelen achter elkaar.

GENERATION is een andere variabele. Deze wordt enkel voor het trainen gebruikt, om bij te houden bij welke generatie we zijn. De GENERATION en de markets werken in de trainingsfase samen om een lijst met verschillende *tickers*⁴¹ door te werken.

```
START_BALANCE = 5000
date = datetime.now()
market = VirtualMarket(date - timedelta(days=366), date, "RYDAF")
```

⁴¹ Een *ticker* is een verhandelbaar product op de beurs. Bijvoorbeeld een aandeel, index of optie. Het EUR/USD-pair is ook een ticker.

```
markets = []
GENERATION = 0
```

7.1.2 Trainingsfase I: de voorbereiding

Dan is het nu tijd om onze AI te trainen. Daarvoor gebruiken we de functie die hieronder staat. De functie omvat alle stappen die plaats moeten vinden voor en na de training. Het daadwerkelijke trainen wordt dus niet door deze functie geregeld.

We beginnen met het configuratiebestand (zie hoofdstuk 7.1.6) te laden en de benodigde instellingen eruit te halen. Daarna maakt het programma een populatie aan met de instellingen. Het programma voegt een *reporter* toe die ons per generatie informatie zal geven over hoe die generatie het gedaan heeft qua fitness.

Wij hebben besloten om 1500 generaties te trainen. Dit bleek voor ons een goede compromis tussen computerkracht en prestaties van het neurale netwerk.

Aan het eind van de trainingssessie regelt dit codeblok ook dat de beste virtuele belegger (*genoom*) wordt opgeslagen, zodat we die later kunnen laden om grafieken te genereren of om advies te vragen.

```
def run(config_path):
    config = neat.config.Config(neat.DefaultGenome,
neat.DefaultReproduction, neat.DefaultSpeciesSet,
                                neat.DefaultStagnation, config_path)

    pop = neat.Population(config)

    pop.add_reporter(neat.StdOutReporter(True))
    stats = neat.StatisticsReporter()
    pop.add_reporter(stats)

    winner = pop.run(main, 1500)
    with open("winner.pkl", "wb") as f:
        pickle.dump(winner, f)
        f.close()
```

7.1.3 Trainingsfase II: het échte trainen

Als de voorbereiding voorbij is, dan kunnen we de AI trainen met bekende tickers. Dit doen we met een *loop* die versneld door alle dagen van de virtuele markt gaat. We geven de AI alleen de waardes van het moment zelf. De waardes in de toekomst krijgt de AI niet van te voren te zien.

De gegevens die de AI nodig heeft, slaan we op in variabelen. Dat zijn de verschillen met de vorige dag, met 2 dagen geleden, met 5 dagen geleden, met 2 weken geleden en met 31 dagen geleden. We slaan deze gegevens op in variabelen om de snelheid van ons programma te bevorderen.

```
while run:  
    market.tick_day()  
    market.increment_day()  
  
    today = market.get_today()  
    today_value = market.get_day_value(today)  
    tomorrow_value = market.get_day_value(market.get_tomorrow())  
  
    delta_1d = today_value -  
              market.get_day_value(market.get_yesterday())  
    delta_2d = today_value -  
              market.get_day_value(today - timedelta(days=2))  
    delta_5d = today_value -  
              market.get_day_value(today - timedelta(days=5))  
    delta_14d = today_value -  
               market.get_day_value(today - timedelta(days=14))  
    delta_31d = today_value -  
               market.get_day_value(today - timedelta(days=31))
```

Nu *lopen* we door alle verschillende traders die we hebben (300 per trainingssessie) en geven we de eerder opgehaalde gegevens door aan de AI zodat hij kan bepalen wat hij wil gaan doen. We activeren zijn netwerk en krijgen van het netwerk twee outputwaardes.

```
for i, trader in enumerate(traders):  
    output = networks[i].activate((delta_1d, delta_2d, delta_5d,  
delta_14d, delta_31d))
```

Als een van deze outputwaardes gelijk is aan 0,5, dan beschouwen we dit neuron als geactiveerd. Als het eerste neuron geactiveerd is, dan zal de virtuele belegger de keuze nemen om 500 euro in het desbetreffende aandeel te investeren. Als het eerste neuron niet groter dan 0,5 is maar het tweede wel, dan zal de virtuele belegger slechts 100 dollar investeren. Wanneer beide neuronen niet geactiveerd zijn, dan zal de virtuele belegger deze dag niks doen.

De virtuele belegger neemt positie in. Vervolgens halen we de slotkoers op. Aan de hand van de slotkoers berekenen we hoeveel geld de virtuele belegger verdiend of verloren heeft. Dit winst voegen we aan zijn rekening toe, het verlies ervan af.

Hierna volgt nog een check of het saldo van de virtuele belegger positief is. Als dit niet zo is, dan heeft de de virtuele belegger ‘verloren’ en verwijderen we hem. Hij heeft dan gefaald, en zijn fitness wordt naar een hele lage waarde gezet om het algoritme te laten zien dat de methode die de virtuele belegger heeft gebruikt, niet werkt.

```
if output[0] >= 0.5:  
    traders[i].balance +=  
        amount500 * (tomorrow_value - today_value)  
elif output[1] >= 0.5:  
    traders[i].balance +=  
        amount100 * (tomorrow_value - today_value)  
  
if trader.get_total_balance() <= 0:  
    ge[i].fitness = -99999999  
    ge.pop(i)  
    traders.pop(i)  
    networks.pop(i)  
    continue
```

We hebben, na wat proberen, ervoor gekozen om de fitness van de AI te definiëren door de hoeveelheid geld op zijn rekening te delen door 100, en dat tot de macht drie te nemen. Een kleine verbetering zorgt zo voor een grote motivatie voor de AI om door te gaan.

```
ge[i].fitness = (trader.get_total_balance() / 100) ** 3
```

7.1.4 Testfase

Nu kan de AI worden getest op onbekende tickers. De code die dit mogelijk maakt, lijkt veel op de code die in hoofdstuk 7.1.3 staat. De twee grote verschillen zijn dat we nu alleen de vooraf gedefinieerde ‘beste’ virtuele belegger aan het werk zetten, in plaats van 300 virtuele beleggers en dat deze beste belegger niet meer evolueert.

7.1.5 Grafieken en data-export

Als de testfase succesvol is afgerond, kunnen we data exporteren en grafieken genereren. Per getest aandeel maken we twee grafieken (datum-balans en datum-percentage) en ook twee excelbestanden. Dat doen we met onderstaande code. De percentagegrafiek is voor ons het meest zinvol, omdat het exacte saldo van de AI niet echt van belang is, zolang hij procentueel maar winst maakt.

```

graph(dates, balances, "balance")

graph(dates, percentages, "percentage", dates, percentages_stock)

if not excel(dates, balances, "balance",
             print("Error in generating xlsx file for balances.")

if not excel(dates, perc, "percentage", percentages_stock):
    print("Error in generating xlsx file for percentages.")

```

De code hieronder geeft ons een koop- of verkoopadvies. Dit advies wordt gebaseerd op de virtuele belegger met de hoogste fitness. De code maakt voor de virtuele belegger met de hoogste fitness een neurale netwerk aan met exact dezelfde instellingen als we hadden tijdens de training. Uit de output van dat neurale netwerk rolt een beleggingsadvies.

```

def advice(winner_genome, neat_config, stock_data, day):
    network = neat.nn.FeedForwardNetwork.create(winner_genome,
neat_config)

    closes = stock_data.get("Close")
    today_value = get_stock_value(closes, day)

    delta_1d = today_value - get_stock_value(closes, day -
timedelta(days=1))
    delta_2d = today_value - get_stock_value(closes, day -
timedelta(days=2))
    delta_5d = today_value - get_stock_value(closes, day -
timedelta(days=5))
    delta_14d = today_value - get_stock_value(closes, day -
timedelta(days=14))
    delta_31d = today_value - get_stock_value(closes, day -
timedelta(days=31))

    output = network.activate((delta_1d, delta_2d, delta_5d, delta_14d,
delta_31d))
    return output

```

7.1.6 NEAT-configuratie

NEAT heeft een *configuratiebestand* nodig om te werken. In dit bestand staan onder andere de eigenschappen van het neurale netwerk. Naast onze programmacode is ook dit bestand nodig als iemand ons onderzoek wil repliceren.

Wij zijn uitgegaan van de standaardconfiguratie met enkele aanpassingen. Om te bepalen welke aanpassingen we moesten doen, hebben we gekeken naar de waarden en besloten of deze accuraat en geschikt zouden zijn voor ons programma. Het kostte veel tijd om geschikte mutatiekansen (`mutation_rates`) te vinden. Deze mutatiekansen zijn verantwoordelijk voor het tegengaan van een lokaal minimum. We hebben ook de *fitness threshold* verhoogd zodat de AI pas stopt als het virtuele jaar voorbij is en niet bij een bepaald niveau van de fitnessfunctie.

```
[NEAT]
fitness_criterion      = max
fitness_threshold       = 9999999999999999999999999999
pop_size                = 300
reset_on_extinction     = False

[DefaultGenome]
# node activation options
activation_default       = tanh
activation_mutate_rate   = 0.3
activation_options        = tanh

# node aggregation options
aggregation_default     = sum
aggregation_mutate_rate = 0.0
aggregation_options      = sum

# node bias options
bias_init_mean           = 0.0
bias_init_stdev          = 1.0
bias_max_value           = 30.0
bias_min_value           = -30.0
bias_mutate_power        = 0.5
bias_mutate_rate          = 0.7
bias_replace_rate         = 0.1

# genome compatibility options
compatibility_disjoint_coefficient = 1.0
```

```

compatibility_weight_coefficient = 0.5

# connection add/remove rates
conn_add_prob = 0.5
conn_delete_prob = 0.5

# connection enable options
enabled_default = True
enabled_mutate_rate = 0.01

feed_forward = True
initial_connection = full

# node add/remove rates
node_add_prob = 0.2
node_delete_prob = 0.2

```

In het tweede deel van de NEAT-config stellen we eigenschappen van het neurale netwerk in. Wij geven vijf datapunten als input (het verschil met 1 dag geleden, met 2 dagen geleden, met 5 dagen geleden, met 2 weken geleden en met 31 dagen geleden) en wensen twee gegevens als output (500 euro investeren of 100 euro investeren). Kleine tests wijzen uit dat tussenlagen voor ons geen meerwaarde bieden. Daarom staan die op nul.

```

# network parameters
num_hidden = 0
num_inputs = 5
num_outputs = 2

# node response options
response_init_mean = 1.0
response_init_stdev = 0.0
response_max_value = 30.0
response_min_value = -30.0
response_mutate_power = 0.0
response_mutate_rate = 0.0
response_replace_rate = 0.0

# connection weight options
weight_init_mean = 0.0
weight_init_stdev = 1.0
weight_max_value = 30

```

```
weight_min_value      = -30
weight_mutate_power   = 0.5
weight_mutate_rate    = 0.8
weight_replace_rate   = 0.1

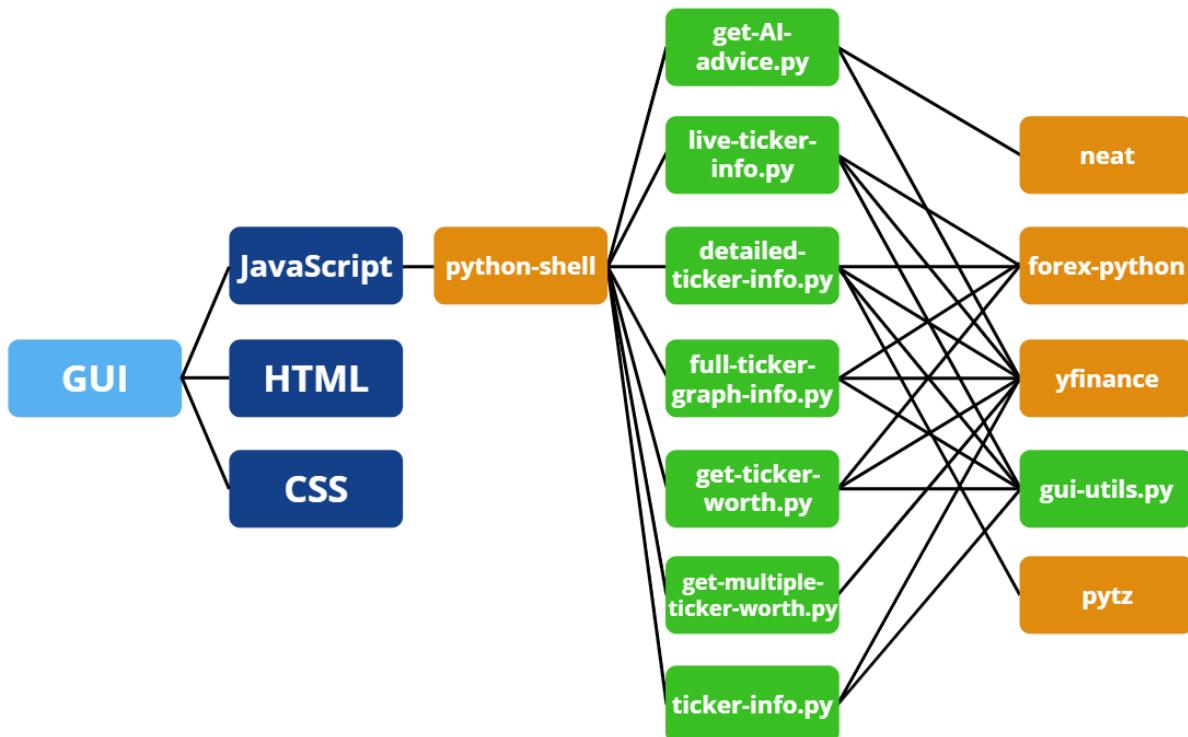
[DefaultSpeciesSet]
compatibility_threshold = 3.0

[DefaultStagnation]
species_fitness_func = max
max_stagnation       = 20
species_elitism       = 2

[DefaultReproduction]
elitism                = 2
survival_threshold     = 0.2
```

7.2 Het GUI-deel

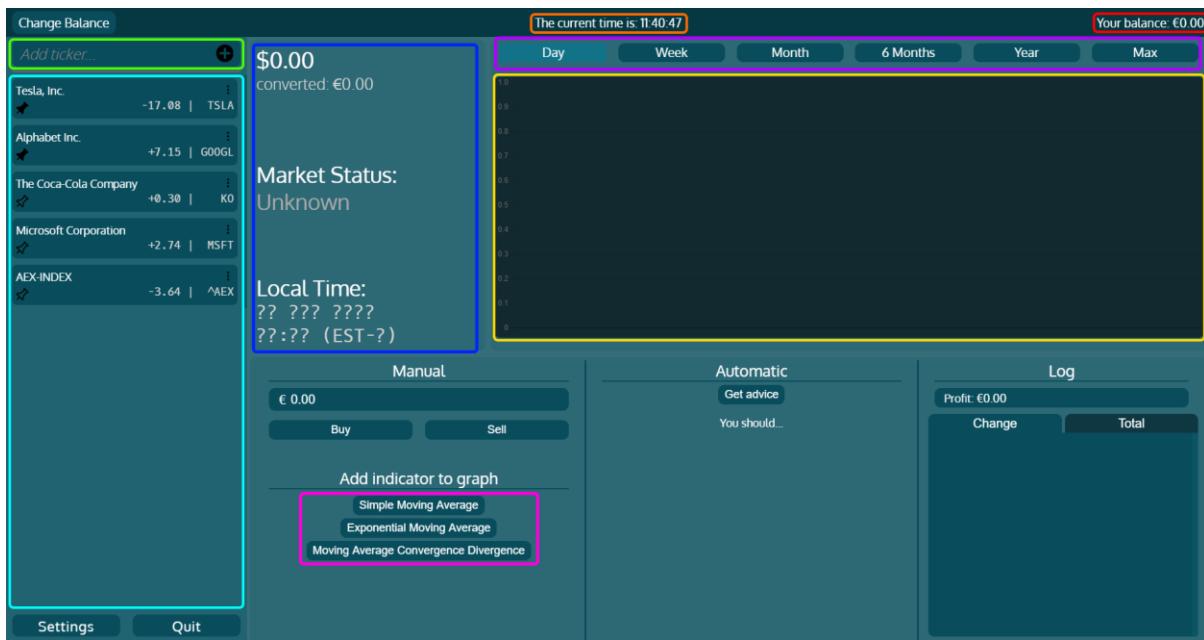
De GUI (*Graphical User Interface*) is het gereedschap waarmee de gebruiker (wij dus) de interacties met het programma uitvoeren. De GUI bestaat uit een aantal onderdelen die nauw met elkaar samenwerken. Alle onderdelen van onze GUI staan in afbeelding 8.a. We bespreken hier alleen de code van de belangrijkste onderdelen. De volledige code staat op onze GitHub-pagina⁴².



Afbeelding 8.a: een overzicht van hoe alle GUI-code op elkaar aansluit. Blauw is het gedeelte dat de gebruiker kan zien, groen zijn python-files en oranje zijn libraries.

Maar hoe ziet onze GUI er eigenlijk uit? Afbeelding 8.b is een screenshot van de GUI. Enkele onderdelen zijn met kleuren omcirkeld. Daar gaan we nu meer over vertellen.

⁴² Onze GitHub-pagina is te vinden op: <https://github.com/Toastwer/PWS>



Afbeelding 8.b: een overzicht van verschillende GUI-elementen.

7.2.1 Rood: de balance

De *balance* is de hoeveelheid geld die de gebruiker beschikbaar heeft om mee te traden. Onderstaande functie gebruiken we om de balance te definiëren en aan te kunnen passen. De maximale grootte hebben we gesteld op $bal = 1000000000000000$. Dit is om te voorkomen dat de computer moet werken met oneindig grote getallen.

```
let balance = 0;
function setBal(bal) {
    if(bal > 1000000000000000)
        bal = 1000000000000000;

    balance = bal;
    $(".balance .value").html(EUR.format(bal));
    setConfig("balance", balance);
}
```

7.2.2 Oranje: de tijd

Tijd speelt in de aandelenmarkt een grote rol en daarom is het belangrijk dat we deze tijd opnemen in ons programma. Onderstaand stukje code gebruiken we voor het correct weergeven van de tijd.

```

setInterval(() => {
    const date = new Date();
    $(".time .value").html(date.getHours() + ":" +
    (date.getMinutes() < 10 ? "0" : "") + date.getMinutes() + ":" +
    (date.getSeconds() < 10 ? "0" : "") + date.getSeconds());
    if(date.getSeconds() == "0")
        updateLiveInfo(selectedTicker);
}, 1000);

```

7.2.3 Geel (1): de vormgeving van de grafiek

Het volgende codeblok is verantwoordelijk voor het gereedmaken van de omgeving waar we een grafiek in kunnen laten zien. We definiëren hier enkele eigenschappen van de toekomstige grafiek, zoals het feit dat de grafiek in 2D moet zijn en dat deze gebogen lijnen moet bevatten. De hier onderstaande functie is alleen verantwoordelijk voor de vormgeving van de grafiek en dus niet voor de inhoud.

```

const ctx = document.getElementById("chart").getContext("2d");
const chart = new Chart(ctx, {
    type: 'line',
    options: {
        responsive: true,
        maintainAspectRatio: false,
        elements: {
            line: {
                tension: 0.1
            }
        },
        animation: {
            duration: 0
        }
    }
});

```

7.2.4 Even tussendoor: de backend

We moeten het nu over een stukje code hebben dat niet direct zichtbaar is. Dit wordt de *backend*⁴³ genoemd.

⁴³ Het gedeelte van de code dat niet zichtbaar is, maar data vasthoudt en verandert.

Hieronder staat de functie die ervoor zorgt dat wij een ticker kunnen maken en toevoegen aan ons programma. Deze functie laadt een python bestand (`ticker-info.py`). Wij kunnen aangeven voor welke ticker wij data willen hebben. Vervolgens geeft het bestand data terug zoals de volledige naam en het verschil in marktwaarde (van een dag).

```
function addTicker(ticker) {
    ticker = encrypt(ticker);

    const options = {
        args: [ticker]
    }
    showLoading();
    PythonShell.run('../GUI Scripts/ticker-info.py', options,
function(err, results) {
    if (err) {
        console.log(err);
        sortTickers();
        hideLoading();
    } else {
        const data = {
            title: results[0],
            ticker: ticker.toUpperCase(),
            change: results[1].toString().charAt(0) == "-" ?
results[1] : "+" + results[1]
        };
        tickers.set(ticker, data);
        sortTickers();

        hideLoading();

        setConfig("tickers", Array.from(tickers.keys()));
    }
}) ;
}
```

7.2.5 Lichtblauw: de tickers op een rij

Het volgende codeblok zorgt ervoor dat alle tickers gesorteerd goed in de lijst aan de linkerkant (in het lichtblauwe gedeelte) van het programma komen te staan.

```
function sortTickers() {
    tickers.forEach((data, _) => {
        if(decrypt(data.ticker).length > maxLength)
            maxLength = decrypt(data.ticker).length;
    }) ;

    const container = $(".side-bar .tickers");
    let content = "";

    pinnedTickers.forEach(ticker => {
        const data = tickers.get(ticker);
        if(data != undefined)
            content += `<div id="${ticker}" ${selectedTicker ===
ticker ? "class='active'" : ""}><span
class="title">${data.title}</span><span
class="more"><button></button></span><span class="pin"><button
class="active" id="${data.ticker}-pin"></button></span><span
class="info">${data.change} |
${equalSpacing(decrypt(data.ticker))}</span></div>`;
    });
    tickers.forEach((data, ticker) => {
        if(!pinnedTickers.includes(ticker)) {
            content += `<div id="${ticker}" ${selectedTicker ===
ticker ? "class='active'" : ""}><span
class="title">${data.title}</span><span
class="more"><button></button></span><span class="pin"><button
id="${data.ticker}-pin"></button></span><span
class="info">${data.change} |
${equalSpacing(decrypt(data.ticker))}</span></div>`;
        }
    });

    container.html(content);
}
```

7.2.6 Lichtgroen: voeg een ticker toe

De twee functies hieronder zijn voor het invoerveld bovenaan de lijst. Als een gebruiker op de plusknop drukt, wordt de bovenste functie uitgevoerd. Als een gebruiker niet op de plusknop maar op de entertoets drukt, wordt de onderste functie uitgevoerd.

Beide functies voeren de `addTicker()`-functie uit met de ingevoerde ticker en verwerken de data die de gebruiker invoert in het lichtgroene gedeelte.

```
$("#addTicker").on("click", () => {
    const input = $("#ticker-input");
    addTicker(input.val());
    input.val("");
});

$("#ticker-input").on("keypress", function(e) {
    const keyCode = e.keyCode || e.which;
    if(keyCode == 13) {
        addTicker($(this).val());
        $(this).val("");
    }
});
```

7.2.7 Donkerblauw: naam, waarde, geldeenheid en tijd

Het onderstaande codeblok is een grote functie, maar dat heeft een reden. Deze functie laadt alle gegevens in als de gebruiker een bepaalde ticker uit de lijst selecteert. Dit doet hij via de python-file `detailed-ticker-info.py`. In dit bestand wordt nogmaals de volledige naam van de ticker gegeven, maar ook de actuele waarde, de bijbehorende geldeenheid, de lokale tijd en alle data voor het maken van de dag-grafiek. Deze functie heeft direct invloed op alle objecten in het donkerblauwe gedeelte.

```
function updateInfo() {
    rangeData.clear();
    rangeLabel = "";
    if(detailedGraphFetch != null) {
        detailedGraphFetch.childProcess.kill("SIGINT");
        detailedGraphFetch = null;
    }
}
```

```
if(currentRange != "day") {
    $("#" + currentRange).removeClass("active");
    $("#day").addClass("active");
    currentRange = "day";
}

if(selectedTicker == "") {
    clearInterval(timeOffset);
    liveInfo = false;
    $(".main .info").html(`<div class="worth value">$0.00</div>
    <div class="converted">converted: <span
class="value">€0.00</span></div>
    <div class="break"></div>
    <div class="status">Market Status: <span
class="value">Unknown</span></div>

    <div class="local-time">
        <div class="header">Local Time:</div>
        <div class="data">
            <span class="date">?? ?? ????</span>
            <span class="time">?:?:? (EST-?)</span>
        </div>
    </div>`);
} else {
    clearInterval(liveWorthInterval);
    const ticker = selectedTicker;
    showLoading();
    PythonShell.run('../GUI Scripts/detailed-ticker-info.py',
{args: [ticker]}, function(err, results) {
        if (err) throw err;

        const data = {
            title: results[0],
            ticker: decrypt(ticker).toUpperCase(),
            value: results[1],
            symbol: String.fromCharCode.apply(String,
results[2].split("|")));
    });
}
```

```

        converted: results[3],
        localDate: results[4],
        offset: parseInt(results[5]),
        graphDate: results[6],
        graphValue: results[7],
        graphSplits: results[8],
        graphDividends: results[9]
    } ;

    const dayData = sortGraphData(data.graphDate,
data.graphValue, data.graphSplits, data.graphDividends);
    rangeData.set("day", dayData);
    rangeLabel = data.title;

    clearInterval(timeOffset);

    timeOffset = setInterval(() => {
        const date = new Date();
        const time = date.getHours() - 1 + data.offset +
"::" + (date.getMinutes() < 10 ? "0" : "") + date.getMinutes();

        if(isNaN(data.offset))
            $(".local-time .data .time").html(`--:-- (EST-
~)`);
        else
            $(".local-time .data .time").html(` ${time}
(EST${data.offset >= 0 ? "+" + data.offset : data.offset})`);

    }, 1000);

    liveInfo = true;
    updateLiveInfo(data.ticker, data.symbol);

    $(".main .info").html(`<div class="worth
value">${data.symbol}${data.value}</div>
<div class="converted">${data.symbol !== "€" ?
`converted: <span class="value">€${data.converted}</span>` :
""}</div>
<div class="break"></div>

```

```

        <div class="status">Market Status: <span
class="value closed">Closed</span></div>

        <div class="local-time">
            <div class="header">Local Time:</div>
            <div class="data">
                <span class="date">${data.localDate}</span>
                <span class="time">???:?? (EST-?)</span>
            </div>
        </div>`);

if(indicator != "") {
    chart.data = {
        labels: dayData[0],
        datasets: [
            {
                label: rangeLabel,
                borderColor: 'rgb(0, 200, 100)',
                data: dayData[1],
                backgroundColor: 'rgba(0, 100, 0, 0.1)'
            }, {
                label: "Dividends",
                borderColor: 'rgb(200, 200, 0)',
                data: dayData[2],
                backgroundColor: 'rgba(200, 200, 0, 0.2)'
            }, {
                label: "No Data",
                borderColor: 'rgb(200, 0, 0)',
                data: dayData[3],
                backgroundColor: 'rgba(200, 0, 0, 0.2)'
            }, {
                label: indicator,
                borderColor: 'rgb(0, 50, 200)',
                data: calcIndicator(indicator, dayData),
                backgroundColor: 'rgba(0, 0, 0, 0)'
            }
        ]
    };
} else {
    chart.data = {

```

```

        labels: dayData[0],
        datasets: [
            {
                label: rangeLabel,
                borderColor: 'rgb(0, 200, 100)',
                data: dayData[1],
                backgroundColor: 'rgba(0, 100, 0, 0.1)'
            }, {
                label: "Dividends",
                borderColor: 'rgb(200, 200, 0)',
                data: dayData[2],
                backgroundColor: 'rgba(200, 200, 0, 0.2)'
            }, {
                label: "No Data",
                borderColor: 'rgb(200, 0, 0)',
                data: dayData[3],
                backgroundColor: 'rgba(200, 0, 0, 0.2)'
            }
        ]
    };
}
chart.update();

loadFullGraphData(ticker);

hideLoading();
}) ;
}
}

```

7.2.8 Geel (2): de inhoud van de grafiek

De functie hieronder is een aanvulling op de bovenstaande `update-ticker()`-functie. Deze functie wordt direct uitgevoerd nadat de update-functie klaar is. In deze functie wordt de Python-file `full-ticker-graph-info.py` geladen. Dit bestand geeft dan de waardes voor de overige grafieken terug. Deze functie is grotendeels gericht op de backend. De functie slaat data op en pas als door de gebruiker om deze data gevraagd wordt, zal hij deze tonen in de grafiek.

```

function loadFullGraphData(ticker) {
    detailedGraphFetch = PythonShell.run('../GUI Scripts/full-
ticker-graph-info.py', {args: [ticker]}, function(err, results) {

```

```

if (err) throw err;

const ranges = ["week", "month", "6months", "year", "max"];

if(results == null)
    return;

for (let i = 0; i < results.length; i += 4) {
    const range = ranges[i / 4];

    rangeData.set(range, sortGraphData(results[i],
results[i + 1], results[i + 2], results[i + 3]));
}
}
}

```

7.2.9 Paars (1): de knoppen boven de grafiek

Het codeblok hieronder is verantwoordelijk voor de knoppen boven de grafiek.

```

$(".select-range").on("click", "button", function() {
    const btn = $(this);
    if(!btn.hasClass("active")) {
        $("#" + currentRange).removeClass("active");
        currentRange = btn.attr("id");
        btn.addClass("active");
        changeRange(currentRange);
    }
})

```

7.2.10 Paars (2): de backend van de knoppen boven de grafiek

Deze functie sluit aan bij de vorige en is verantwoordelijk voor het daadwerkelijk veranderen van de data in de grafiek. Dat doet hij door de eerder opgeslagen data van de **update-info()**-functie en van de **loadFullGraphData()**-functie op te roepen, en ze te laten zien in de grafiek. Deze functie regelt dus de backend van het paarse gedeelte.

```

function changeRange(_range) {
    if(selectedTicker == "") {
        return;
    }

    range = _range;

    if(rangeData.has(range)) {
        const data = rangeData.get(range);
        if(indicator != "") {
            chart.data = {
                labels: data[0],
                datasets: [ {
                    label: rangeLabel,
                    borderColor: 'rgb(0, 200, 100)',
                    data: data[1],
                    backgroundColor: 'rgba(0, 100, 0, 0.1)'
                }, {
                    label: "Dividends",
                    borderColor: 'rgb(200, 200, 0)',
                    data: data[2],
                    backgroundColor: 'rgba(200, 200, 0, 0.2)'
                }, {
                    label: "No Data",
                    borderColor: 'rgb(200, 0, 0)',
                    data: data[3],
                    backgroundColor: 'rgba(200, 0, 0, 0.2)'
                }, {
                    label: indicator,
                    borderColor: 'rgb(0, 50, 200)',
                    data: calcIndicator(indicator, data),
                    backgroundColor: 'rgba(0, 0, 0, 0)'
                } ]
            };
        } else {
            chart.data = {
                labels: data[0],
                datasets: [ {
                    label: rangeLabel,

```

```

        borderColor: 'rgb(0, 200, 100)',
        data: data[1],
        backgroundColor: 'rgba(0, 100, 0, 0.1)'
    }, {
        label: "Dividends",
        borderColor: 'rgb(200, 200, 0)',
        data: data[2],
        backgroundColor: 'rgba(200, 200, 0, 0.2)'
    }, {
        label: "No Data",
        borderColor: 'rgb(200, 0, 0)',
        data: data[3],
        backgroundColor: 'rgba(200, 0, 0, 0.2)'
    } ]
}
};

chart.update();
} else {
    showLoading();
    const load = setInterval(() => {
        if(rangeData.has(range)) {
            hideLoading();
            clearInterval(load);
            changeRange(range);
        }
    }, 500);
}
}

```

7.2.11 Roze: indicatoren

De onderstaande functie berekent de indicatoren voor de grafiek. We hebben alleen de SMA-berekening (zie hoofdstuk 2.2.3.1) ingebouwd. De formule voor de SMA is:

$$SMA_n = \frac{(P_x + P_{x-1} + P_{x-2} + \dots + P_n)}{N} \quad (7.1)$$

Merk op dat deze formule er in programmeertaal (het stuk met de lichte achtergrond) heel anders uitziet. Formules zoals de SMA zijn lastig te vertalen in een programmeertaal. Dit is slechts één voorbeeld van de obstakels die wij tijdens het programmeren moesten overbruggen.

De rest van de functie pakt de invoer van de knoppen (roze gedeelte) en laat de gevraagde indicator zien in de grafiek (gele gedeelte).

```
function calcIndicator(indicator, data) {
    const values = [];
    data[1].forEach(value => {
        values.push(parseFloat(value));
    });
    data[2].forEach((value, i) => {
        if(values[i] == null && value != null)
            values[i] = parseFloat(value);
    });

    const output = [];
    values.forEach((value, i) => {
        if(indicator === "SMA") {
            let total = 0;
            let j = i;
            while (j > 0 && j > (i - 50)) {
                total += values[j];
                j--;
            }

            output.push(Math.round(total / Math.min(i, 50) * 100) / 100);
        } else if(indicator === "EMA") {
            [...]
        } else if(indicator === "MACD") {
            [...]
        }
    });
    return output;
}
```

7.3 Afsluiting

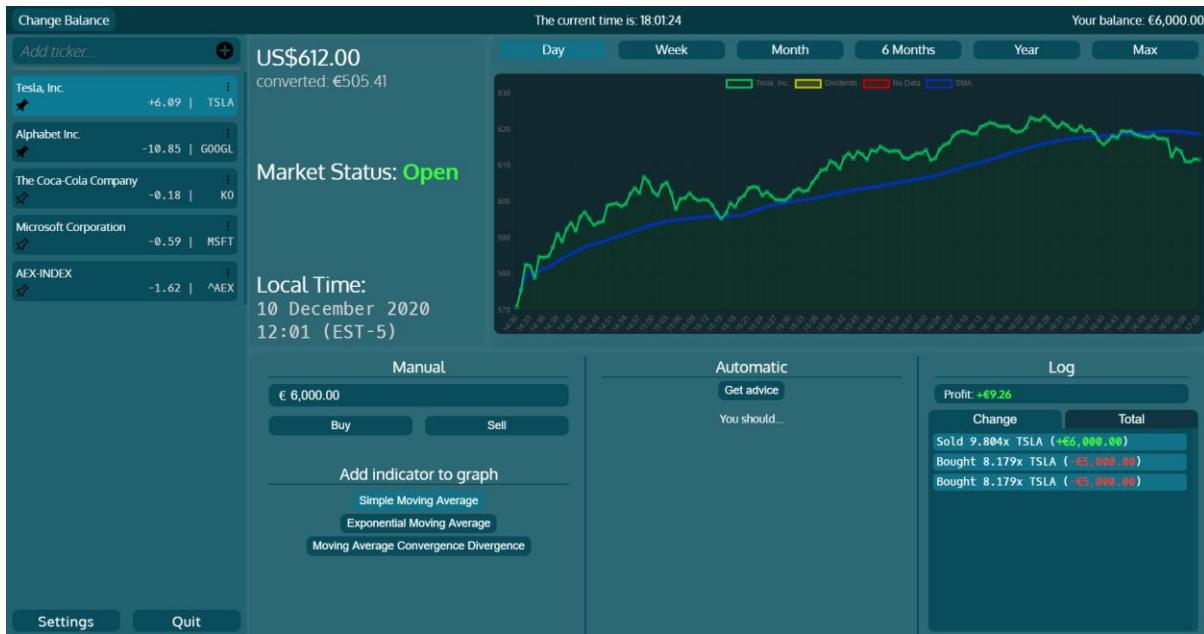
Onze code is dus een samenspel van veel verschillende functies (codeblokken). Deze functies maken het mogelijk om de theorie die we geleerd hebben in hoofdstuk 2, hoofdstuk 4 en hoofdstuk 5 om te zetten naar een werkend product.

De complete programmacode is op te vragen via onze GitHub-pagina:

<https://github.com/Toastwer/PWS>

8. Evaluatie en conclusies

De AI staat. We hebben een programma geschreven dat met succes aandelen kan verhandelen op de beurs. Maar werkt het? En is ons genetisch algoritme beter dan een mens?



Afbeelding 8.a: een screenshot van de GUI van ons programma.

8.1 AI-versiegeschiedenis

Allereerst is het goed om te weten dat de code die we in hoofdstuk 7 bespreken het resultaat is van wekenlang puzzelen met de juiste instellingen, parameters en functies. Ons profielwerkstuk is te kort om alle versies die we hebben gebouwd uitgebreid te bespreken. Daarom geven we hieronder een kort overzicht van de verschillende versies van ons programma weer⁴⁴.

v1.0

De AI krijgt wel data binnen, maar maakt nog geen keuzes. We gaan de NEAT-instellingen aanpassen om dit te verhelpen.

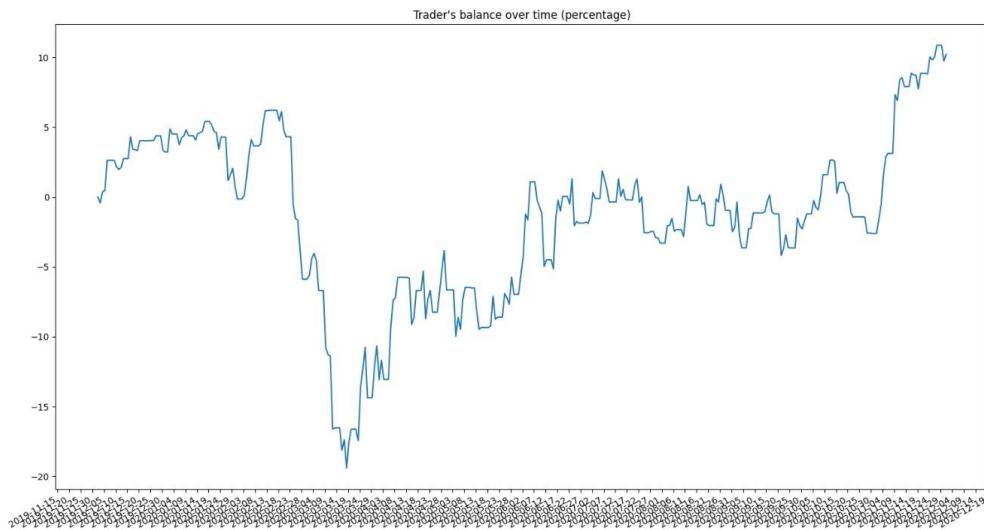
v1.1

De AI krijgt data binnen en kan deze interpreteren. Na vijf euro winst stopt de AI met investeren. We gaan in de NEAT-instellingen de *mutation rate* en de fitnessfunctie aanpassen om dit te verhelpen.

⁴⁴ In bovenstaand overzicht zijn de verschillende versies van de GUI niet opgenomen.

v2.0

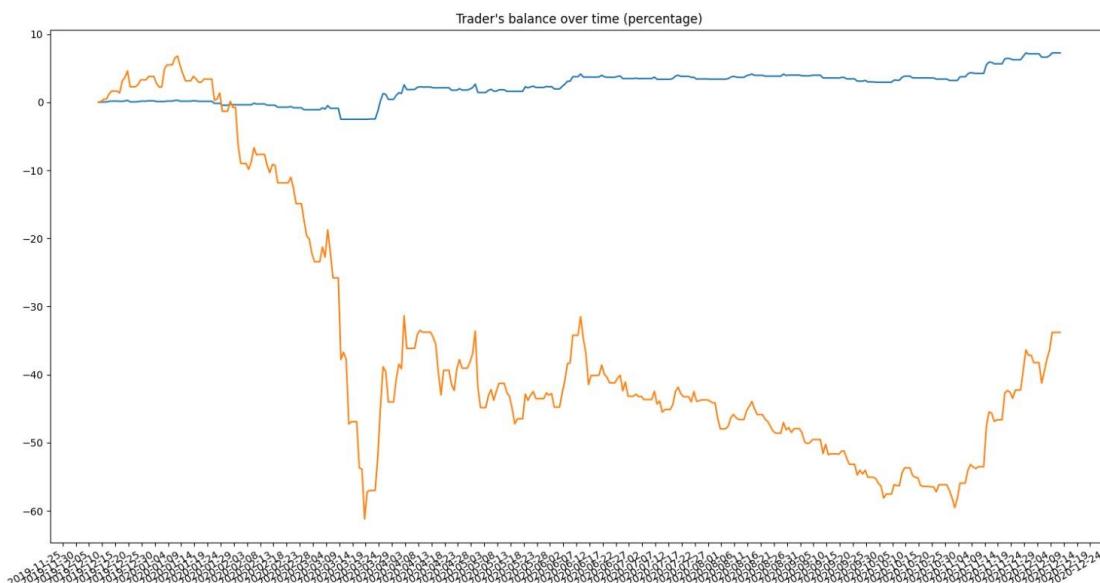
De AI maakt keuzes, maar laat zich teveel leiden door de koers van het aandeel. De grafieken van de winst van de AI en de koers van het aandeel zijn nagenoeg identiek.



Figuur 8.b: snapshot v2.0. De blauwe lijn is het resultaat van de AI en de oranje lijn is de koers van het geteste aandeel. De oranje lijn is niet zichtbaar, omdat de blauwe lijn er precies overheen past.

v3.0

Het programma werkt redelijk bij aandelen die bearish zijn, maar kan niet goed omgaan met aandelen die bullish zijn.



Figuur 8.c: snapshot v3.0. De blauwe lijn is het resultaat van de AI en de oranje lijn is de koers van het geteste aandeel. Het geteste aandeel is bearish, en daar gaat de AI goed mee om.

v4.0

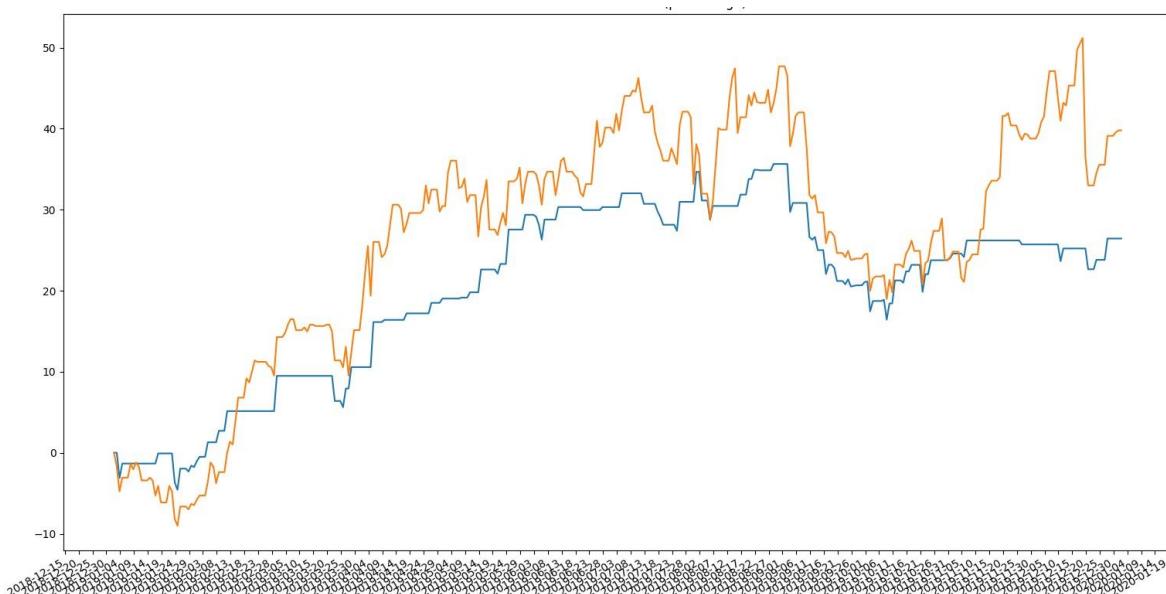
We hebben de AI inmiddels geleerd niet meer de koers van het aandeel te volgen. Het programma neemt weinig risico en maakt geringe winst.

v4.1

De AI volgt weer exact de koers van het aandeel.

v5.0

We hebben de AI agressiever getraind. Een compromis tussen alle instelbare parameters lijkt te zijn gevonden.



Figuur 8.d: De AI (blauw) is winstgevend en volgt weliswaar redelijk de koers van het geteste aandeel (oranje), maar maakt zichtbaar zijn eigen beslissingen, omdat de koersen niet precies overlappen.

v5.1

Onze AI kan nu ook een realtime beleggingsadvies genereren.

8.2 Resultaten

We testen verschillende aandelen over verschillende tijdsperiodes. Allereerst bekijken we 9 verschillende *tickers*⁴⁵ van 1 januari 2019 tot 1 januari 2020. Hiermee simuleren we de markt in een normale situatie, dus voor corona. Om onze AI in zoveel mogelijk situaties te testen, geven we

⁴⁵ Mocht hoofdstuk 7 zijn overgeslagen: een *ticker* is een verhandelbaar product op de beurs. Bijvoorbeeld een aandeel, index of optie. Het EUR/USD-pair is ook een ticker.

het programma 2 stabiele tickers, 1 bullish aandeel, 1 bearish aandeel en 2 tickers met een hoge volatiliteit als input.

We bespreken daarnaast ook de periode van 1 januari 2020 tot 10 december 2020. De coronacrisis zorgde voor een onrustige beurs. We zijn benieuwd hoe ons programma daarmee omgaat. In deze periode geven we ons programma daarom 2 stabiele tickers, 2 bullish aandelen, 2 bearish aandelen en 2 tickers met een hoge volatiliteit.

Welke tickers we hebben gekozen, leggen we later in dit hoofdstuk uit. We geven dan ook onze motivatie voor de tickerkeuzes. De grafieken zullen we steeds geven in percentages. De rode lijn is de koers die de desbetreffende ticker over de bekeken periode liet zien en de blauwe lijn is de procentuele winst die de AI door te investeren in dat aandeel heeft gemaakt.

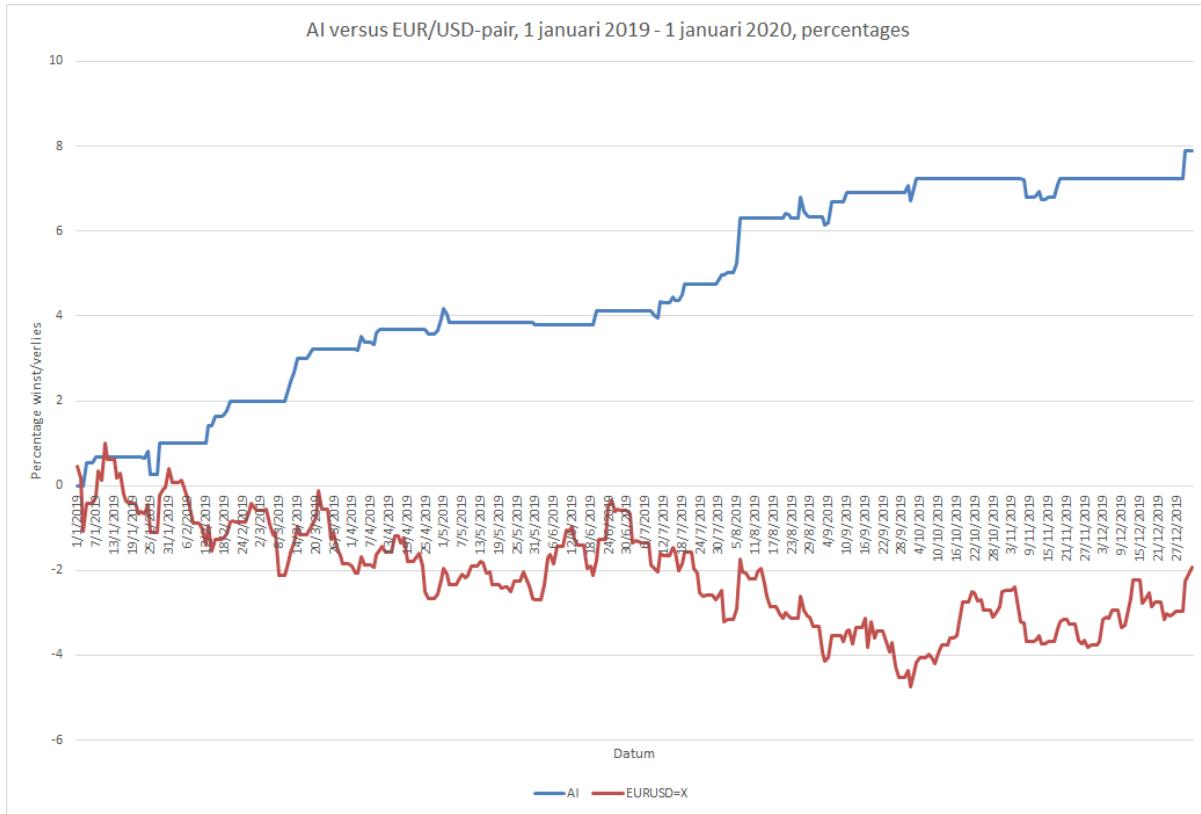
8.2.1 1 januari 2019 - 1 januari 2020 (voor corona)

We bespreken allereerst de tickers in een 'normale' situatie, dus voor corona. Onze tests verlopen allemaal volgens hetzelfde procedé:

- 1) We trainen de AI. Dit doen we met 14 tickers die we van tevoren willekeurig hebben gekozen. De 14 tickers zijn: KLMR, RYDAF, BP, GOOG, FSR, GOLD, TSLA, AMZN, SI=F, CL=F, XRP-USD, BCH-USD, LTC-USD en MSFT;
- 2) We slaan de virtuele belegger met de hoogste fitness op. De virtuele belegger traint dus niet meer 'door' met de waardes die hij in de stappen hierna van ons krijgt;
- 3) We laden de data in van de periode en de ticker die we gaan bekijken;
- 4) We kennen een saldo van 5000 euro aan de virtuele belegger;
- 5) We geven de virtuele belegger op dag x als input het koersverschil met de vorige dag, 2 dagen geleden, 5 dagen geleden, 2 weken geleden en 31 dagen geleden;
- 6) De virtuele belegger neemt positie in (niets, voor 100 euro kopen of voor 500 euro kopen);
- 7) De slotkoers van de desbetreffende dag wordt opgehaald;
- 8) De virtuele belegger sluit alle posities tegen de slotkoers;
- 9) Een nieuwe dag wordt gestart ('computernotatie': $x = x + 1$) en stappen 5 tot en met 8 worden herhaald tot het einde van de periode die we bekijken is bereikt;
- 10) We exporteren de gegevens naar Excel voor verdere verwerking.

8.2.1.1 EUR/USD-pair (EURUSD=X), super stabiel

In hoofdstuk 1.7 bespraken we dat forextickers zeer stabiel zijn. Dat blijkt ook uit de onderstaande grafiek. Het EUR/USD-pair in de periode van 1 januari 2019 - 1 januari 2020 is met ongeveer 2% gedaald. Onze AI zet dit met behulp van slimme trades om in een winst van **7,88%**.



Grafiek 8.e: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

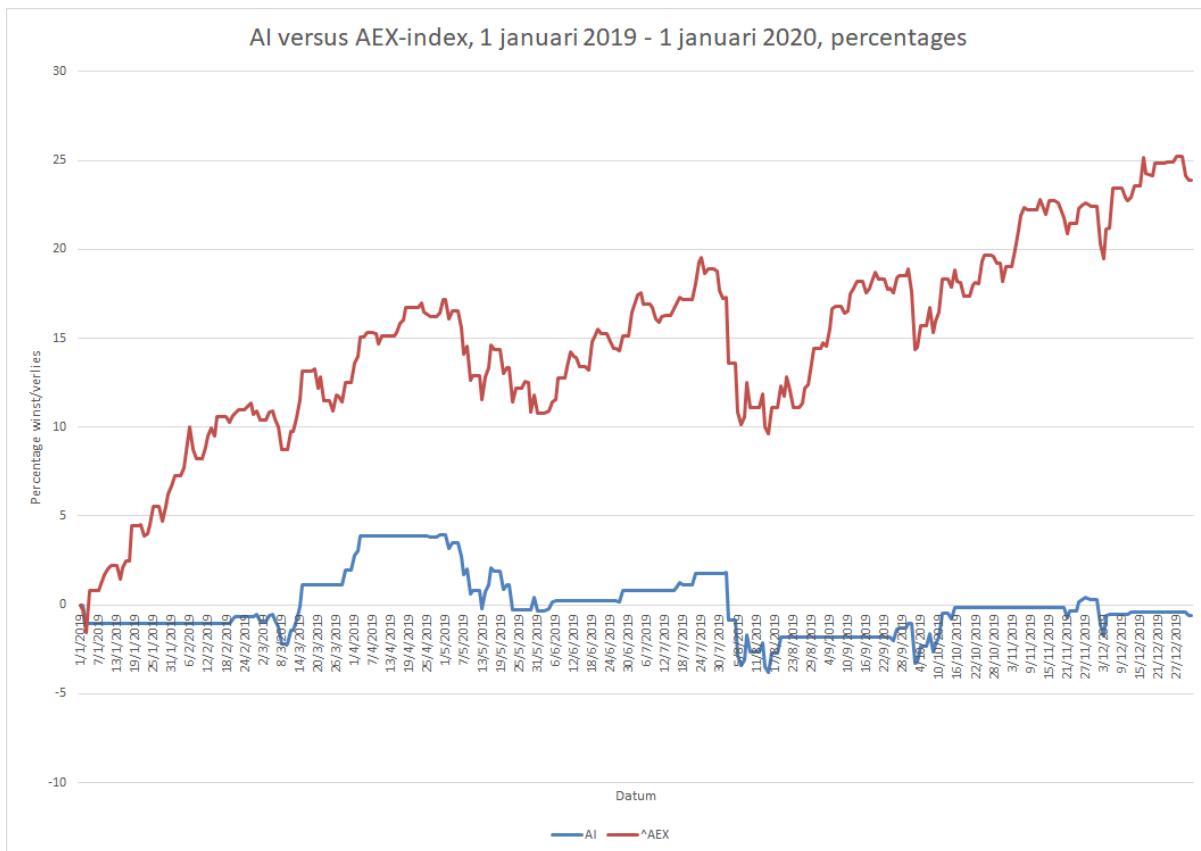
Opvallend zijn de rechte stukken. We hebben twee verklaringen gevonden voor het feit dat de AI op veel dagen niets doet:

- De AI is toch in lokaal minimum beland en komt daar lastig uit;
- De koers daalde op dat moment, en omdat de AI niet de mogelijkheid heeft om aandelen te shorten, koos de AI voor niets doen.

Een combinatie van a) en b) is natuurlijk ook mogelijk.

8.2.1.2 AEX-index (^AEX), vrij stabiel

We kijken ook naar de AEX-index. De AEX-index steeg in 2019 met ongeveer 24%. Onze AI weet deze stijging echter niet te benutten, en sluit het jaar af met een negatief rendement: **-0,59%**.



Grafiek 8.f: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

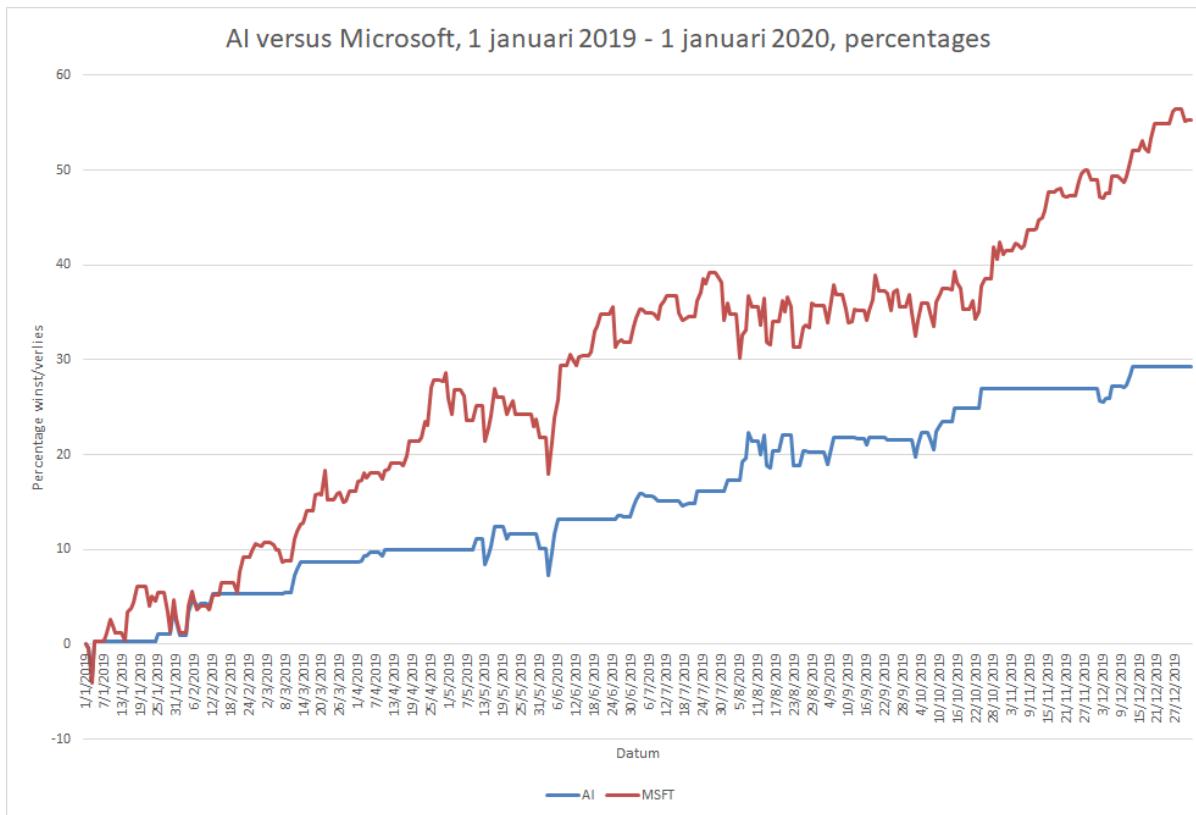
De vraag waarom de AI hier de mist in lijkt te gaan, hield ons een tijdje bezig. Uiteindelijk wijten we dit aan een combinatie van de volgende factoren:

- Ons programma begreep het patroon niet;
- De stijgende trends waren van relatief korte duur, waardoor het programma slecht actie kon ondernemen;
- De AI raakte vaak vast in een lokaal minimum;
- Onze instellingen waren niet geoptimaliseerd voor een opwaartse trend met relatief diepe dalen.

We zouden onze AI kunnen hertrainen, zodat hij wel een positief rendement zou hebben voor de AEX-index. Dit zien wij als valsspelen, omdat we dan eigenlijk handelen met voorkennis. Immers, als de AI in realtime zou worden ingezet, heeft de gebruiker ook geen mogelijkheid om terug in de tijd te gaan en het algoritme aan te passen. Daarom hebben we ervoor gekozen de AI niet opnieuw te trainen.

8.2.1.3 Microsoft (MSFT), bullish

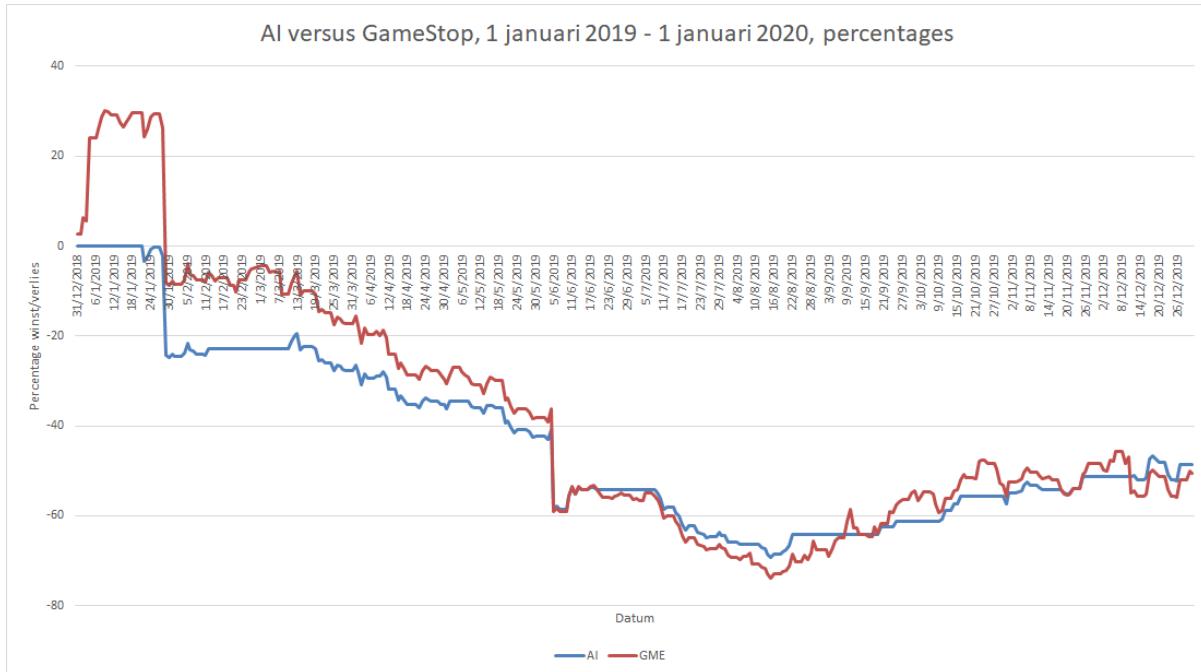
Techbedrijf Microsoft deed goede zaken in 2019. De waarde van een aandeel steeg met ongeveer 55%. Bij deze grafiek zien we voor het eerst dat onze AI anticipeert op de koers. Ons programma lijkt weinig risico te nemen, maar neemt wel degelijk cruciale beslissingen. De beslissingen pakken goed uit en het rendement bevestigt dit: **29,26%**.



Grafiek 8.g: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.1.4 GameStop (GME), bearish

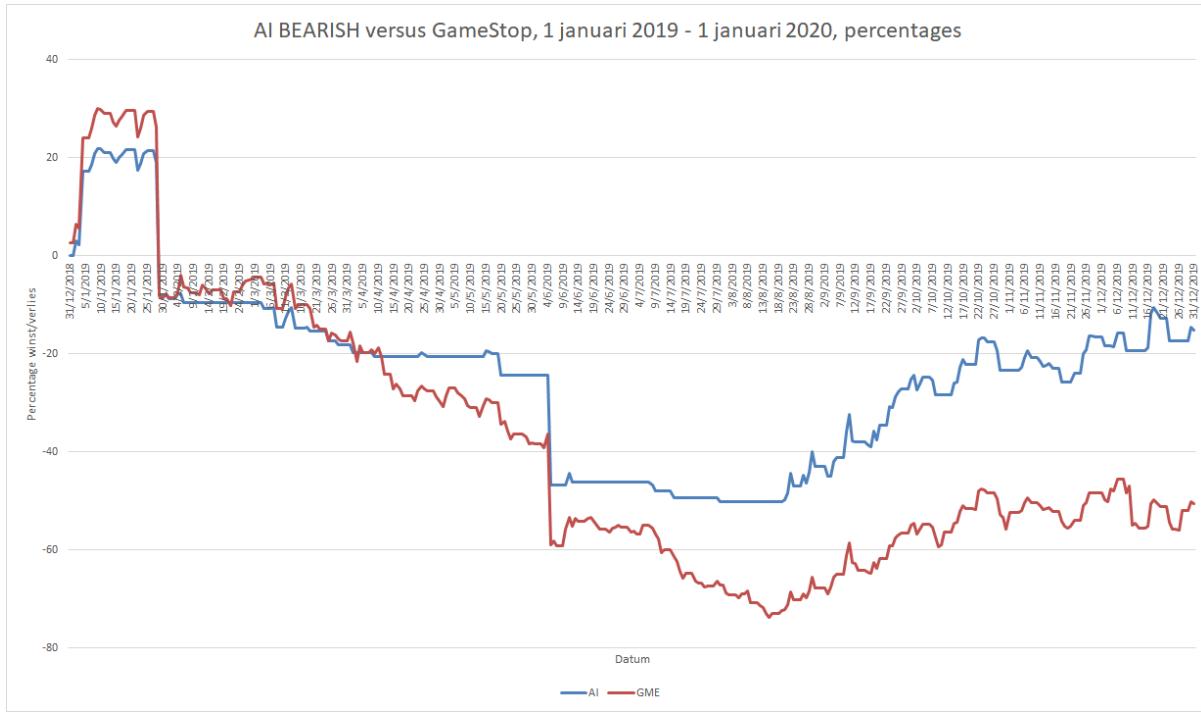
Het is lastig om een aandeel te vinden dat in 2019 een beartrend laat zien. 2019 was een goed jaar voor de beurs en er waren weinig aandelen met een negatief rendement. Uiteindelijk hebben we gekozen voor GameStop, een Amerikaanse retailer van computerspellen en consumentenelektronica. Het aandeel GameStop daalde in 2019 met 50%. Onze AI kan hier slecht mee omgaan: **-48,62%**.



Grafiek 8.h: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

Zelfs de speciale bearishversie⁴⁶ van ons programma weet geen positief rendement te behalen met het aandeel GameStop. Uiteindelijk eindigt het jaar voor de AI met een rendement van **-15,19%**. In de discussie (hoofdstuk 8.4) bespreken we waarom wij denken dat ons programma moeite had met dit aandeel.

⁴⁶ We vertellen later (hoofdstuk 8.2.2.5) wat dit precies is en wat de aanleiding voor een tweede AI was.

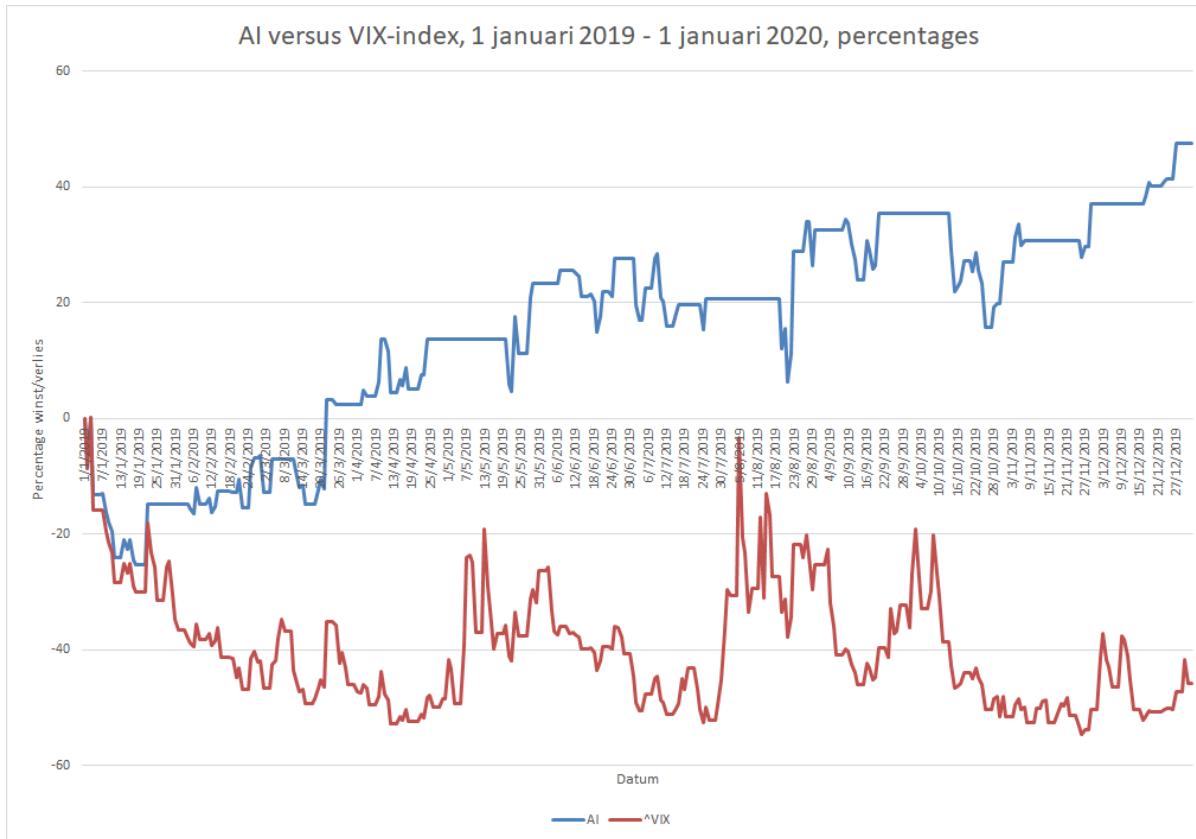


Grafiek 8.i: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.1.5 VIX-index (^VIX), hoge volatiliteit

De volatiliteitsindex van de S&P-500, de VIX (zie ook hoofdstuk 1.6) vertelt ons niet alleen wat over de volatiliteit van een aantal aandelen, maar is zelf ook behoorlijk volatiel. Dit komt omdat de exacte volatiliteit van de S&P500 sterk wisselt. De koersgrafiek vertoont veel pieken en dalen, en het is lastig te voorspellen hoe de koers zich in de toekomst zal ontwikkelen. Beleggers mijden dit soort zeer volatiele tickers meestal (want weinig zekerheid), maar voor traders zijn volatiele tickers een paradijs (want veel mogelijkheden om winst te maken). We zijn benieuwd hoe onze AI omgaat met deze index.

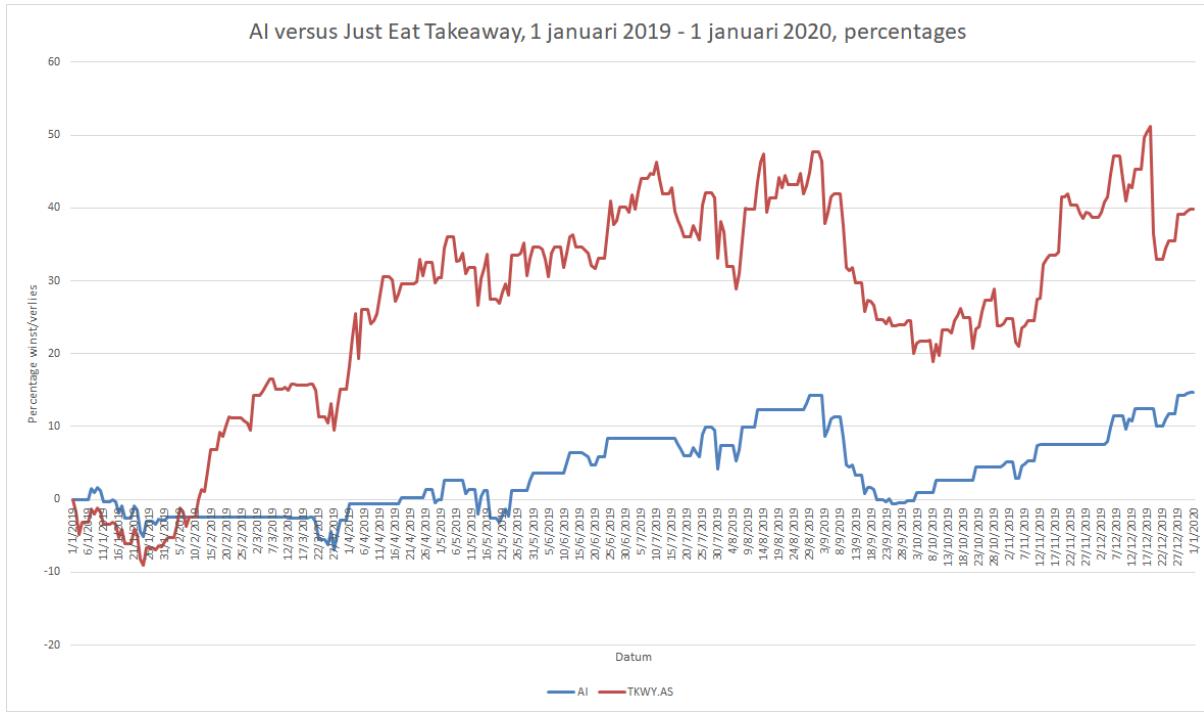
De VIX-index daalde met ongeveer 46% in 2019. Ons programma weet de volatiliteit van de index echter uitstekend te benutten en behaalt een rendement van maar liefst **47,50%**.



Grafiek 8.j: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.1.6 Just Eat Takeaway (TKWY.AS), hoge volatiliteit

Just Eat Takeaway, het moederbedrijf van onder andere thuisbezorgd.nl, heeft aandelen die behoorlijk in waarde schommelen. Omdat de AI goed presteerde bij de VIX, willen we hem ook testen bij een ander volatiel aandeel. De koers van Just Eat Takeaway stijgt, in tegenstelling tot de VIX, wel, met zo'n 40%. Ons programma weet ook hier de volatiliteit prima te benutten, met als resultaat een jaarrendement van **14,68%**.



Grafiek 8.k: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.2 1 januari 2020 tot 10 december 2020 (met corona)

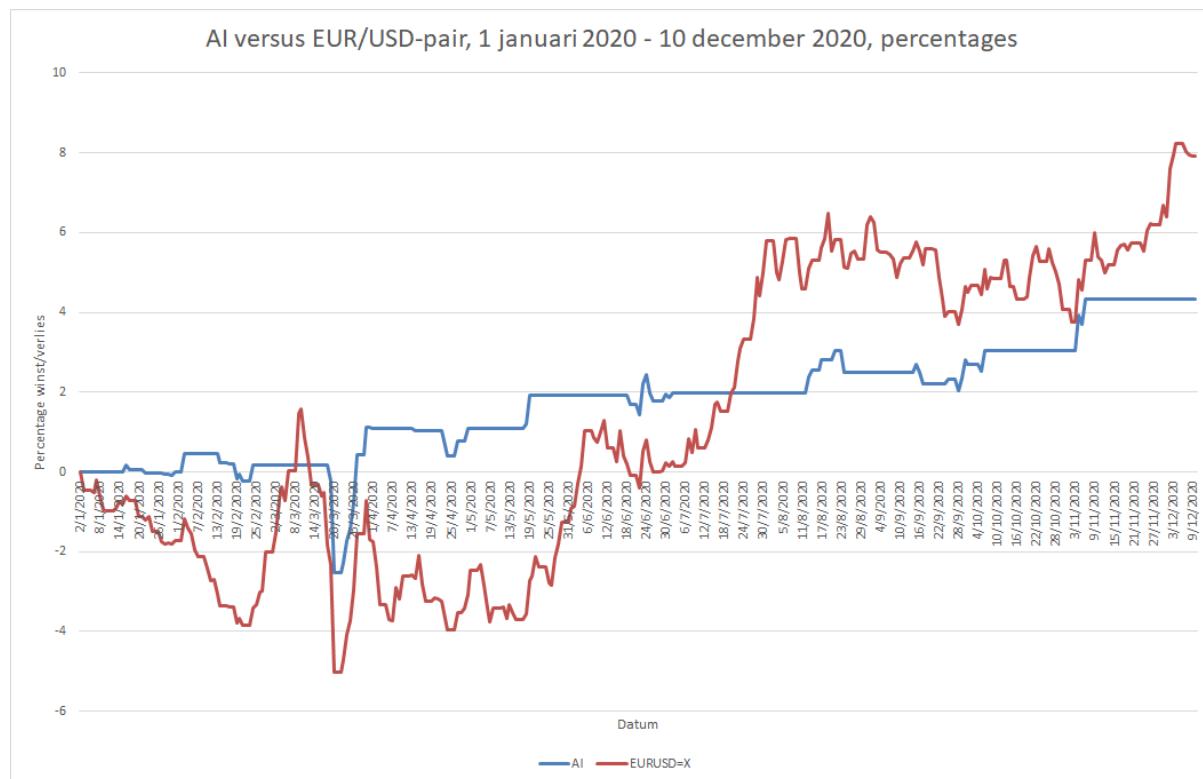
Het jaar 2020 was een rommelig beursjaar dankzij de coronacrisis. We zijn benieuwd hoe ons programma hiermee omgaat.

Ook nu verlopen onze tests volgens hetzelfde procedé:

- 1) We trainen de AI. Dit doen we met 14 tickers die we van tevoren willekeurig hebben gekozen. De 14 tickers zijn: KLMR, RYDAF, BP, GOOG, FSR, GOLD, TSLA, AMZN, SI=F, CL=F, XRP-USD, BCH-USD, LTC-USD en MSFT;
- 2) We slaan de virtuele belegger met de hoogste fitness op. De virtuele belegger traint dus niet meer 'door' met de waardes die hij in de stappen hierna van ons krijgt;
- 3) We laden de data in van de periode en de ticker die we gaan bekijken;
- 4) We kennen een saldo van 5000 euro aan de virtuele belegger;
- 5) We geven de virtuele belegger op dag x als input het koersverschil met de vorige dag, 2 dagen geleden, 5 dagen geleden, 2 weken geleden en 31 dagen geleden;
- 6) De virtuele belegger neemt positie in (niets, voor 100 euro kopen of voor 500 euro kopen);
- 7) De slotkoers van de desbetreffende dag wordt opgehaald;
- 8) De virtuele belegger sluit alle posities tegen de slotkoers;
- 9) Een nieuwe dag wordt gestart ('computernotatie': $x = x + 1$) en stappen 5 tot en met 8 worden herhaald tot het einde van de periode die we bekijken is bereikt;
- 10) We exporteren de gegevens naar Excel voor verdere verwerking.

8.2.2.1 EUR/USD-pair (EURUSD=X), super stabiel

Ook in deze periode testen we als eerst het EUR/USD-pair. Zelfs met de coronacrisis is deze ticker relatief stabiel en dat maakt de ticker een geschikte nulmeting. Het EUR/USD-pair stijgt in de periode 1 januari 2020 tot 10 december 2020 met ongeveer 8%. Onze AI laat zich niet beïnvloeden door de fluctuaties in de koers en dat levert een ons een jaarrendement van **4,33%** op.

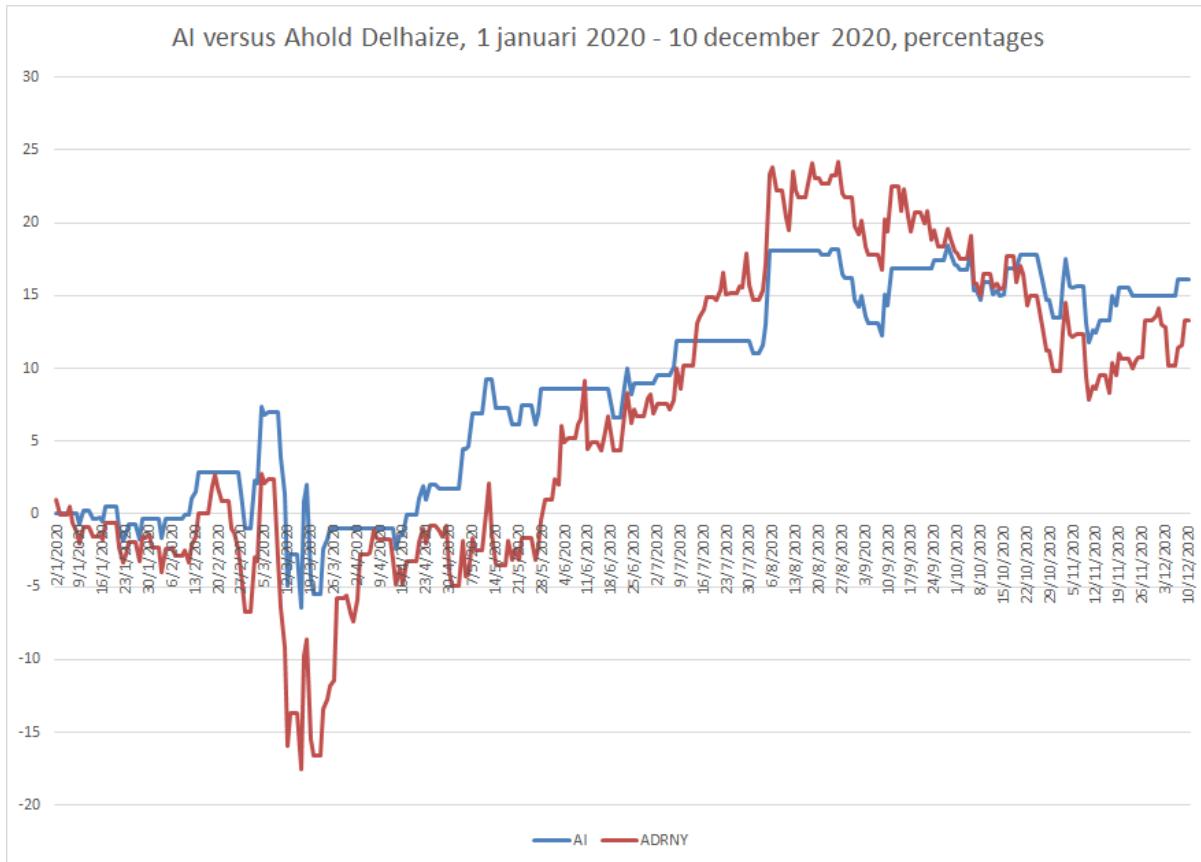


Grafiek 8.1. rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.2.2 Ahold Delhaize (ADRNY), vrij stabiel

Als we een andere stabiele ticker door ons programma willen laten analyseren, kunnen we in de periode 1 januari 2020 - 10 december 2020 beter niet de AEX-index kiezen. Die vertoont namelijk veel schommelingen. Een goed alternatief zijn volgens ons de aandelen van Ahold Delhaize. Die firma klinkt onbekend, maar Ahold Delhaize is het moederbedrijf van onder andere Albert Heijn. De vraag naar levensmiddelen blijft, crisis of niet, relatief constant en daarom is het aandeel Ahold Delhaize zelfs tijdens de coronacrisis vrij stabiel.

Het aandeel Ahold Delhaize steeg dit jaar met iets meer dan 13%. De AI volgt de koersbewegingen meer dan we tot nu toe hebben gezien. Hier hebben we geen specifieke verklaring voor kunnen vinden. De AI heeft uiteindelijk een rendement van **16,10%**.

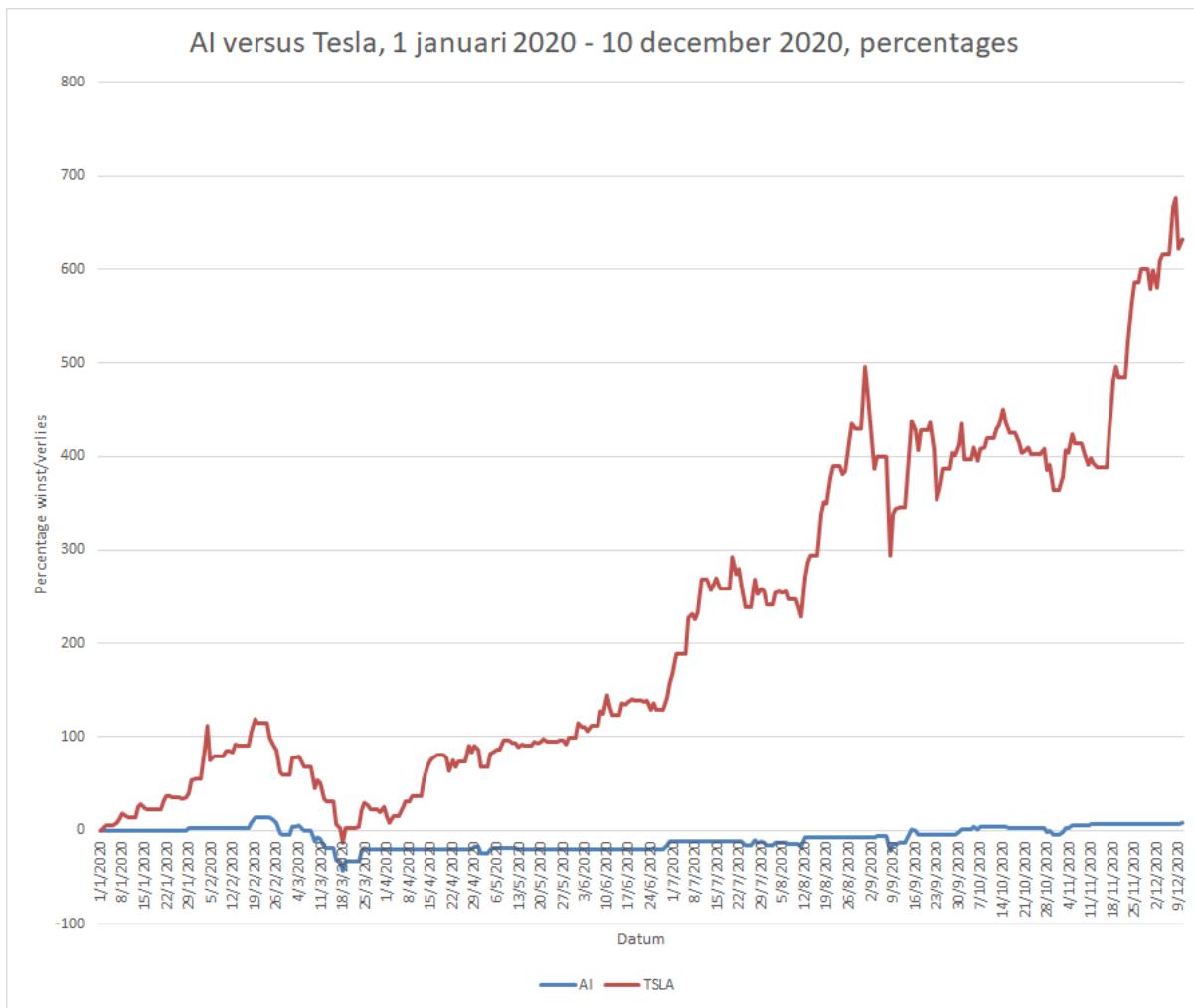


Grafiek 8.m: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.2.3 Tesla (TSLA), hevig bullish

In hoofdstuk 1.4 schreven we het aandeel Tesla de 'bubbelsatus' toe. Niet geheel onterecht, want het aandeel steeg dit jaar met maar liefst 630%. We hadden hoge verwachtingen van onze AI, want de koersgrafiek van Tesla heeft weinig grote dalen, dus ons programma kan op weinig plekken de mist ingaan.

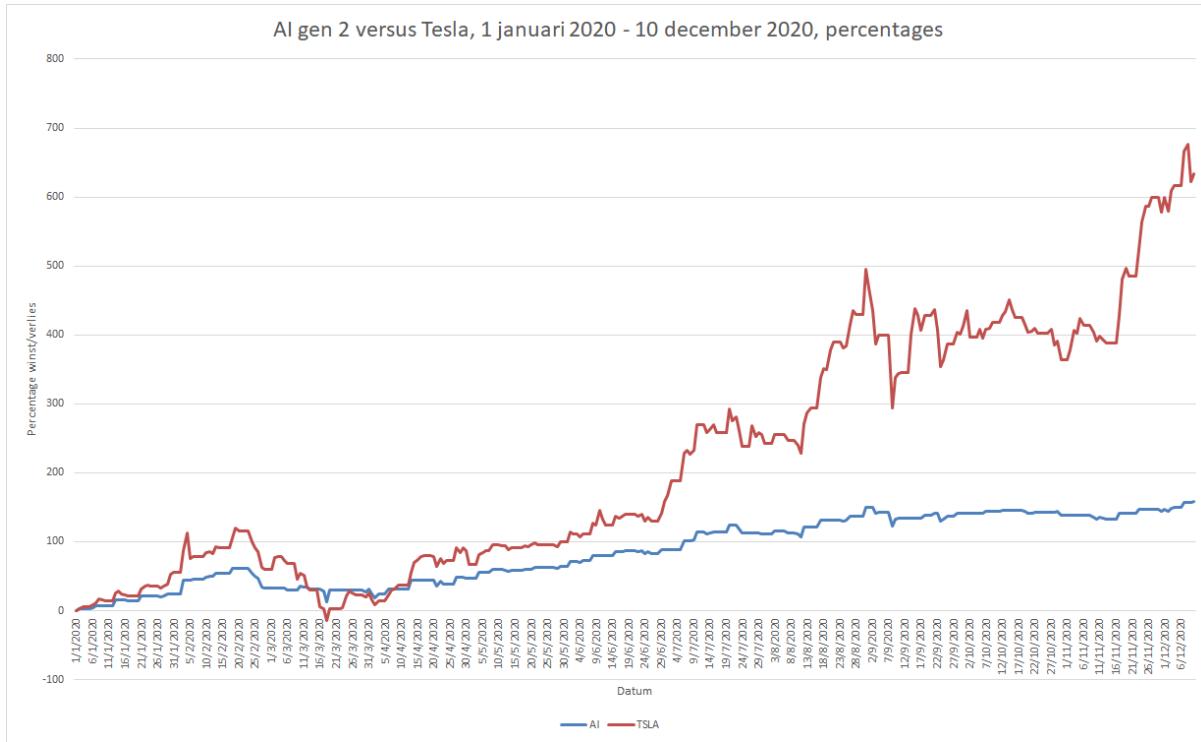
De uitkomst van deze test verraste ons. De AI heeft slechts **8,15%** winst gemaakt. Veel minder dan we hadden verwacht. En sterker nog, de AI heeft het grootste deel van het jaar gewerkt met een negatief rendement.



Afbeelding 8.n: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

Uiteindelijk hebben we een mogelijke verklaring voor dit verschijnsel gevonden. We vermoeden dat de AI zichzelf geleerd heeft dat kopen in een dal een goed idee is. Dit is niet gek en heeft zelfs een naam in de aandelenwereld: de *buy-low-sell-highstrategie*. Tesla heeft weinig ‘lows’ gehad en er waren dus weinig momenten waarop de AI aandelen wilde kopen. Waarschijnlijk is de winst van de AI daarom beperkt gebleven.

Vervolgens hebben we de AI geforceerd om de buy-low-sellhighstrategie niet meer te gebruiken. De tweede generatie AI kan beter omgaan met het sterk stijgende Tesla en behaalt een rendement van **158,71%**.



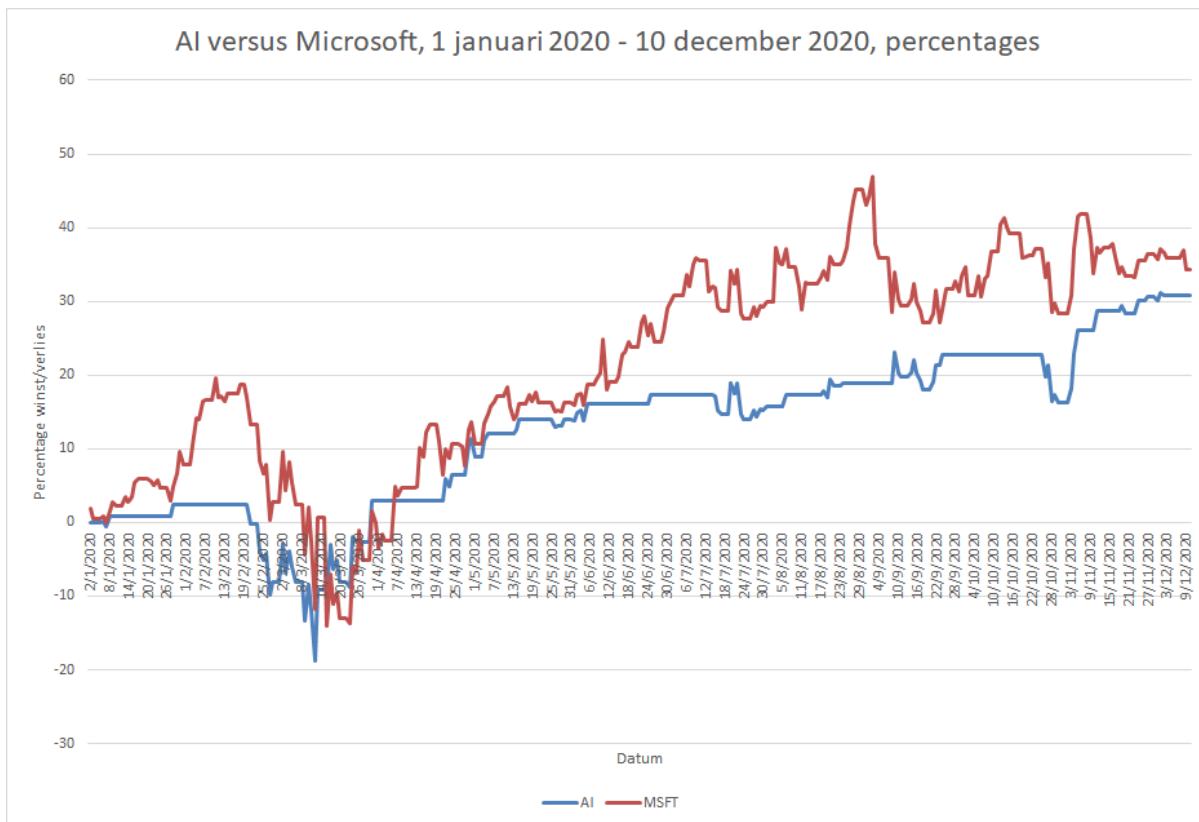
Grafiek 8.o: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

We hebben de tickers die we hierboven bespraken ook met de tweede generatie AI getest. Dit leverde een lager rendement op. Achteraf kunnen switchen van type AI is natuurlijk handelen met voorkennis, dus noteren we **8,15%** rendement per jaar als eindresultaat voor het aandeel Tesla.

8.2.2.4 Microsoft (MSFT), bullish

En ook in 2020 bekijken we het aandeel Microsoft. Microsoft kunnen we weer classificeren als bullish. Dit jaar was het rendement op aandelen in het computerbedrijf zo'n 34%.

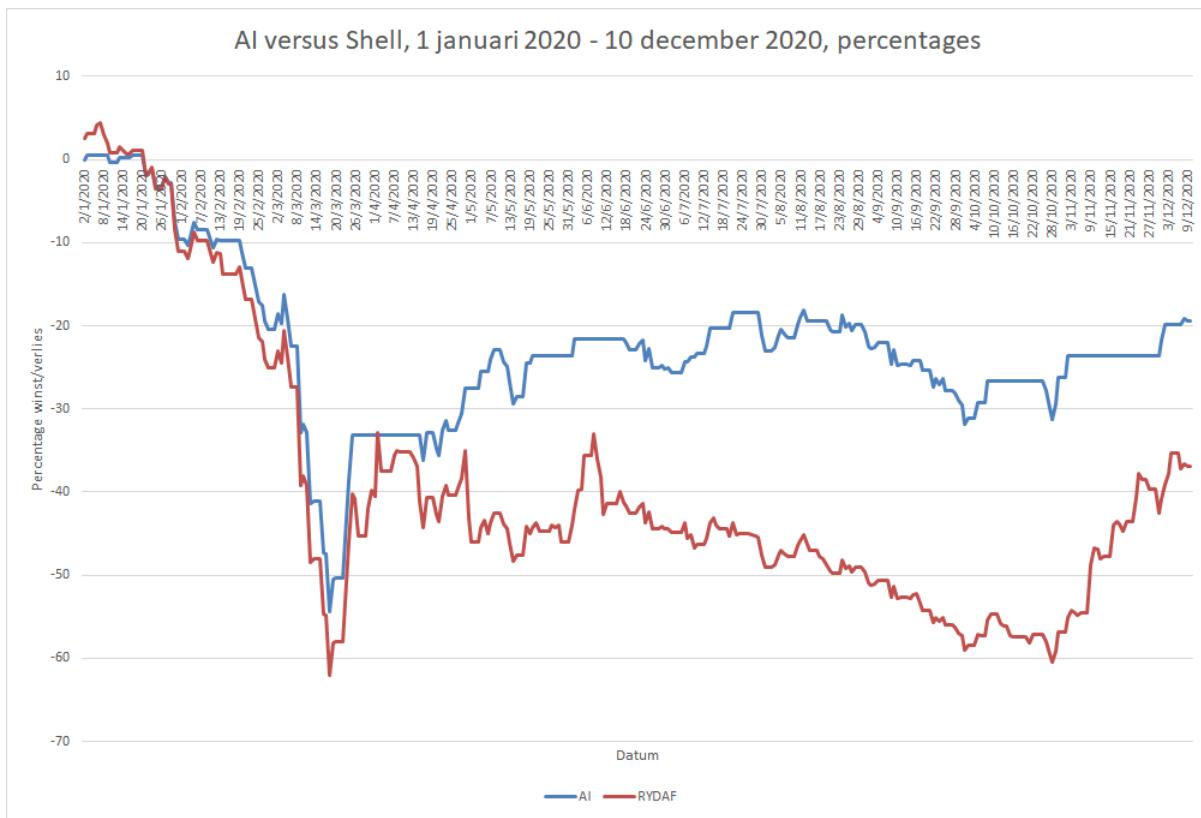
Vooral in de maanden februari en maart lijkt zowel Microsoft als onze AI het zwaar te hebben. Uiteindelijk breekt onze AI toch door de y-as en wordt het jaar afgesloten met een positief rendement: **30,84%**.



Grafiek 8.p: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

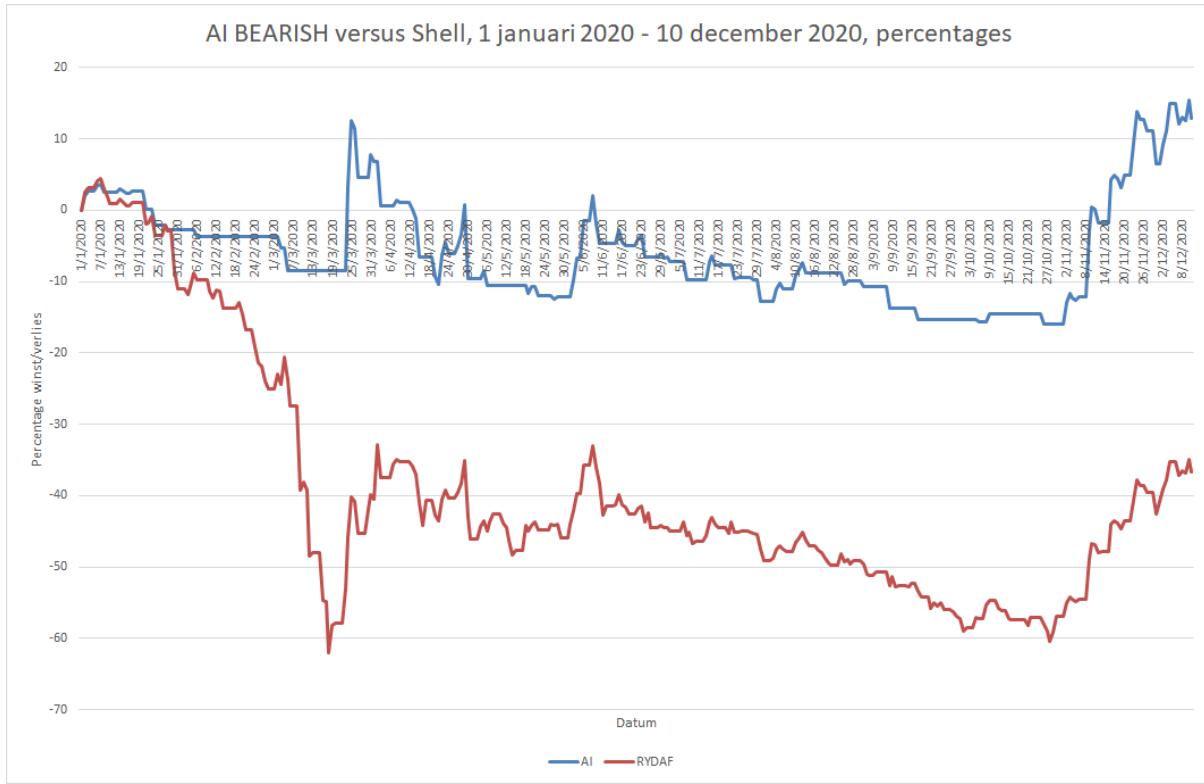
8.2.2.5 Shell (RYDAF), bearish

Het oliebedrijf Shell had het zwaar tijdens de coronacrisis. Het aandeel van Shell daalde dit jaar met ongeveer 37%. Onze AI heeft het zwaar. Hij behaalt weliswaar een hoger winstpercentage dan het aandeel zelf, maar met een rendement van **-19,45%** maakt ons programma absoluut geen winst.



Afbeelding 8.q: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

Deze tegenslag weerhield ons er niet van een oplossing te bedenken. Klaarblijkelijk werkt onze AI slecht met bearish aandelen. We passen de instellingen van NEAT aan totdat de AI ons een positief rendement geeft. Deze aangepaste instellingen noemen we de bearishversie van ons programma. De bearishversie geeft ons in dit geval wel een positief rendement: **12,91%**.

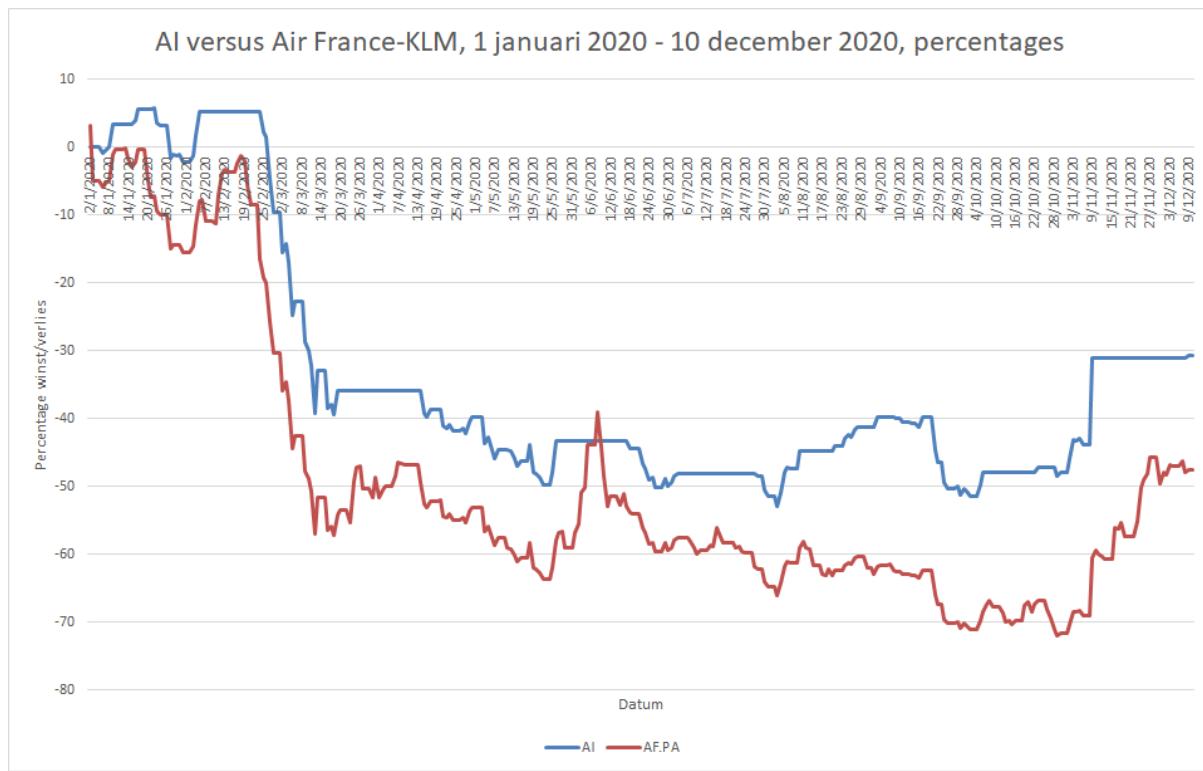


Afbeelding 8.r: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

Er zijn argumenten te noemen waarom de nieuwe bearishversie valt onder handelen met voorkennis. Wij vinden echter dat de bearishversie prima te verantwoorden is, omdat we, met behulp van indicatoren en trends (hoofdstuk 2.2), objectief kunnen vaststellen of een aandeel bearish of niet is. In de praktijk zou een trader (of een computerprogramma) eerst met bijvoorbeeld een SMA-indicator kunnen vaststellen welke trend een koers heeft, alvorens een versie van het programma te kiezen.

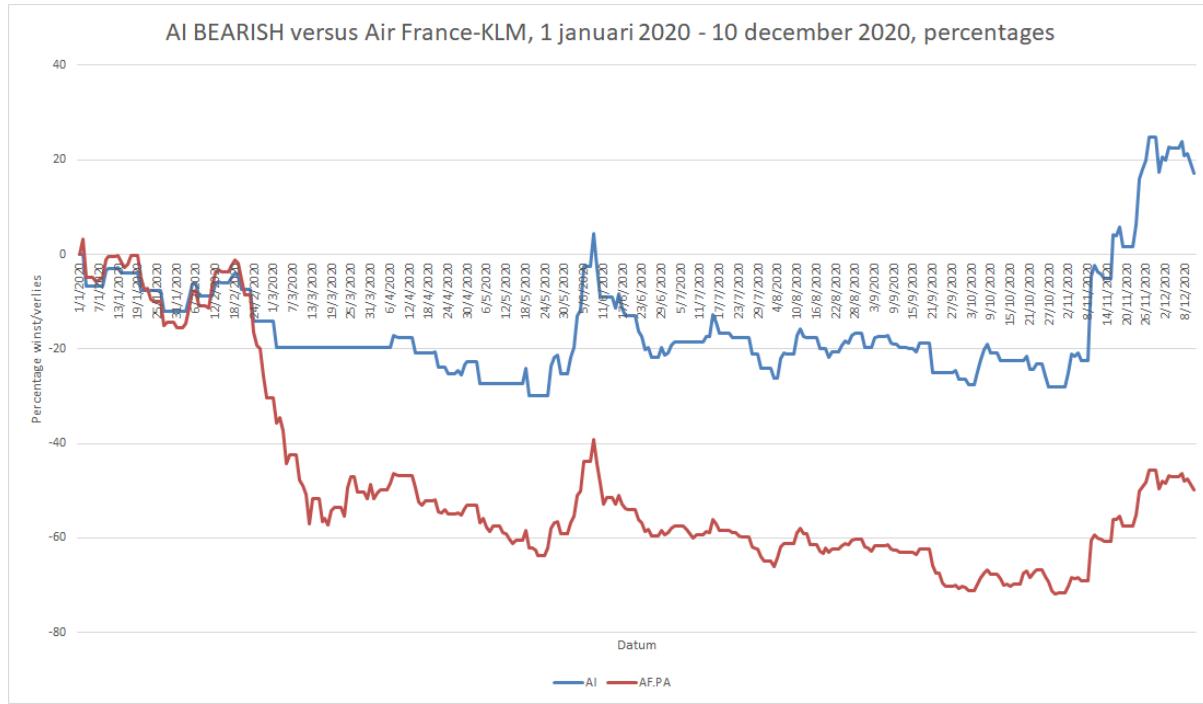
8.2.2.6 Air France-KLM (AF.PA), bearish

Air France-KLM is regelmatig in ons verslag behandeld. Het aandeel daalde dit jaar flink, zo'n 48%. Ook hier zien we dat de normale versie van de AI het moeilijk heeft. Na de sterke daling in maart breekt de AI de rest van het jaar niet meer door de y-as. Het eindresultaat is een negatief rendement: **-30,74%**.



Afbeelding 8.s: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

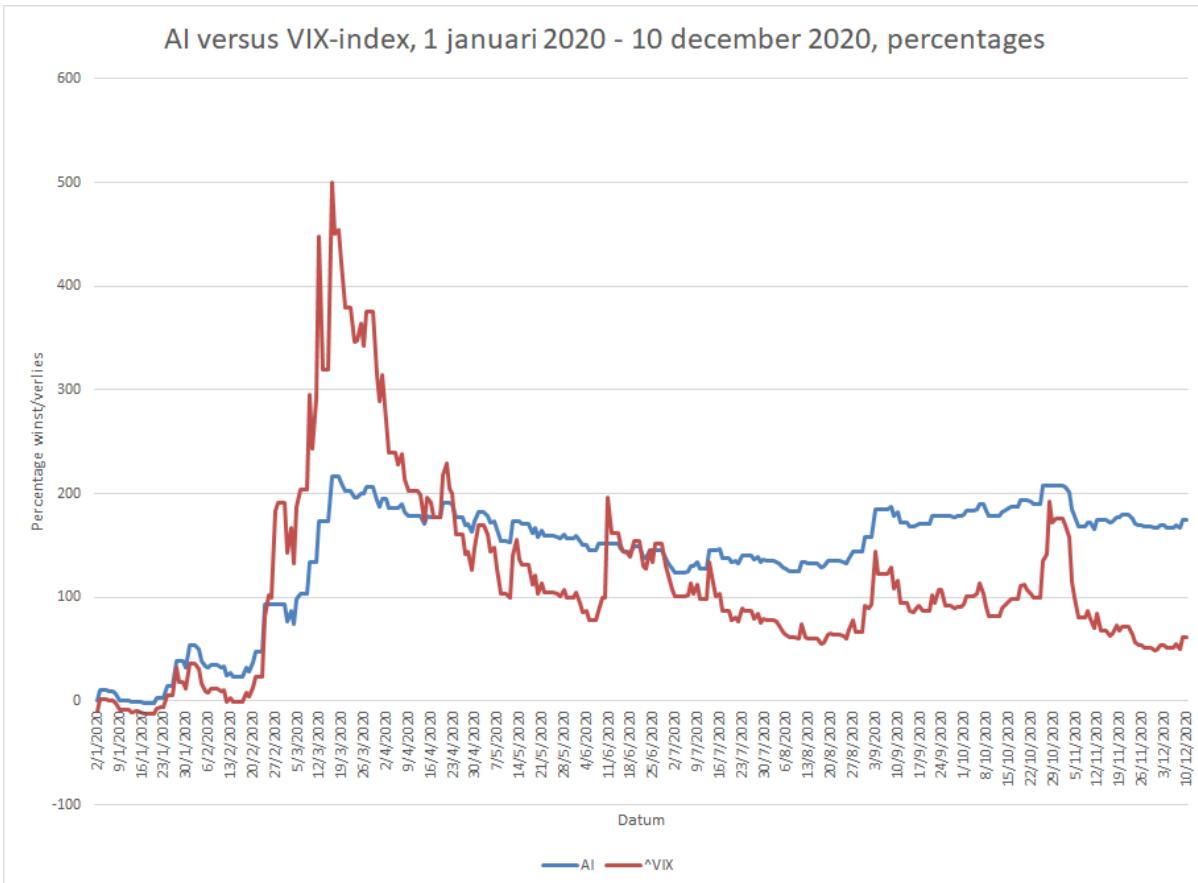
Ook op deze ticker hebben we onze bearishversie getest. Met succes, want het rendement van onze AI-bearishversie is hier **17,06%**.



Afbeelding 8.t: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.2.7 VIX-index (^VIX), hoge volatiliteit

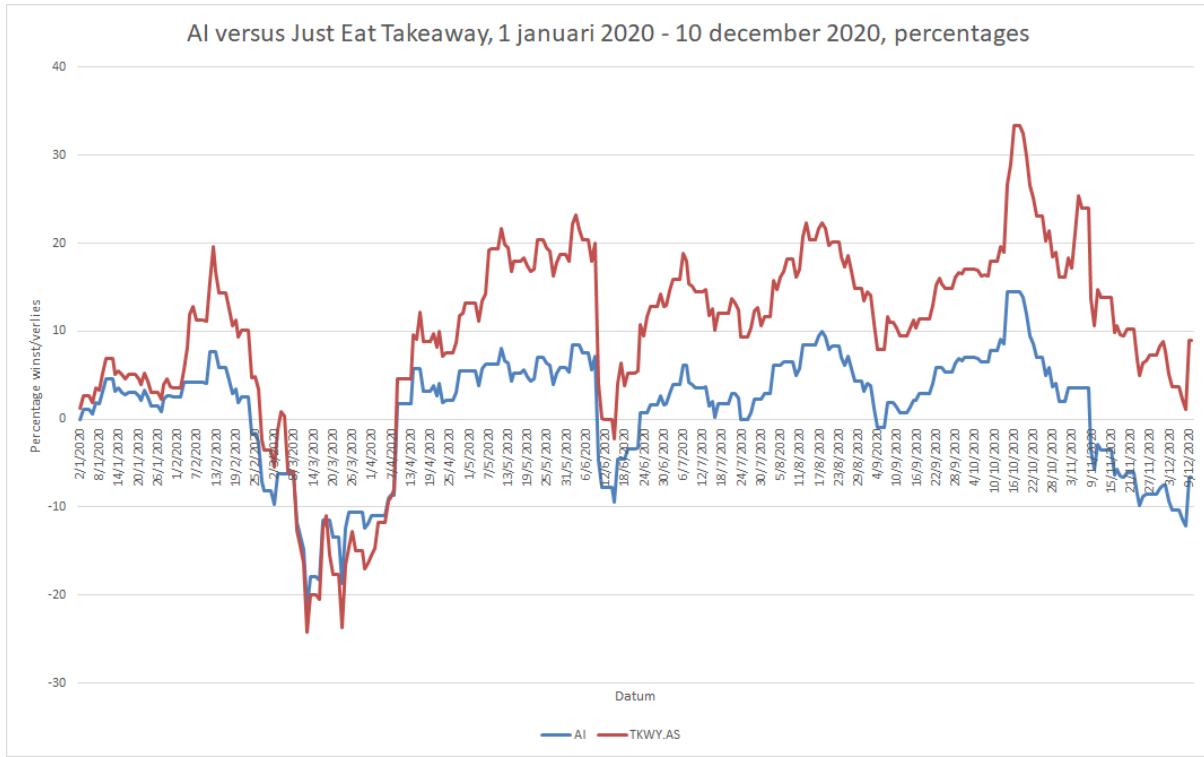
We kijken nogmaals naar de VIX, want ook in de periode 1 januari 2020 - 1 december 2020 ging deze behoorlijk op en neer. De VIX steeg dit jaar met 62%. Bij deze index stelt het programma ons niet teleur. De AI maakt gebruik van de hoge volatiliteit en dat levert een rendement van **174,78%** op.



Grafiek 8.u. rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.2.2.8 Just Eat Takeaway (TKWY.AS), hoge volatiliteit

En tot slot testen we onze AI nogmaals met de aandelen van Just Eat Takeaway. Ook in 2020 schommelde dit aandeel flink. Het jaarrendement van Just Eat Takeaway is ongeveer 9%. Onze AI laat zich flink beïnvloeden door de schommelende koers. Zonder succes, want het rendement duikt in de laatste maand onder de y-as: **-6,54%**.



Afbeelding 8.v: rendementsgrafiek van het desbetreffende aandeel (rood) en onze AI (blauw). Op de y-as is het winst-/verliespercentage uitgezet en op de x-as de datum.

8.3 Conclusie

Ons programma presteert aardig. Laten we alle rendementen even op een rijtje zetten:

1 januari 2019 - 1 januari 2020

Ticker	Rendement
EURUSD=X	+7,88%
^AEX	-0,59%
MSFT	+29,26%
GME	-15,19%
^VIX	+47,50%
TKWY.AS	+14,69%

De tabel voor 1 januari 2020 – 10 december 2020 staat op de volgende pagina.

1 januari 2020 - 10 december 2020

Ticker	Rendement
EURUSD=X	+4,33%
ADRNY	+16,10%
TSLA	+8,15%
MSFT	+30,84%
RYDAF	+12,91%
AF.PA	+17,06%
^VIX	+174,78%
TKWY.AS	-6,54%

Uit deze gegevens kunnen we concluderen dat ons programma bij aandelen met een hoge volatiliteit procentueel de meeste winst kan maken. Dit vinden wij niet heel gek, aangezien ons programma getraind is met drie cryptovaluta: Ripple (XRP-USD), Bitcoin Cash (BCH-USD) en Litecoin (LTC-USD). Cryptovaluta zijn anders dan ‘traditionele’ valuta nog niet helemaal ingeburgerd, en zijn daarom van nature vrij volatiel.

We kunnen ook de conclusie trekken dat onze AI beter presteert bij bullish aandelen dan bij bearish aandelen. Dit heeft waarschijnlijk te maken met de buy-low-sell-highstrategie (zie hoofdstuk 8.2.2.3) die de AI zichzelf heeft aangeleerd, maar kan ook heel goed te wijten zijn aan de trainingsdata die we ons programma hebben gegeven. Alleen Shell (RYDAF) liet in onze trainingsset een bearish koersverloop zien⁴⁷. We denken daarom dat we de AI onvoldoende hebben ‘voorbereid’ op dalende koersen.

Een opvallend resultaat was de tegenvallende winst bij het aandeel Tesla. Opvallend omdat het aandeel Tesla (TSLA) in onze trainingsdataset zat, en tegenwoordig omdat het aandeel steeg met 630% en onze AI slechts 8,15% winst gemaakt had. We wijten dit aan de buy-low-sell-highstrategie.

En tot slot moeten we het nog even hebben over de negatieve rendementen bij de AEX-index (^AEX), GameStop (GME) en Just Eat Takeaway (TKWY.AS). Hier gaven we in hoofdstuk 8.2.1.2 al een korte verklaring voor:

- 1) Ons programma begreep de aanwezige patronen niet;
- 2) De stijgende trends waren van relatief korte duur, waardoor het programma slecht actie kon ondernemen;
- 3) De AI raakte vaak vast in een lokaal minimum;
- 4) Onze instellingen waren niet geoptimaliseerd.

Aan dit rijtje willen we ook nog toevoegen dat toeval en geluk een rol gespeeld kunnen hebben. Waarom dit bij NEAT een rol speelt, bespraken we eerder in hoofdstuk 5.

⁴⁷ Dit komt omdat er statistisch gezien minder aandelen zijn met een bearish koersverloop. In onze testdataset hebben we ervoor gezorgd dat het aantal bearish aandelen overeenkwam met het aantal bullish aandelen, maar onze trainingsdataset was meer gebaseerd op ‘de echte wereld’.

8.3.1 De hamvraag: is ons programma beter dan een...

Dan is het nu tijd voor de hamvraag. Presteert ons programma beter dan andere spelers op de aandelenmarkt? Om dat te bepalen, nemen we het gemiddelde van de behaalde rendementen⁴⁸.

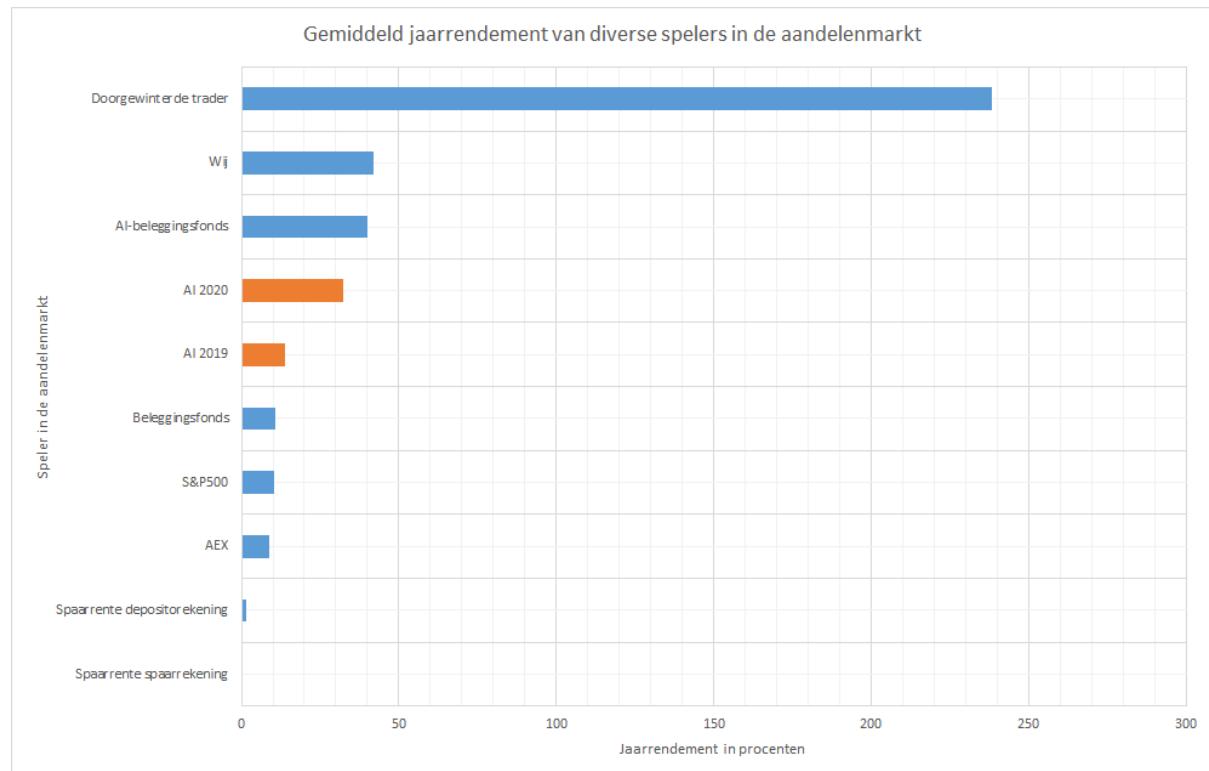
Gemiddeld rendement AI in 2019:

$$\frac{7,88 - 0,59 + 29,26 - 15,19 + 47,50 + 14,69}{6} \approx 13,93\% \quad (8.1)$$

Gemiddeld rendement AI in 2020:

$$\frac{4,33 + 16,10 + 8,15 + 30,84 + 12,91 + 17,06 + 174,78 - 6,54}{8} \approx 32,20\% \quad (8.1)$$

Als we deze waardes in ons staafdiagram, dat we presenteerden in de inleiding, zetten, levert dat de volgende einduitslag op:



Afbeelding 8.w: gemiddeld jaarrendement van diverse spelers in de aandelenmarkt.

⁴⁸ Dit gemiddelde is natuurlijk alleen accuraat als de gebruiker precies dezelfde aandelen had gekozen als wij. Wij denken dat de door ons gekozen aandelen een representatieve weergave zijn van de markt: minimaal één valuta, één index, één *techaandeel* en één *retailaandeel*. Het is best aannemelijk dat iemand in de praktijk een soortgelijke verdeling van aandelen zou kiezen, omdat beleggers en traders in hun portefeuille bij voorkeur meerdere sectoren willen hebben.

De spaarrente op een spaar- en depositorekening wordt door ons programma dus met gemak verslagen. Ook indexbeleggingen leveren gemiddeld een lager rendement op dan onze AI. Het traditionele beleggingsfonds presteert minder goed dan onze uitvinding, maar het AI-beleggingsfonds NextAlpha heeft een hoger rendement. Stach (de ‘wij’ in het staafdiagram) kan klaarblijkelijk beter beleggen dan ons computerprogramma en tegen een doorgewinterde trader kan onze AI al helemaal niet op.

Toch denken we dat het mogelijk is om onze AI aan te passen zodat ook het AI-beleggingsfonds en Stach worden verslagen. We bespreken dit in hoofdstuk 8.5.

8.3.2 Maatschappelijke impact

In de inleiding beloofden we plechtig dat we met ons programma de mensheid niet zouden uitroeien. Volgens ons hebben we die belofte niet gebroken, maar we vermoeden dat AI in de toekomst een grote rol kan gaan spelen op de beurs. Als drie enthousiaste jongens en een computer een redelijk rendement kunnen behalen, wat zegt dat dan over de mogelijkheden die een firma met duizend techneuten heeft? Dat zou zomaar het einde van de menselijke traders kunnen betekenen, maar gelukkig niet het einde van de mensheid.

8.4 Discussie

In de discussie vergelijken we eerst onze resultaten met de door ons opgestelde hypothese. Vervolgens bespreken we de validiteit en de betrouwbaarheid van ons onderzoek.

8.4.1 Terugblik op de hypothese

In de inleiding stelden we dat we ‘geen torenhoge rendementen verwachtten, maar hoopten om een rendement van 20-30% te behalen’. Met gemiddelde rendementen van 13,93% en 32,20% vinden we dat we aardig in de buurt van onze hypothese zitten. Er is meer onderzoek nodig om definitief vast te stellen of handelen met een AI een goed idee is, maar daarover praten we in hoofdstuk 8.5.

8.4.2 Validiteit en betrouwbaarheid van ons onderzoek

We kunnen ons afvragen of ons onderzoek betrouwbaar en valide is. We hebben 14 verschillende tickers met ons programma getest en bij 11 tickers een positief rendement gekregen. Dit maakt ons onderzoek redelijk betrouwbaar. We hebben wel slechts data van twee beursjaren gebruikt. Twee beursjaren zijn volgens ons niet genoeg om te stellen dat ons product in alle situaties even goed werkt. Dat maakt ons onderzoek minder betrouwbaar. Als we meer tijd en computerkracht tot onze beschikking hadden, zouden we dit probleem kunnen aanpakken.

Dan de validiteit. Kunnen we op basis van de gevonden resultaten harde conclusies trekken? Hier zijn voor- en tegenargumenten voor. Het belangrijkste tegenargument is dat Shell en Tesla zowel in onze trainingsdataset als in onze testdataset zaten. We hebben dus eigenlijk getest op iets wat de AI exact had kunnen leren. Dit argument kan echter gemakkelijk worden ontkracht met twee tegenargumenten:

- a) De uiteindelijke ‘beste belegger’ is gebaseerd op 14 verschillende tickers. Als ieder aandeel in de trainingsdataset even zwaar weegt, tellen de aandelen Tesla en Shell dus maximaal voor 2/14 mee;
- b) Bij het aandeel Tesla behaalde ons programma een rendement van slechts 8,15%, terwijl het aandeel zelf met 630% steeg. De AI heeft er klaarblijkelijk weinig van onthouden.

Als argument voor een hoge validiteit dragen we de keuze van de aandelen aan. De trainingsdataset werd willekeurig gekozen en de testdataset was een redelijk representatieve weergave van de totale markt. Wij beschouwen de gevonden resultaten dus als valide.

8.5 Aanbevelingen voor vervolgonderzoek

Als we meer tijd en computerkracht hadden, zouden we een aantal dingen anders doen:

- a) **Meer jaren onderzoeken.** Om ons onderzoek betrouwbaarder te maken, is het vereist om de AI in te zetten op data van meer dan twee jaar. Ons advies zou zijn om tenminste tien jaren te bekijken;
- b) **Aandelen nog beter spreiden, testdataset vergroten.** We hebben ons best gedaan een zo representatief mogelijke testdataset samen te stellen, maar als er meer aandelen in de testdataset worden opgenomen, kan het daadwerkelijke rendement van ons programma nog beter vastgesteld worden. Daarbij komt dat de trainingsdataset op dit moment slechts één bearish aandeel bevat. Ons programma zal waarschijnlijk beter werken als er meer aandelen met een dalende koers in de trainingsdataset worden opgenomen;
- c) **De AI laten werken met meer koerswaarden.** We geven nu alleen de openings- en slotkoersen aan de AI. We denken dat de accuraatheid en het rendement van de AI zullen toenemen als we ook de *intradaywaarden* (waarden tussen openings- en slotkoers) als input geven. Dit vereist veel computerkracht;
- d) **Experimenteren met de vorm en grootte van het neurale netwerk.** We hebben met enkele korte tests vastgesteld dat een neuraal netwerk met 5 inputs, 0 tussenlagen en 2 outputs het beste voor ons zou werken. Als in een volgend onderzoek meer tussenlagen worden toegevoegd, kan het netwerk misschien complexere patronen herkennen;
- e) **Bearishversie concreet implementeren.** We zouden ooit nog wel een programma willen schrijven dat volautomatisch een van de twee AI-versies kiest op basis van de waarde van een indicator;

- f) **Optimaliseren snelheid.** Het kost een redelijk snelle computer⁴⁹ zo'n twintig minuten om het neurale netwerk te trainen. Tijdens dit proces wordt maximaal 30% van de CPU gebruikt. Dit biedt mogelijkheden voor optimalisatie;
- g) **Handelen met opties (short gaan).** Opties werken in de praktijk iets ingewikkelder dan we hebben uitgelegd in hoofdstuk 2.3. Zo hebben opties een (sterk) wisselende prijs, een vervaldatum en zijn er soms wel 50 verschillende versies per aandeel beschikbaar. Het bleek voor ons niet haalbaar om opties in onze AI te verwerken. Dit heeft als direct gevolg dat de AI geen winst kan maken als hij denkt dat het aandeel zal dalen;
- h) **Dividend.** In de simulatie krijgt de AI geen dividend uitgekeerd. Hierdoor valt het rendement (iets) lager uit dan het daadwerkelijk zou zijn;
- i) **Transactiekosten.** De AI houdt geen rekening met stroom-, internet- en transactiekosten. Dit kan opgelost worden door elke dag een constante c van het saldo van de virtuele belegger af te trekken;
- j) **Flitshandel mogelijk maken.** Dat kan nu namelijk niet, omdat het programma simpelweg te traag werkt;
- k) **Uitzoeken wat de AI precies doet.** Ondanks dat we de AI zelf geprogrammeerd hebben, weten we niet precies wat hij doet. In een volgende versie van het programma zou daar meer aandacht aan besteed moeten worden. We denken na over een concrete oplossing voor dit probleem;
- l) **Meerdere aandelen tegelijkertijd bekijken.** Het programma geeft advies over één aandeel per keer. Het zou voor de gebruiker nuttiger zijn als het programma een aantal aandelen tegelijkertijd zou kunnen bekijken en dan een koopadvies kan geven voor het aandeel met de grootste stijgingskans.

⁴⁹ Specificaties: Windows 10 Home, Intel Core i7 950, 24 GB DDR3 RAM, NVIDIA GeForce GTX 750, SSD-opslag.

Literatuurlijst

- 3Blue1Brown. (2017, november 3). *Backpropagation calculus / Deep learning, chapter 4*. Opgehaald van YouTube: <https://www.youtube.com/watch?v=tleHLnjs5U8>
- 3Blue1Brown. (2017, oktober 5). *But what is a Neural Network? / Deep learning, chapter 1*. Opgeroepen op september 10, 2020, van YouTube: <https://www.youtube.com/watch?v=aircAruvnKk>
- 3Blue1Brown. (2017, november 3). *What is backpropagation really doing? / Deep learning, chapter 3*. Opgehaald van YouTube: <https://www.youtube.com/watch?v=llg3gGewQ5U>
- ActueleRentestanden. (2020, november 10). *Hoogste spaarrente - overzicht spaarrekeningen met hoogste rente*. Opgeroepen op december 10, 2020, van ActueleRentestanden: <https://www.actuelerentestanden.nl/sparen/hoogste-spaarrente.asp>
- Anand, P. (2019, januari 23). *Types of Activation Functions used in Machine Learning. OpenGenus IQ: Learn Computer Science*. Opgeroepen op oktober 22, 2020, van <https://iq.opengenus.org/types-of-activation-function/>
- Barlow, S. (2019, mei 8). *Can AI and automated trading predict stock market with 100% accuracy?* Opgeroepen op september 10, 2020, van <https://metro.co.uk/2019/05/06/can-we-trust-machines-to-predict-the-stock-market-with-100-accuracy-9325480/>
- Bartram, S. M. (2017, juni 20). *Agnostic Fundamental Analysis Works*. Opgeroepen op november 21, 2020, van https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2479817
- Beursbrink. (2018, januari 5). *Volatiliteit op de beurs (VIX-index)*. Opgehaald van <https://beursbrink.com/kennisbank/v/volatiliteit/>
- BinckBank Nederland. (sd). *IPO / Wat betekent IPO / Begrippenlijst / BinckBank Nederland*. Opgeroepen op juli 13, 2020, van <https://www.binck.nl/kennis/begrippenlijst/ipo>
- bizjournals.com. (sd). Opgehaald van <https://www.bizjournals.com/chicago/news/2015/02/05/electronic-trading-killed-the-open-outcry-star-cme.html>
- BJF Trading Group. (sd). *Buy Professional Expert Advisors for MT4, MT5 / Best Forex Robots*. Opgeroepen op september 8, 2020, van <https://iticsoftware.com/en/expert-advisors.html>
- Blaauboer, R. (2015, december 22). *Nieuw algoritme kan belegger stinkend rijk maken*. Opgeroepen op december 4, 2020, van Scientias.nl: <https://www.scientias.nl/nieuw-algoritme-kan-belegger-stinkend-rijk-maken/>
- Blauw, S. (2019, augustus 3). *Iedereen heeft het over kunstmatige intelligentie. Maar wat is het?* Opgeroepen op september 2020, 2020, van De Correspondent: <https://decorrespondent.nl/10396/iedereen-heeft-het-over-kunstmatige-intelligentie-maar-wat-is-het/4258555459048-6d83134a>
- Bos, W. (2020, februari 28). *Wat is dividend? Betekenis dividend en berekenen dividrendement*. Opgeroepen op november 4, 2020, van <https://www.lynx.nl/kennis/artikelen/wat-is-dividend/>
- ChartSchool. (sd). *Quadrant Lines [ChartSchool]*. Opgeroepen op november 29, 2020, van https://school.stockcharts.com/doku.php?id=chart_analysis:quadrant_lines#:~:text=Quadrant%20Lines%20divide%20the%20high%2Dlow%20range%20into%20four%20equal

%20sections.&text=The%20top%20line%20marks%20the,not%20an%20indicator%20as
%20such

Chen, J. (2020, mei 7). *Cup and Handle*. Opgeroepen op november 29, 2020, van Investopedia:
<https://www.investopedia.com/terms/c/cupandhandle.asp>

Chicago Board Options Exchange. (2019). *White Paper Cboe Volatility Index®*. Opgeroepen op september 23, 2020, van <https://www.cboe.com/micro/vix/vixwhite.pdf>

CMC Markets. (sd). *What are candlestick charts?* Opgehaald van <https://www.cmcmarkets.com/en/trading-guides/what-are-candlestick-charts>

Code Bullet. (2018, mei 8). *How Artificial intelligence learns / Genetic Algorithm explained*. Opgehaald van YouTube:
https://www.youtube.com/watch?v=VnwjxityDLQ&ab_channel=CodeBullet

Computerphile. (2013, december 30). *The Problem with Time & Timezones - Computerphile*. Opgeroepen op december 13, 2020, van YouTube: https://www.youtube.com/watch?v=-5wpm-gesOY&ab_channel=Computerphile

Corporate Finance Institute. (2019, maart 27). *Open Outcry*. Opgeroepen op september 29, 2020, van <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/open-outcry/>

Desmos.com. (2020). *Desmos / Graphing Calculator*.

Eisenstein, P. A. (2018, september 8). *Tesla stock plummets after Elon Musk smokes weed on live show and two execs quit in one day*. Opgeroepen op november 18, 2020, van nbcnews.com:
<https://www.nbcnews.com/tech/tech-news/tesla-stock-plummets-after-elon-musk-smokes-weed-live-show-n907476>

Elfenbaum, O. (2019, april 29). *How does the stock market work? - Oliver Elfenbaum*. Opgehaald van YouTube: https://www.youtube.com/watch?v=p7HKvqRI_Bo

Encyclo. (sd). *Encyclo*. Opgeroepen op september 23, 2020, van <https://www.encyclo.nl/begrip/Volatiliteit>

Epstein, L., & Roze, G. (2018). *Tradэн voor Dummies*. Amersfoort: BBNC Uitgevers.

EuroNext. (sd). *Amsterdam / European Equities*. Opgeroepen op september 29, 2020, van <https://web.archive.org/web/20131216001709/https://europeanequities.nyx.com/markets/nysse-euronext/amsterdam>

EuroNext. (sd). *AScX-Index*. Opgeroepen op september 29, 2020, van <https://live.euronext.com/nl/product/indices/NL0000249142-XAMS/market-information>

European Journal of Operational Research, Krauss, C., Do, X., & Huck, N. (2017). *Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500*. Opgehaald van <https://doi.org/10.1016/j.ejor.2016.10.031>

Fernando, J. (2020, november 17). *Relative Strength Index (RSI)*. Opgeroepen op december 5, 2020, van Investopedia: <https://www.investopedia.com/terms/r/rsi.asp>

Fiverr. (2020, september 8). *Expert Advisor stocks/forex*. Opgeroepen op september 8, 2020, van Fiverr: <https://www.fiverr.com/gigs/expert-advisor>

FlowTraders. (sd). *About Us / FlowTraders*. Opgeroepen op december 7, 2020, van <https://www.flowtraders.com/about-us>

- Forsythe, K. (2019, augustus 28). *Humans are Terrible at Assessing Risk - Kimberly Forsythe*. Opgeroepen op november 18, 2020, van <https://newworldoptimist.medium.com/humans-are-terrible-at-assessing-risk-a84992ca3e16>
- FSA. (2019, mei 9). *Hoe is het om te werken bij Flow Traders?* Opgeroepen op december 7, 2020, van <https://fsa.nl/for-students/blog/flowtraders/>
- GitHub. (sd). *Electron/electron*. Opgeroepen op december 13, 2020, van <https://github.com/electron/electron>
- Google Finance. (2020, november 4). *Google Finance*. Opgeroepen op november 4, 2020, van <https://www.google.com/finance>
- Hallden, K. (2019, december 6). *I Coded A Trading Bot And Gave It \$1000 To Trade!* Opgehaald van YouTube: <https://www.youtube.com/watch?v=-MHhA-Y3DSk>
- Hessen, A. (sd). *AI voor dummies*. Opgehaald van <https://brs85.nl/schrijfsels/blogs/ai-voor-dummies>
- Horowitz, E. (2013, maart 1). *Why Are People Bad at Evaluating Risks?* Opgeroepen op november 18, 2020, van <https://www.psychologytoday.com/us/blog/the-inertia-trap/201303/why-are-people-bad-evaluating-risks>
- Hui, J. (2020, februari 7). *MAP (mean Average Precision) for Object Detection - Jonathan Hui*. Opgeroepen op november 25, 2020, van <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
- Hulstaert, L. (2018, mei 31). *Gradient descent vs. neuroevolution - Towards Data Science*. Opgeroepen op december 4, 2020, van <https://towardsdatascience.com/gradient-descent-vs-neuroevolution-f907dace010f>
- ig.com. (2019, maart 12). *Expert advisors explained*. Opgeroepen op september 8, 2020, van <https://www.ig.com/en/trading-strategies/expert-advisors-explained-190312>
- Investing With Rose. (2019, mei 30). *Stock Market Index Definition (BEGINNER FRIENDLY EXPLANATION!)*. Opgeroepen op juli 13, 2020, van YouTube: <https://www.youtube.com/watch?v=LxI12aUaabc>
- Investopedia. (2020, december 4). *Koers BRKA*. Opgeroepen op december 4, 2020, van <https://www.investopedia.com/markets/quote?tvwidgetsymbol=BRKA>
- Investopedia. (2020, februari 28). *What Are the Pros and Cons of Hedge Funds?* Opgeroepen op november 2, 2020, van Investopedia: <https://www.investopedia.com/articles/02/111302.asp>
- Investopedia. (sd). *Stock Option*. Opgeroepen op november 23, 2020, van <https://www.investopedia.com/terms/s/stockoption.asp#:~:text=What%20is%20a%20Stock%20Option,that%20a%20stock%20will%20rise>
- Investopedia. (sd). *Tracking Volatility*. Opgeroepen op september 23, 2020, van <https://www.investopedia.com/articles/active-trading/070213/tracking-volatility-how-vix-calculated.asp>
- Investopedia. (sd). *Understanding Expiration Dates*. Opgeroepen op november 23, 2020, van <https://www.investopedia.com/terms/e/expiration-date.asp>
- Jain, V. (2020, maart 28). *Everything you need to know about "Activation Functions" in Deep learning models*. Opgeroepen op november 27, 2020, van

- <https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253>
- Jong, J. (2019, juni 25). *Hoe het menselijke brein ons inspireerde slimme AI te maken*. Opgeroepen op november 25, 2020, van Scientias.nl: <https://www.scientias.nl/hoe-het-menselijke-brein-ons-inspireerde-slimme-ai-te-maken/>
- Jordan, J. (2018, januari 26). *Neural networks: representation*. Opgeroepen op augustus 12, 2020, van Jeremy Jordan: <https://www.jeremyjordan.me/intro-to-neural-networks/>
- K., E. (2007). *Bear and Bull, Reinhard Dachlauer, Frankfurt*. Wikipedia.
- Kahn, J. (2020, augustus 26). *Can an A.I. hedge fund beat the market?* Opgeroepen op december 4, 2020, van fortune.com: <https://fortune.com/2020/08/25/can-an-a-i-hedge-fund-beat-the-market/>
- Kahn, J. (2020, augustus 26). *Can an A.I. hedge fund beat the market?* Opgeroepen op december 4, 2020, van Fortune.com: <https://fortune.com/2020/08/25/can-an-a-i-hedge-fund-beat-the-market/>
- KIJK-redactie. (2012, juni 27). *Hoe computers de beurs beheersen*. Opgeroepen op september 9, 2020, van KIJK: <https://www.kijkmagazine.nl/artikel/beurshandel/>
- Killian, A. (2019, september 30). *Cup and handle chart pattern explained*. Opgeroepen op november 29, 2020, van IG Bank: <https://www.ig.com/en-ch/trading-strategies/cup-and-handle-chart-pattern-explained-190930#information-banner-dismiss>
- Lambert, E. (2010). *The Futures: The Rise of the Speculator and the Origins of the World's Biggest Markets*. New York: Basic Books.
- Langmann, K. (sd). *How to Calculate Standard Deviation in Excel*. Opgehaald van <https://spreadsheetspot.com/standard-deviation-excel/>
- Lauro, D., & Jesan, J. (2003, november 23). *Ubiquity: Human brain and neural network behavior*. Opgehaald van Ubiquity: <https://ubiquity.acm.org/article.cfm?id=958078>
- Macroption. (sd). *VIX Calculation Explained - Macroption*. Opgeroepen op september 23, 2020, van <https://www.macroption.com/vix-calculation/#:%7E:text=VIX%20Calculation%20Step%20by%20Step&text=Calculate%20Step%20day%20variance%20by,The%20result%20is%20VIX>
- Malik, F. (2019, mei 18). *Neural Networks Bias And Weights - FinTechExplained*. Opgeroepen op november 25, 2020, van Medium: <https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da>
- McElroy, M. (2013, januari 2). *While in womb, babies begin learning language from their mothers*. Opgeroepen op november 25, 2020, van UW News: <https://www.washington.edu/news/2013/01/02/while-in-womb-babies-begin-learning-language-from-their-mothers/>
- MissingLink.ai. (sd). *7 Types of Activation Functions in Neural Networks: How to Choose?* Opgehaald van <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>
- ML Glossary documentation. (sd). *Gradient Descent — ML Glossary documentation*. Opgeroepen op november 18, 2020, van Read The Docs: https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

- Nielsen, M. (2019, december). *How the backpropagation algorithm works. Neural networks and deep learning*. Opgeroepen op november 28, 2020, van <http://neuralnetworksanddeeplearning.com/chap2.html>
- NOS. (2015, april 21). *Beurshandelaar opgepakt voor "flash crash" 2010*. Opgeroepen op november 21, 2020, van <https://nos.nl/artikel/2031644-beurshandelaar-opgepakt-voor-flash-crash-2010.html>
- Ondernemen & Internet. (2019, december 15). *Aandelen: wat kun je ermee en wat zijn ze waard?* Opgeroepen op juli 13, 2020, van <https://www.ondernemeninternet.nl/aandelen/#verkopen>
- Outside Insight. (2018, november 30). *How hedge funds employ AI to facilitate trading*. Opgeroepen op december 4, 2020, van <https://outsideinsight.com/insights/how-hedge-funds-employ-ai-to-facilitate-trading/>
- Pablormier. (sd). *Differential Evolution optimizing the 2D Ackley function*. Opgehaald van https://en.wikipedia.org/wiki/Differential_evolution#/media/File:Ackley.gif
- Paperspace.com. (sd). Opgehaald van <https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>
- Pixabay. (2020). *Animal-wildlife-elephant-ivory*. Opgehaald van <https://pixabay.com/nl/photos/olifant-afrika-2923912/>
- Poole, D., Goebel, R., & Mackworth, A. (1998). *Computational Intelligence*. Opgehaald van <https://archive.org/details/computationalint00pool>
- PowerfulJRE. (2019, augustus 28). *Joe Rogan Experience #1169 - Elon Musk*. Opgeroepen op november 18, 2020, van YouTube: <https://newworldoptimist.medium.com/humans-are-terrible-at-assessing-risk-a84992ca3e16>
- reddit.com. (2012, december 21). *I used to be a penny stock promoter in the late 90's, AMA and learn how to spot these scams!* Opgeroepen op juli 10, 2020, van https://www.reddit.com/r/investing/comments/158vi7/i_used_to_be_a_penny_stock_promoter_in_the_late/
- Rodrigues, R. (2019, oktober 2). *What does the final layer of a deep neural network return?* Opgeroepen op december 10, 2020, van Quora.com: <https://www.quora.com/What-does-the-final-layer-of-a-deep-neural-network-return>
- Ross, M. (2018, juni 6). *Under The Hood of Neural Network Forward Propagation — The Dreaded Matrix Multiplication*. Opgeroepen op november 30, 2020, van Medium: <https://towardsdatascience.com/under-the-hood-of-neural-network-forward-propagation-the-dreaded-matrix-multiplication-a5360b33426>
- Rout, S. (2020, augustus 25). *10 Different activation functions for activating the neurons (Deep Learning)*. Opgehaald van Medium: <https://medium.com/@ss6239/10-different-activation-functions-for-activating-the-neurons-deep-learning-81ea8618743e>
- Ruder, S. (2020, maart 20). *An overview of gradient descent optimization algorithms*. Opgeroepen op november 18, 2020, van <https://ruder.io/optimizing-gradient-descent/>
- Russell, S., Norvig, P., & Canny, J. (2003). *Artificial Intelligence* (2e ed.). Opgehaald van <http://aima.cs.berkeley.edu/>
- Schneier, B. S. (2017, juni 5). *Why the Human Brain Is a Poor Judge of Risk*. Opgeroepen op november 18, 2020, van <https://www.wired.com/2007/03/security-matters0322/>

- Schuermann, L. (2016, februari 27). *The Math Behind Backpropagation* / Lucas V. Schuermann. Opgeroepen op november 18, 2020, van <https://lucasschuermann.com/writing/math-behind-backpropagation>
- Scott, T. (2019, februari 4). *Slowing Down A Stock Exchange With 38 Miles Of Cable*. Opgehaald van YouTube: <https://www.youtube.com/watch?v=d8BcCLLX4N4>
- SEC.gov. (2011, maart 10). *Stop Order*. Opgeroepen op september 8, 2020, van <https://www.sec.gov/fast-answers/answersstopordhtm.html>
- Securities and Exchange Commission. (2009, november 12). *Important Information on Penny Stocks*. Opgeroepen op juli 10, 2020, van <https://www.sec.gov/investor/schedule15g.htm>
- Sesay, D. (2018, januari 2). *Do 90% of day traders lose money, because I have been trading for only a short period of time and my principal has increased by 30%, so is this simply beginners luck?* Opgeroepen op november 23, 2020, van Quora: <https://www.quora.com/Do-90-of-day-traders-lose-money-because-I-have-been-trading-for-only-a-short-period-of-time-and-my-principal-has-increased-by-30-so-is-this-simply-beginners-luck>
- Sharma, S. (2019, december 10). *Activation Functions in Neural Networks - Towards Data Science*. Opgeroepen op november 25, 2020, van Medium: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- Silva, S. (2020, maart 28). *The Maths behind Back Propagation - Towards Data Science*. Opgeroepen op november 20, 2020, van Medium: <https://towardsdatascience.com/the-maths-behind-back-propagation-cf6714736abf>
- Skalski, P. (2020, februari 16). *Deep Dive into Math Behind Deep Networks - Towards Data Science*. Opgeroepen op november 25, 2020, van Medium: <https://towardsdatascience.com/https-medium-com-piotr-skalski92-deep-dive-into-deep-networks-math-17660bc376ba>
- SpeedTrader. (2019, december 18). *How to Read Stock Market Options Chains*. Opgeroepen op november 23, 2020, van <https://speedtrader.com/how-to-read-stock-market-options-chains/>
- Stanley, K. O. (2018, oktober 2). *Welcoming the Era of Deep Neuroevolution*. Opgeroepen op december 2020, 2020, van <https://eng.uber.com/deep-neuroevolution/>
- StockCharts. (sd). *Moving Average Envelopes [ChartSchool]*. Opgeroepen op november 29, 2020, van https://school.stockcharts.com/doku.php?id=technical_indicators:moving_average_envelopes
- Taleb, N. N. (2008). *De Zwarde Zwaan*. Amsterdam, Nederland: Nieuwezijds.
- The Balance. (2020, juni 29). *How Much Money Do Stock Day Traders Make?* Opgeroepen op november 2, 2020, van The Balance: <https://www.thebalance.com/how-much-money-stock-day-traders-make-1031069>
- This Person Does Not Exist. (sd). *This Person Does Not Exist*. Opgeroepen op november 23, 2020, van <https://thispersondoesnotexist.com/towardsdatascience.com>.
- (sd). *A mostly complete chart of Neural Networks*. Opgehaald van <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>
- U.S. Commodity Futures Trading Commission & U.S. Securities & Exchange Commission. (2010, september). *FINDINGS REGARDING THE MARKET EVENTS OF MAY 6, 2010*. Opgehaald van <https://www.sec.gov/news/studies/2010/marketevents-report.pdf>

U.S. Securities and Exchange Commission. (sd). *Trading Basics: understanding the Different Ways to Buy and sell stock*. Opgeroepen op september 8, 2020, van <https://www.sec.gov/investor/alerts/trading101basics.pdf>

Wikipedia contributors. (2020, oktober 24). *AEX index*. Opgeroepen op december 10, 2020, van Wikipedia: https://en.wikipedia.org/wiki/AEX_index#Annual_Returns

Wikipedia contributors. (2020, november 22). *Futures contract*. Opgeroepen op november 23, 2020, van https://en.wikipedia.org/wiki/Futures_contract

Wikipedia contributors. (2020, augustus 28). *Investment strategy*. Opgeroepen op september 9, 2020, van https://en.wikipedia.org/wiki/Investment_strategy

Wikipedia contributors. (2020, juni 5). *Order (exchange)*. Opgeroepen op september 8, 2020, van [https://en.wikipedia.org/wiki/Order_\(exchange\)#Conditional_orders](https://en.wikipedia.org/wiki/Order_(exchange)#Conditional_orders)

Wikipedia contributors. (2020, april 27). *Oscillator (technical analysis)*. Opgeroepen op december 2, 2020, van [https://en.wikipedia.org/wiki/Oscillator_\(technical_analysis\)](https://en.wikipedia.org/wiki/Oscillator_(technical_analysis))

Wikipedia contributors. (2020, december 6). *S&P 500 Index*. Opgeroepen op december 10, 2020, van Wikipedia: https://en.wikipedia.org/wiki/S%26P_500_Index#Returns_by_year

Wikipedia-bijdragers. (2018, maart 28). *Bibliotheek (informatica)*. Opgeroepen op december 12, 2020, van [https://nl.wikipedia.org/wiki/Bibliotheek_\(informatica\)](https://nl.wikipedia.org/wiki/Bibliotheek_(informatica))

Wikipedia-bijdragers. (2018, december 6). *Dividend*. Opgeroepen op november 4, 2020, van https://nl.wikipedia.org/wiki/Dividend#Juridische_beperkingen

Wikipedia-bijdragers. (2019, december 6). *Koersgrafiek*. Opgeroepen op november 12, 2020, van <https://nl.wikipedia.org/wiki/Koersgrafiek>

Wikipedia-bijdragers. (2019, september 11). *Primaire emissie*. Opgeroepen op juli 13, 2020, van https://nl.wikipedia.org/wiki/Primaire_emissie

Wikipedia-bijdragers. (2019, juli 26). *Volatiliteit*. Opgehaald van <https://nl.wikipedia.org/wiki/Volatiliteit>

Wikipedia-bijdragers. (2020, juli 27). *AMX Index*. Opgeroepen op september 29, 2020, van https://nl.wikipedia.org/wiki/AMX_Index

Wikipedia-bijdragers. (2020, april 5). *Backtesten*. Opgeroepen op november 23, 2020, van Wikipedia: <https://nl.wikipedia.org/wiki/Backtesten>

Wikipedia-bijdragers. (2020, april 5). *Backtesten*. Opgeroepen op november 10, 2020, van Wikipedia: <https://nl.wikipedia.org/wiki/Backtesten>

Wikipedia-bijdragers. (2020, juni 8). *Conjunctuur*. Opgeroepen op november 2020, 2020, van <https://nl.wikipedia.org/wiki/Conjunctuur>

Wikipedia-bijdragers. (2020, maart 18). *Dow Jones Industrial Average*. Opgeroepen op juli 13, 2020, van https://nl.wikipedia.org/wiki/Dow_Jones_Industrial_Average

Wikipedia-bijdragers. (2020, februari 24). *Effectenbeurs*. Opgeroepen op juli 10, 2020, van <https://nl.wikipedia.org/wiki/Effectenbeurs>

Wikipedia-bijdragers. (2020, november 25). *Fitness function*. Opgeroepen op november 27, 2020, van Wikipedia: https://en.wikipedia.org/wiki/Fitness_function

Wikipedia-bijdragers. (2020, november 26). *Genetic algorithm*. Opgeroepen op november 28, 2020, van Wikipedia: https://en.wikipedia.org/wiki/Genetic_algorithm#Initialization

Wikipedia-bijdragers. (2020, augustus 13). *Relatieve Sterkte Index*. Opgeroepen op december 5, 2020, van https://nl.wikipedia.org/wiki/Relatieve_Sterkte_Index

Wikipedia-bijdragers. (2020, juni 11). *S&P 500*. Opgeroepen op juli 13, 2020, van https://nl.wikipedia.org/wiki/S%26P_500

Wikipedia-bijdragers. (2020, oktober 13). *Selection (genetic algorithm)*. Opgeroepen op november 27, 2020, van Wikipedia: [https://en.wikipedia.org/wiki/Selection_\(genetic_algorithm\)](https://en.wikipedia.org/wiki/Selection_(genetic_algorithm))

Wikipedia-bijdragers. (2020, april 7). *Standaardafwijking*. Opgehaald van <https://nl.wikipedia.org/wiki/Standaardafwijking>

Wikipedia-bijdragers. (2020, juni 6). *Technische analyse*. Opgeroepen op november 21, 2020, van https://nl.wikipedia.org/wiki/Technische_analyse

Wikipedia-bijdragers. (2020, november 27). *Unix time*. Opgeroepen op december 10, 2020, van Wikipedia: https://en.wikipedia.org/wiki/Unix_time

Wikipedia-bijdragers. (2020, augustus 31). *Vanishing gradient problem*. Opgeroepen op oktober 12, 2020, van Wikipedia: https://en.wikipedia.org/wiki/Vanishing_gradient_problem

Wikipedia-bijdragers. (2020, juni 23). *Vereenigde Oostindische Compagnie*. Opgeroepen op juli 10, 2020, van https://nl.wikipedia.org/wiki/Vereenigde_Oostindische_Compagnie

wikiwand.com. (sd). *Nervenzelle*. Opgehaald van <https://www.wikiwand.com/de/Nervenzelle>

Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*. Trend Research.

Yahoo Finance. (2020, december 10). *Historische koersdata*. Opgehaald van Yahoo Finance: <https://finance.yahoo.com>

Yardeni, E., Yardeni Research, Inc., & Abbott, J. (2020, november 30). *Stock Market Indicators: S&P 500/400/600 Divisors*. Opgeroepen op december 4, 2020, van <https://www.yardeni.com/pub/spdivisors.pdf>

Appendix I: logboek

Zie volgende 7 pagina's.

Datum	Stach	Omschrijving	Noah	Sander	Tijd besteed (minuten)		Totale tijd (minuten)		Alles bij elkaar		Opmerkingen / Problemen / (Nieuwe) doelen	Noah	Sander		
					Stach	Noah	Sander	Stach	Noah	Sander	Totaal				
08/05/2020		Allereerste brainstorm, orientatie onderzoek (doc)			60	60	60	60	60	60	3 uur	1.25%			
13/05/2020		Beknopt plan van aanpak opstellen, begeleider zoeken, pitchen (doc)			60	120	120	120	120	120	6 uur	2.50%			
20/06/2020		Kort onderzoek naar literatuurbronnen, oriënteren (doc)			15	135	135	135	135	135	6.75 uur	2.81%			
28/06/2020		Hoofd- en deelvragen bedacht (doc)			30	165	165	165	165	165	8.25 uur	3.44%			
03/07/2020		Gesprek met mevrouw Van de Ven			20	185	185	185	185	185	9.25 uur	3.85%			
04/07/2020		Oppervlakkige orientatie en oefening met TensorFlow			70	120	185	255	305	12.41 uur	5.17%				
09/07/2020	Leidraad voorlopige hoofdstukindeling opgesteld (doc)				25	210	255	305	305	12.83 uur	5.35%	Dit document schept een beetje orde in de chaos. Zo wordt duidelijk welke deelvragen bij welk hoofdstuk passen.			
09/07/2020	Bronnen voor hoofdstuk 1 gezocht van drie filmpjes (doc)				90	300	255	305	14.33 uur	5.97%			Misschien wel handig om deze samenvattingen even te lezen		
09/07/2020	Hoofdstukindeling aangevuld (doc)				25	300	255	330	14.75 uur	6.15%			Opmerking van Sander heb ik behandeld. Zelf ook nog twee opmerkingen bij werk gezet.		
10/07/2020	Completeren uittreksels filmpjes (doc)				30	330	255	330	15.25 uur	6.35%					
10/07/2020	Gewerkt aan hoofdstuk 'aandelen algemeen' (doc)				120	450	255	330	17.25 uur	7.19%					
10/07/2020	Gewerkt aan hoofdstuk 'aandelen algemeen' (doc)				100	550	255	330	18.91 uur	7.88%					
13/07/2020	Documentaire over de financiële crisis en hoe opties hier een rol in speelden gekeken				130	680	255	330	21.08 uur	8.78%					
16/08/2020													Status van project en onderzoek besproken. Artikel over een AI hedgefund bekijken en gelezen. Aankondiging gesprek met Van de Ven doorgesproken. Taken voor week A verdeeld: Stach doet hoofdstuk computersystemen aandelenhandel, Sander doet Programmerenbenodigdheden 1+3. Streetdeadline 12 september.		
03/09/2020	Tweemaal 'pauzemeting' over planning en voortgang				40	720	295	370	23.08 uur	9.62%					
04/09/2020	Overleg met Sander over de indeling-sheets/verktracker in dit logboekdocument					80	730	295	450	24.58 uur	10.24%				
04/09/2020													GitHub gemaakt zodat Noah en ik allebei aan de code kunnen werken en extra pagina toegevoegd aan dit logboek waar we de documenten etc. bijhouden		
05/09/2020	Diverse werkzaamheden waaronder het klarzettten van documenten, bekijken van de tekst van Sander en aanvullen van het 'inleidingstabblad'					40	300	770	295	750	30.25 uur	12.60%			
07/09/2020	Gesprek met mevrouw Van de Ven					30	800	325	780	31.75 uur	13.23%			Voortgang besproken	
07/09/2020	Bronnen verzameld en eerste onderzoek gedaan voor hoofdstuk 'aandelenhandel' (doc)					35	180	180	835	505	960	38.33 uur	15.97%		
08/09/2020	Gewerkt aan hoofdstuk 'huidige computersystemen in de aandelenhandel' (doc). Filmpje en filmpje gezien (nuttig) over AI, zodat ik ook begrijp waar het over gaat.					150	985	505	960	40.83 uur	17.81%			Dit is lijkt me ook een nuttige site: http://neurallnetworksanddeeplearning.com/	
09/09/2020	Pauzemeeting					15	1.000	520	975	41.58 uur	17.33%			Voortgang besproken	
09/09/2020	Gewerkt aan hoofdstuk 'computerystemen aandelenhandel' (doc)					60	180	60	1.060	700	1.035	46.58 uur	19.41%		
10/09/2020	Gewerkt aan hoofdstuk 'computerystemen aandelenhandel' (doc). Doorlozen van het prijsvraagende PWS over AI om te kijken hoe alles hebben verwoerd en opgeschreven					170	30	1.230	700	1.065	49.91 uur	20.80%		Zie opmerkingen in het tablad 'Indeling'. + Kan men meedelen over de volgende vragen: 1) wat zijn de nadelen van AI op de beurs? en 2) waarom zit er wel toekomst in AI op de beurs?	
11/09/2020	Gewerkt aan hoofdstuk 'computeristemmen aandelenhandel' (doc) en gewerkt aan hoofdstuk aandelen geavanceerd' (doc)					150	280	280	1.380	980	1.345	61.75 uur	25.73%		Nadelen: Er bestaan geen patronen in de aandelenmarkt. Het kan dus nooit perfect worden voorspeld.

Datum	Stach	Omschrijving	Noah	Opmerkingen / Problemen / (Nieuwe) doelen								Sander	
				Sander	Tijd besteed (minuten)	Total	Noah	Sander	Stach	Total	% klaar		
12/09/2020		Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc) Noah's codeblock gefixed op meerdere fronten (doc)		120	1,380	980	1,465	63.75 uur	63.75 uur	63.75 uur	26.56%	Buiten het feit dat de JSON codeblock er niet uitzag was hij ook nog qua functionaliteit compleet fout...	
13/09/2020		Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc) en gewerkt aan hoofdstuk 'aandelen geavanceerd' (doc) ook heb ik de indeling van hoofdstuk 4 beter uitgespreid.		80	1,380	980	1,545	65.08 uur	65.08 uur	65.08 uur	27.12%		
14/09/2020	PWS-boekje doorgelezen. Deadlines en criteria verzameld en vermeld in de juiste documenten. Diverse kleine werkzaamheden verricht. Nieuw type trainings-AI ding (weet ik veel al dat jargon) ontdekt waarvan ik denk die AI goede resultaten gaat opleveren.	Gewerkt in het hoofdstuk 'Algoritmes en AI' aan 'Voorbeeld van gebruik' (doc).		60	40	1,440	980	1,585	66.75 uur	66.75 uur	27.81%	Beoordelingscriteria: plan v.aanpak; 10%, logboek 5%; inhoud 60%; lay-out 10% en proces 15%. Dan het type AI. Zie dit filmpje . Denken we dat hier mogelijk is om een x aantal virtuele beleggers tegen elkaar te laten spelen en dan de beste laten evolueren om zo een nieuwe generatie te krijgen?	
14/09/2020	Vooruitgang besproken in de pauze	Strategisch overleg over het profielwerkstuk			15	1,455	995	1,600	67.5 uur	67.5 uur	28.13%	Stach legt de AI-strategie uit die hij gisteren heeft ontdekt. Sander zegt dat hij daar niet alleen aan kan beginnen en er Noah voor nodig heeft. Noah zegt toe over enkele dagen te beginnen met programmeren. Hier nog een ander goed profielwerkstuk over AI.	
15/09/2020	Voorwoord (doc) geschreven	Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc)			15	1,470	1,010	1,615	68.25 uur	68.25 uur	28.44%	Lekker veel tekst verwijderd omdat ik erachter kwam dat ik alles al had gezegd bij Algoritmes en AI. Vinden jullie dat ik meer moet bedenken bij mijn stukje in 'Programmeerberoendigheden' of moet ik het zo laten?	
15/09/2020		Strategisch overleg over het profielwerkstuk			60	100	1,530	1,010	1,715	70.91 uur	29.55%	Noah zegt toe vandaag een stukje te gaan schrijven. Stach zegt dat hij iets meer onderzoek gaat doen naar de mogelijkheden van een evoloving-AI.	
16/09/2020	Het en ander uitgezocht over mijn evolwing-AI-idee. En een super high-tech formule ingebouwd zodat de grafiek op het tabblad prestaties alleen nog waardes geeft bij echte entries in deze log.	Gewerkt aan hoofdstuk 'programmeerbendoigheden' (doc)			10	1,540	1,020	1,725	71.41 uur	71.41 uur	29.76%	Het algoritme dat jullie moeten hebben, heet NEAT (Neuroevolution of Augmenting Topologies). Er is veel documentatie over beschikbaar, en hele duidelijke instructies over hoe te implementeren zijn hier te vinden. Leuker kan ik het niet maken, wel makkelijker.)	
16/09/2020		Gewerkt aan hoofdstuk 'Programmeerbendoigheden'			40	240	40	1,580	1,260	1,765	76.75 uur	31.98%	
22/09/2020	Gewerkt aan hoofdstuk 'Bijna' alles wat je moet weten over aandelen (doc)				50	1,630	1,260	1,765	77.58 uur	77.58 uur	32.33%		
23/09/2020	Gewerkt aan hoofdstuk 'Bijna' alles wat je moet weten over aandelen (doc)				65	1,695	1,260	1,765	78.66 uur	78.66 uur	32.78%		
24/09/2020	Gewerkt aan hoofdstuk 'Bijna' alles wat je moet weten over aandelen (doc)	Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc)			40	1,735	1,260	1,805	80 uur	80 uur	33.33%		
26/09/2020	Onderzoek gedaan naar de mogelijkheden en limitaties van het NEAT-algoritme	Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc)				200	1,775	1,260	2,005	84 uur	84 uur	35.00%	Korte samenvatting. NEAT staat ons in staat om evolutie in een snelle reisvaart na te bootsen. Ingebouwde mogelijkheden voor gemutaties en survival-of-the-fittest scenario's. Limitatie/problem: training vindt zeer gedacht plaats. Er moeten specifieke criteria worden opgesteld aan de hand waarvan geëvolueerd wordt.
27/09/2020	Diverse kleine werkzaamheden, o.a. naar werk van Noah kijken	Gewerkt aan hoofdstuk 'programmeerbendoigheden' (doc)										Reactie op Noah: lekker bezig. Helemaal akkoord met het nieuwe idee dat je wil schrijven. Verget mij favoute NEAT niet.:-)	
28/09/2020	Noah en Stach hebben samen hard gewerkt aan het PWS. Erst een kort overleg (notulen). Gedaan en daarna heeft Stach gewerkt een hoofdstuk 1 alles wat je moet weten over aandelen (doc) en Noah aan het programma (GitHub).	Gewerkt aan programma (GitHub)			5	280	120	1,780	1,540	2,125	90.75 uur	37.81%	Ik heb mijzelf aan de hand van een voorbeeld van een AI die Flappy Bird speelt goed geleerd hoe ik moet forex moet nog. Daar ben ik vandaag niet aan toegekomen.
		Gewerkt aan programma (GitHub)			120	1,780	1,540	2,245	92.75 uur	92.75 uur	38.65%	Onderzoek gedaan naar betere Python GUI libraries	
29/09/2020	Noah en Stach hebben samen hard gewerkt aan het PWS. Erst een kort overleg (notulen). Gedaan en daarna heeft Stach gewerkt een hoofdstuk 1 alles wat je moet weten over aandelen (doc) en Noah aan het programma (GitHub).	Gewerkt aan hoofdstuk 'Algoritmes en AI' (doc)			180	320	110	1,960	1,860	2,355	102.91 uur	42.88%	Alles waar ik aan werkte is nu klaar voor eindredactie. Als het goed is moet forex nog wel. Daar ben ik gisteren een grote veranderingen mee aan te brengen. (gesloten)

Datum	Stach	Omschrijving	Noah	Sander	Tijd besteed (minuten)	Totale tijd (minuten)	Alles bij elkaar	Stach	Noah	Sander	Opmerkingen / Problemen / (Nieuwe) doelen	Noah	Sander		
01/10/2020		Ik heb nu ook filmpje 3 en 4 van de Afspeellijst (link) gezien en ik heb me verdiept in de schriftwijzer (doc) voor academische rapporten.		Diverse kleine werkzaamheden	90	10	2,050	1,860	2,365	104,58 uur	43 . 58%	De calculus achter backpropagation vind ik nog wat ingewikkeld, maar ik zag dat Sander het begrijpelijk behandeld heeft in een van zijn hoofdstukken. Zie dit document voor de schrifwijzer van het Rathenau Instituut. Als we allemaal (ongeveer) dezelfde stijlregels aanhouden, scheelt dat mij veel tijd in de eindredactie.			
02/10/2020	Strategisch voorgangsgesprek	Strategisch voorgangsgesprek		Diverse werkzaamheden aan het hoofdstuk 'Algoritmes en AI' (doc) en 'Programmeerbenedigdheden (doc)	20	20	2,070	1,860	2,385	105,25 uur	43 . 85%	Door de volle planning van mevrouw Van de Ven lukt het ons niet om een voortgangsgesprek voor de deadline van 7 oktober te plannen. Daarom hebben we tijdens de wiskunde-D-les kort over PWS gesproken en besloten om het échte voortgangsgesprek te verplaatsen naar vrijdag (9/10) het 6e d-uur.			
03/10/2020		Gesprek met mevrouw Van de Ven & overleg onderling			15	2,070	1,860	2,400	105,5 uur	43 . 96%					
05/10/2020		Deadline: uiterlijk contact met begeleider, bespreken voortgang			5	2,075	1,865	2,405	105,75 uur	44 . 06%					
07/10/2020		Gesprek met mevrouw Van de Ven Verplaatsd naar 9 oktober			50	2,125	1,915	2,455	108,25 uur	45 . 10%	Mevrouw Van de Ven gaat de hoofdstukken die Stach in de eindredactie gedaan heeft hakken en commentaar geven waar nodig. Sander en Noah gaan druk in de weer met programmeren. Noah loopt qua tijd achter op Stach en Sander, dus vooraf Noah moet nu een groot deel van het werk doen. We spreken weer af in de eerste week na de herfstvakantie. Lijst op een dinsdag en anders op donderdag.				
09/10/2020				Werkzaamheden aan het hoofdstuk 'Programmeerbenodigdheden' (doc) en gewerkt aan Algoritmes en AI' (doc)	90	2,125	1,915	2,545	109,75 uur	45 . 73%					
09/10/2020				Gewerkt aan Algoritmes en AI' (doc)	90	2,125	1,915	2,635	111,25 uur	46 . 35%	Sander vroeg me vandaag of ik feedback kan geven op zijn hoofdstuk 'Algoritmes en AI'. Dat heb ik gedaan. Ik denk dat ik zo'n 50 inhoudelijke opmerkingen in de kantlijn heb gezet.				
13/10/2020		Inhoudelijke feedback gegeven op 'Algoritmes en AI' (doc)			60	2,185	1,915	2,635	112,25 uur	46 . 77%					
14/10/2020		Gewerkt aan inleiding (doc)		Gewerkt aan 'Algoritmes en AI' (doc)	50	15	2,235	1,915	2,650	113,33 uur	47 . 22%				
15/10/2020		Filmpjes (1 / 2 / 3 / 4) getekend over het proces om een neuraal netwerk met NEAT te maken en dit artikel (link) over neurale netwerken gelezen.			60	2,295	1,915	2,650	114,33 uur	47 . 64%					
18/10/2020				Gewerkt aan 'Algoritmes en AI' (doc)	120	2,295	1,915	2,770	116,33 uur	48 . 47%					
19/10/2020		Gesprek met Sander		Gesprek met Stach	90	2,385	1,915	2,860	119,33 uur	49 . 72%	We hebben uitgebreid gekeken naar hoofdstuk 4...				
20/10/2020		Gewerkt aan 'Inleiding' (doc) en de opmerkingen van mevrouw Van de Ven bekeken en indien mogelijk herwerkt.		Gewerkt aan het programma (GitHub). Met Stach overlegd over de huidige functies en de toekomst van het programma. Ik heb Stach geleerd hoe hij PyCharm moet gebruiken. Samen met Stach een 'Actieplan programmeren' (doc) opgesteld.							De feedbackkolom werkt zo: - De letters SR/NV/SI/MV staan voor resp. Stach, Noah, Sander en Van de Ven - Als iemand een onderdeel van het verslag heeft bekeken en geen opmerking heeft, kleurt deze persoon het vakje bij zijn letter groen. - Als iemand een onderdeel van het verslag heeft bekeken en wel opmerkingen heeft, kleurt deze persoon het vakje bij zijn letter oranje.				
28/10/2020				Gewerkt aan 'Inleiding' (doc) en feedback gegeven op Huidige computersystemen (doc).	170	350	70	2,555	2,265	2,930	129,16 uur	53 . 82%	Het was vooraf oefening met de API aangeraden ik een nieuwe uitdelen heb. Er zitten zeker ook wel nuttige elementen in voor net uiteindelijke programma's.		
29/10/2020		Data-analyse (doc) gemaakt van de PWS-tijd tot aan de deadline.		Hoofdstuk 1 doogelezen en waar nodig feedback gegeven (doc).	120	60	2,675	2,265	2,990	132,16 uur	55 . 07%	Gewerkt aan 'V2 Algoritmes en AI' (doc).			
29/10/2020				Gesprek met mevrouw Van de Ven		20	2,695	2,285	3,010	133,16 uur	55 . 49%	Voortgangsgesprek. Noah heeft onder andere de huidige status van de code laten zien.			

Datum	Omschrijving		Opmerkingen / Problemen / (Nieuwe) doelen					
	Stach	Noah	Sander	Stach	Noah	Sander	Stach	Noah
30/10/2020	Gxperimenteerd met indicatoren via Yahoo finance. (link) . Gewerkt aan hoofdstuk 'Aandelen algemeen' (doc). Werk van Sander bekijken. De nieuwe versie van het programma bekijken en getest. Gewerkt aan Inleiding (doc).		Gewerkt aan V2 Algoritmes en AI' (doc)		Gewerkt aan het programma (GitHub)			Ook onderzoek gedaan naar hoe we de GUI van de app gaan doen. Waarschijnlijk ElectronJS. Opgezocht hoe dat moet werken aangezien ElectronJS niet origineel gemaakt is voor Python maar ik heb er wel al ervaring mee. Volgende sites kunnen handig zijn: 1 2 3
31/10/2020	Op verzoek van Sander naar zijn stukje over 'de enkele laag' gekomen en een deel hiervan herschreven (doc). Ik ben gekomen tot het stuk waar Sander over de matrices begint.		Gewerkt aan het programma (GitHub)		Gewerkt aan V2 Algoritmes en AI' (doc)			
01/11/2020			Gewerkt aan het programma (GitHub)		Gewerkt aan V2 Algoritmes en AI' (doc)			Tweede stap actieplan (doc) af. Heel klein begin aan de derde stap.
02/11/2020	Boek 'Tradex voor Dummies' doorgebladert. Tot de conclusie gekomen dat we een paar cruciale dingen te missen. Hoofdstukindeling aangepast. Daarna gewerkt aan hoofdstuk V2 Algoritmes en AI' (doc). Ook gewerkt aan de inleiding (doc). Tot slot gewerkt aan hoofdstuk 'Aandelen algemeen' (doc).		Inleiding doorgelezen & feedback gegeven		180	10	3.170	2.685
03/11/2020			Gewerkt aan V2 Algoritmes en AI' (doc)		Gewerkt aan het programma (GitHub)			
04/11/2020	Stach heeft een update gekregen over de huidige status van het programma. Feedback gegeven op het programma. Met Noah gepraat over voorstellen voor hoofdstuk 6. Sander: Gewerkt aan hoofdstuk 'Aandelen algemeen' (doc).		Voortgangsgesprek		20	3.190	3.205	3.665
04/11/2020	Gesprek met Sander over hoe we de theorie achter NEAT het beste kunnen opnemen in ons verslag. Inlezen NEAT. Feedback gegeven op werk van Sander. Onderzoek gedaan naar de werking achter het NEAT-algoritme en bronnen verzameld. Zie ook opmerking.		Stach heeft Noah bijgepraat over het programma.		200	80	235	3.390
05/11/2020			Gewerkt aan 'Aandelen algemeen' (doc). Stukje 'Neurone: de kleinste eenheid van het netwerk' (doc) opmerkingen' voor een belangrijke ontdekking over NEAT. Samen met Sander geketen naar zijn werk.		Gewerkt aan het programma (GitHub)			
06/11/2020	De afgelopen paar dagen wat hoofdstukken gelezen uit 'Tradex voor Dummies'. Deel 1.		Gesprek met Sander over hoe we de theorie achter NEAT het beste kunnen opnemen in ons verslag. Inlezen NEAT. Feedback gegeven op werk van Sander. Onderzoek gedaan naar de werking achter het NEAT-algoritme en bronnen verzameld. Zie ook opmerking.		100	70	3.490	3.285
16/11/2020			Gewerkt aan V2 Algoritmes en AI' (doc). Gesprek met Stach over hoe we de theorie achter NEAT het beste kunnen opnemen in ons verslag. Inlezen NEAT. Feedback gegeven op werk van Stach. Gewerkt aan het programma (GitHub)		140	150	3.630	4.120
					60	3.690	3.285	4.120

Datum	Omschrijving		Opmerkingen / Problemen / (Nieuwe) doelen								
	Stach	Noah	Sander	Stach	Noah	Sander	Total	Alles bij elkaar	Stach	Noah	Sander
18/11/2020	Voortgangsgesprek met Sander, overleg over de verdere invulling van het NEAT-hoofdstuk en aanpassingen gedaan aan hoofdstuk V2 Algoritmes en AI' (doc). Gewerkt aan hoofdstuk 'NEAT' (doc). Hoofdstuk 5, deel 'AI' (doc) kan nu ook naar eindredactie.	Gewerkt aan het programma (GitHub)	Voortgangsgesprek met Stach, overleg over de verdere invulling van het NEAT-hoofdstuk en aanpassingen gedaan aan het stukje over gradient descend (gradient).	360	240	60	4.050	3.525	4.180	195.91 uur	81.63%
19/11/2020	Code van Noah bekijken, ingelezen op indicatoren, trends en andere analysemethoden. UI van Sander bekijken. Wenselijkste voor functies voor het programma/de UI opgesteld :)	Betere inleiding over NEAT, een serie gevuld om een AI een spel te laten besturen, en ik heb nu een relatief goed idee van wat ik moet bedenken voor het programma van ons.	Voortgangsgesprek met Stach. Wenselijkste gesprek met Stach, en AI' een spel plannen gemaakt. Gewerkt aan V2 Algoritmes en AI' (doc)	100	300	4.150	3.825	4.180	202.58 uur	84.41%	
20/11/2020	Gewerkt aan het stukje over 'Verliestfunctie & gradient descent' in 'V2 Algoritmes en AI' (doc). Voortgangsgesprek met Sander. Wenselijkste besproken en plannen opgesteld. Feedback gegeven op werk van Sander. Diverse kleine klusjes gedaan.	Gewerkt aan hoofdstuk 'Aandelen geavanceerd' (doc). Gesprek met Stach feedback gegeven op een deel het onderzoek wat af is van de anderen. Gewerkt aan het programma (GitHub)	175	160	4.325	3.825	4.340	208.16 uur	86.74%		
20/11/2020	Deadline: uiterlijk contact met begeleider, bespreken voortgang.		4.325	3.825	4.340	208.16 uur	86.74%	<i>Dit gesprek voeren wij in de week van 23 november.</i>			
21/11/2020	Gewerkt aan hoofdstuk 'Aandelen geavanceerd' (doc). Gesprek met Noah over de status van het programma. Overleg gehad met Sander over de mogelijkheden en limitaties van de GUI.	Gesprek met Stach feedback gegeven op een deel het onderzoek wat af is van de anderen. Gewerkt aan het programma (GitHub)	360	340	460	4.685	4.165	4.800	227.5 uur	94.79%	
22/11/2020	Deadline: programma is af.		4.685	4.165	4.800	227.5 uur	94.79%	Noah en Sander hebben het programma af. Ergens in de week van 23 november spreken we af met mevrouw Van de Ven om het programma door te nemen.			
22/11/2020	Aandelen onderzocht die duidelike trends vertonen en daar grafiekes van gemaakt. Gewerkt aan hoofdstuk aandelen geavanceerd' (doc).	Gewerkt aan GUI van programma. (GitHub)	180	200	4.865	4.165	5.000	233.83 uur	97.43%		
23/11/2020	Hard gewerkt aan hoofdstuk 'Aandelen geavanceerd' (doc). Gewerkt aan inleiding (doc). Voorblad ontworpen (doc) zodat we het niet met zo crappy Wordtemplate hoeven te doen.	Gesprek met Noah. Gewerkt aan het programma (GitHub)	375	420	360	5.240	4.585	5.360	253.08 uur	185.45%	
24/11/2020	Samen met Noah een beetje geëxperimenteerd met NEAT. Gekken na de feedback van Sander. Paar stukjes van Sander op 'afgerond' gezet (doc). Nieuwe kolom aan het indelingstabblad toegevoegd.	Experimenteren met NEAT	100	320	60	5.340	4.905	5.420	261.08 uur	188.78%	
25/11/2020	Gewerkt aan 'V2 Algoritmes en AI' (doc). Enkele checks uitgevoerd en hoofdstukjes op 'groen' gezet.	Oplossing proberen te vinden, gepaard over PWS	160	120	140	5.500	5.025	5.560	268.08 uur	111.70%	
26/11/2020	Gewerkt aan 'V2 Algoritmes en AI' (doc) en aan 'Gautomatiseerde systemen aandelenhandel' (doc).	Gesprek. Aanpassingen gemaakt aan 'V2 Algoritmes en AI' (doc). Gewerkt aan GUI.	80	100	5.580	5.025	5.660	271.08 uur	112.95%		
27/11/2020	Gesprekje met Sander over het verdere verloop van het PWS. Enkele huiselijke klusjes gedaan. Gewerkt aan hoofdstuk 'V2 Algoritmes en AI' (doc).	Oplossing voor het probleem met NEAT gevonden, ik kan verder. Gewerkt aan het programma (GitHub)	180	120	140	5.760	5.145	5.800	278.41 uur	116.01%	
28/11/2020	Gewerkt aan de GUI van het programma. (GitHub)		50	5.760	5.145	5.850	279.25 uur	116.35%	Alles werkt nu voldoend behalve wat in de rode gedeeltes staan in diese screenshots. (Settings ga ik geen moeite voor doen)		

Datum	Stach	Sander	Noah	Omschrijving						Opmerkingen / Problemen / (Nieuwe) doelen					
				Tijd besteed (minuten)	Totalle tijd (minuten)	Alles bij elkaar	% klaar	Stach	Noah	Sander	Stach	Noah	Sander	Stach	Noah
29/11/2020	Gewerkt aan 'V2 Algoritmes en AI' (doc). Sander heeft me bijgepraat over de GUI-functionaliteit. Gewerkt aan 'Aandelen geavanceerd' (doc).	Gewerkt aan het programma (GitHub).	Gewerkt aan V2 Algoritmes en AI' (doc). Gewerkt aan GUI (GitHub). Stach bijgepraat over de GUI functionaliteit.	480	540	160	6.240	5.685	6.010	298.91 uur	124.55%	Wanneer ga ik hem langer trainer en klinkt of ik hem misschien meer data wil geven. Ook da ik proberen uit te vinden hoeveel training nodig is en is er een manier van trainen wel ideaal is.	@(Stach - vandaag gepusht) Data van de acties van de winnende AI kan nu als grafiek uitgezet worden. Verbeteringen in programma + hertraining. Is het misschien beter om de AI te trainen op vóór het afgelopen jaar? Want het ging dit jaar niet zo geweldig met de markt.		
30/11/2020	Gewerkt aan 'Aandelen geavanceerd' (doc).	Gewerkt aan het programma (GitHub).	Gewerkt aan de GUI van het programma. (GitHub)	30	300	60	6.270	5.985	6.070	305.41 uur	127.26%	@Noah, lekker bezig. Trainen op data voor dit jaar is waarschijnlijk representatiever, maar ik ben ook ontzettend benieuwd hoe 'ie met de black swan coronacrisis' omgaat.			
01/12/2020	Gesprek met Sander over de GUI en de manier waarop we de GUI gaan beschrijven in het verslag. Flowchart (doc) gemaakt.	Gesprek met Stach over de GUI en de manier waarop we de GUI gaan beschrijven in het verslag. Flowchart (doc) gemaakt. Gewerkt aan GUI van het programma. (GitHub)	Gesprek met Stach over de GUI en de manier waarop we de GUI gaan beschrijven in het verslag. Flowchart (doc) gemaakt. Gewerkt aan GUI van het programma. (GitHub)	90	150	6.360	5.985	6.220	309.41 uur	128.92%	De flowchart gaan we gebruiken om de structuur van onze bestanden aan te geven. We moeten de AI delen van Noah erg nog aan tevoegegen.				
02/12/2020	Gewerkt aan 'Aandelen geavanceerd' (doc). Gewerkt aan de inleiding (doc).	Gewerkt aan de GUI van het programma. (GitHub)	Gewerkt aan de GUI van het programma. (GitHub)	150	110	6.510	5.985	6.330	313.75 uur	130.73%					
03/12/2020	Gewerkt aan 'Aandelen geavanceerd' (doc).	Gewerkt aan het programma (GitHub)	Gewerkt aan de GUI van het programma. (GitHub)	60	200	6.570	6.185	6.530	321.41 uur	133.92%					
04/12/2020	Gewerkt aan 'Geautomatiseerde systemen in de aardelenhandel' (doc). Gewerkt aan 'Aandelen algemeen' (doc). Gewerkt aan hoofdstuk NEAT' (doc).	Gewerkt aan de GUI van het programma. (GitHub)	Gewerkt aan de GUI van het programma. (GitHub)	240	200	6.810	6.185	6.730	328.75 uur	136.98%	Hoofdstukje 5 af. @Sander: lekker bezig man!				
05/12/2020	Gewerkt aan 'Aandelen geavanceerd' (doc). Gesprek met Noah over opties & het programma.	Gewerkt aan het programma (GitHub)	Gewerkt aan de GUI van het programma. (GitHub)	95	300	200	6.905	6.485	6.930	338.66 uur	141.11%	Hoofdstuk 'Aandelen geavanceerd' is nu ook af, staart op eindeditie, feedback gewenst.			
06/12/2020	Laatste checks uitgevoerd voor hoofdstukken 'Voorwoord', 'Aandelen geavanceerd en Computersystemen' en 'Computersystemen'. Gewerkt aan 'Aandelenhandel'. Gesprek met aandelenhandel' (doc). En aan 'Algoritmes en AI' (doc).	Document met vragen beantwoord.	Gewerkt aan de GUI van het programma. (GitHub)	120	90	230	7.025	6.575	7.160	346 uur	144.17%				
07/12/2020	Gewerkt aan Aandelen algemeen' (doc). En aan 'Algoritmes en AI' (doc).	Vragen op document met vragen beantwoord. Gewerkt aan het programma (GitHub).	Gewerkt aan de GUI van het programma. (GitHub)	120	300	90	7.145	6.875	7.250	354.5 uur	147.71%				
08/12/2020	Deadline: vanaf hier is alle tekst af. Nu alleen nog focussen op opmaak, bronnen, formules etc.						7.145	6.875	7.250	354.5 uur	147.71%	Sander gaat ervoor zorgen dat alle formules goed in Word komen en Stach neemt de bronnen voor zijn rekening.			
08/12/2020	Lijstje (doc) opgesteld met tickers die we in de conclusie gaan verwerken. Gewerkt aan hoofdstuk 'Aandelen algemeen' (doc).	Gewerkt aan de GUI van het programma (GitHub)	Start gemaakt aan Word document. Gewerkt aan Word document (GitHub).	120	100	7.265	6.875	7.350	358.16 uur	149.24%	Noah gaat met dit lijstje aan de slag.	Planning tot het einde opgesteld. Woendag: code af. AI resultaten binnenvoeren. Donderdag: laatste hoofdstuk af. Vrijdag: conclusie af. Zaterdag: laatste checks. Zondag: naar Word. Maandag: printen, puntjes op de i.			
09/12/2020	Diverse kleine handelingen verricht. Eindeditie gedaan voor hoofdstuk Algoritmes en AI' (doc). Laatste checks gedaan voor hoofdstuk 'aandelen algemeen' (doc).	Gewerkt aan het programma - Af. Tickers getraind waar Stach me om vroeg.	200	150	180	7.465	7.025	7.530	367 uur	152.92%		Ik heb voor ons onderzoek nu een aantal grafieken van de lijst van Stach gemaakt. Ik heb de AI gefineertuned naar betere instellingen na veel trial and error.			
09/12/2020	Voortgangsgesprek			50	7.515	7.075	7.580	369.5 uur	153.96%				Ik ga morgen mijn deel van de praktijk volledig afronden, ik zie niet hoe dat in de weg komt met de planning hierboven.		
10/12/2020	Gewerkt aan de inleiding (doc). Gewerkt aan hoofdstuk Algoritmes en AI' (doc). Gewerkt aan 'evaluaties en conclusies' (doc).	Gewerkt aan 'Praktijk' (doc). Voor de missende tickers AI-waarden genereerd.	360	220	280	7.895	7.295	7.860	384.16 uur	160.87%					
11/12/2020	Laatste checks uitgevoerd voor de inleiding (doc). En aan 'Algoritmes en AI' (doc). Gewerkt aan de conclusie (doc) en aan de Word-versie (GitHub).	Gewerkt aan Word versie (GitHub). Gewerkt aan 'Praktijk' (doc).	300	180	170	8.195	7.475	8.030	395 uur	164.58%		@Stach: Grafiek van GME gestuurd, was die vergeten te uploaden.			

Datum	Stach	Noah	Sander	Tijd besteed (minuten)	Total time (minutes)	Alles bij elkaar	Opmerkingen / Problemen / (Nieuwe) doelen
				Stach	Noah	Totaal	
12/12/2020	Gewerkt aan de conclusie (doc) en aan praktijk: zo werkt ons product (doc) en aan 'NEAT' (doc). Laatste check uitgevoerd voor eva en conclu.. Gewerkt aan 'programmeerbiedigheden' (doc). Laatste checks NEAT gedaan (doc). Gewerkt aan het Worddocument. Alle andere laatste checks ook gedaan. Op de bijlagen na staat nu alles in Word.	Vragen beenwoord voor Stach, aantal hoofdstukken op aanvraag bekennen en feedback geven waar nodig.	Gewerkt aan 'Praktijk' (doc). Gewerkt aan 'Programmeerbiedigheden' (doc). Gewerkt aan Word document.	260	60	220	8.455 7.535 8.250 404 uur
13/12/2020	Laatste checks, puntjes op de i, printen, etc.	Laatste checks, puntjes op de i, printen, etc.	Gewerkt aan het Word document.	360	200	8.815 7.535 8.450 413.33 uur	168.33%
14/12/2020	Deadline: werkstuk inleveren	Deadline: werkstuk inleveren	Laatste checks, puntjes op de i, printen, etc.	210	210	9.025 7.535 8.660 420.33 uur	172.22%
15/12/2020							Voor 9:00 uur bij meneer Van Wingen.

Appendix II: tracksheet

Zie volgende 3 pagina's.

Uitleg per kolom:

- **Onderwerp:** In deze kolom staat het hoofdstuk met bijbehorende deelonderwerpen;
- **Status:** In deze kolom kan worden bijgehouden hoever we zijn met elk hoofdstuk.
Opties zijn: Niet gestart, Onderzoek gestart, Werk nodig, Concept klaar, Eindredactie en Afgerond.
- **Feedback:** De feedbackkolom is een manier om aan elkaar te laten zien of wij de inhoud van het onderwerp goed vinden. De letters boven aan staan voor onze initialen. Een kruis in dit vak betekent dat die persoon geen commentaar hoeft te geven omdat het zijn eigen stuk is. Oranje betekent dat er nog commentaar is en groen staat voor geen commentaar.
- **Verwerking:** Hier zijn twee kolommen. De kolom 'Laatste check' staat voor een definitieve laatste check voordat de tekst naar Word werd overgezet. De kolom 'Naar Word' kan verschillende kleuren hebben per hoofdstuk. Oranje betekent dat het wel al in dit document staat, maar er nog tekstuele aanpassingen moeten gedaan. Geel betekent dat de voetnoten en bronnen in orde gemaakt moeten worden. Paars betekent dat al het bovenstaande klaar is, maar dat alleen nog nummering voor formules en figuren moet worden geplaatst en één laatste controle op spelfouten en mogelijke problemen die ontstaan zijn tijdens het overzetten moet worden gedaan. Groen betekent: alles is klaar.
- **Toegewezen:** Dit zijn de personen die werken aan het hoofdstuk of deelonderwerp
- **Link:** Dit is een verwijzing naar het bijbehorende Google-Docs-document.
- **Opmerkingen:** Dit zijn opmerkingen over het hoofdstuk/deelonderwerp. Sommige opmerkingen zijn weggehaald.

Merk op dat in deze *tracksheet* het daadwerkelijke programmeren niet is opgenomen.

Onderwerp		Status	Feedback	Verwerking	Toegewezent	Link	Opmerkingen
SR	NV	MV	Laatste check	Naar Word			
1 Voorwoord		Afgerond	x	6-12	Stach	Document	
Motivatie + minstukje inhoud + kleine terugblik			x				
2 Inleiding			x				
Intressener lezer, introductie onderwerp, relevantie			x				
Onderzoeksraag			x				
Hypothese			x				
Oppbouw van ons theoretiedel			x				
Oppbouw van ons praktijksdeel			x				
Topie van de ijsberg			x				
Disclaimer			x				
3 Aandelen algemeen							
Geschiedenis			x				
IPO's en de beursgang			x				
De functie van beurzen			x				
Tradens of beleggen? + dividend			x				
- Fliishandel & FlowTraders			x				
De waarde van aandelen			x				
Indices			x				
Volatiliteit & volatiliteitsindices			x				
Risico-analyse: aandelen in vergelijking met forex			x				
4 Aandelen geavanceerd							
Introductie			x				
De fundamentele versus de technische analyse			x				
Interpreteren van grafiekjes			x				
- De onderdelen van een koersgrafiek			x				
- Trends en trendlijnen			x				
- Patronen			x				
- Indicatoren & oscillatoren			x				
-- Voortschrijdend gemiddelde			x				
-- Moving average convergence divergence-indicator (MACD)			x				
-- Relatieve Sterkte Index (RSI)			x				
- Relevantie?			x				
Options & Futures			x				
Verschillende posities			x				
5 Computersystemen aandelenhandel							
Conditional orders			x				
Expert advisor (EA)			x				
Artificiële intelligentie (AI)			x				
- Waaron AI niet vaak gebruikt wordt, maar wij het wel proberen			x				
6 Algoritmes en AI - algemeen							
Introductie			x				
Trainen en trainingsdata			x				
Oppbouw van een neuraal netwerk			x				
- Input: dit goedien we een			x				

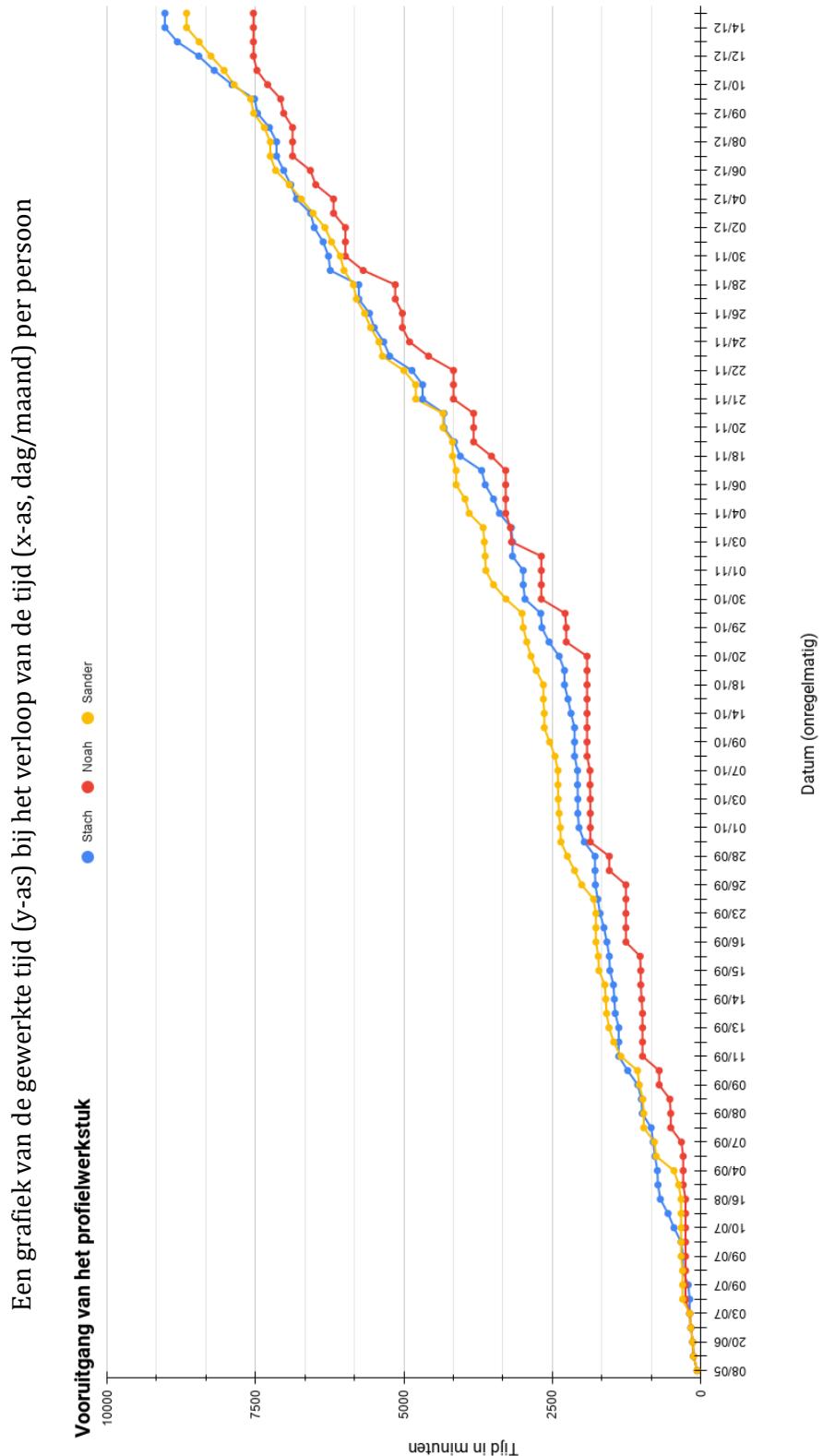
(Stach): **Deze** afspeellijst kan dienen als goede documentatie voor dit onderwerp. Neurale netwerken kunnen verschillend en verschillend. Het wordt dan voorvan uiteraard die manier van uitvoeren.

Onderwerp	Status	Feedback	Verwerking	Toegewezien	Link	Opmerkingen
		SR NV SV MV	Laatste check	Naar Word		
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias		x	x			
- Activatiefunctie: de invoer verandert naar een uitvoer	Afgerond	x	x	11-12	Sander	
-- Het belang van activatiefuncties		x	x		Sander & Stach	
- Enkele laag: samen staan neuronen sterk		x	x		Sander	
- Tussentlagen: dit is waar de magie gebeurt		x	x		Sander & Stach	
- De laatste laag: hier neemt het neurale netwerk zijn beslissing		x	x		Sander	
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk		x	x		Sander & Stach	
- Backpropagation: een neuraal netwerk leert van zijn fouten		x	x		Sander & Stach	
- De wiskunde achter backpropagation		x	x		Sander & Stach	
Conclusie/samenvatting		x	x			
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	
Loops		x	x			
Electron		x	x			
9 Praktijk: zo werkt ons product		x	x			
Introductie & overzicht		x	x			
AI-gedeelte		x	x			
- Gedefinieerde variabelen		x	x			
- Trainingsfase I		x	x			
- Trainingsfase II		x	x			
- Testfase		x	x			
- NEAT-configuratie		x	x			
GUI-gedeelte		x	x			
- Rood: de balance		x	x			
- Oranje: de tijd		x	x			
Geel (1): de vormgeving van de grafiek		x	x			
- Even tussendoor: de backend		x	x			
- Lichtblauw: de tickers op een rij		x	x			
Lichtgroen: voeg een ticker toe		x	x			
- Donkerblauw: naam, waarde, geldigheid en tijd		x	x			
- Geel (2): de inhoud van de grafiek		x	x			
- Paars (1): de knoppen boven de grafiek		x	x			
- Paars (2): de backend van de knoppen boven de grafiek		x	x			
- Roze: indicatoren		x	x			
Afsluiting		x	x			
worden tegelijkertijd een document en een uitleg. Hierdoor staat voor elkaars uitleg een manier van omgaan die voor ons ook interessant.						
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias						
- Activatiefunctie: de invoer verandert naar een uitvoer						
-- Het belang van activatiefuncties						
- Enkele laag: samen staan neuronen sterk						
- Tussentlagen: dit is waar de magie gebeurt						
- De laatste laag: hier neemt het neurale netwerk zijn beslissing						
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk						
- Backpropagation: een neuraal netwerk leert van zijn fouten						
- De wiskunde achter backpropagation						
Conclusie/samenvatting						
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	
Loops		x	x			
Electron		x	x			
9 Praktijk: zo werkt ons product		x	x			
Introductie & overzicht		x	x			
AI-gedeelte		x	x			
- Gedefinieerde variabelen		x	x			
- Trainingsfase I		x	x			
- Trainingsfase II		x	x			
- Testfase		x	x			
- NEAT-configuratie		x	x			
GUI-gedeelte		x	x			
- Rood: de balance		x	x			
- Oranje: de tijd		x	x			
Geel (1): de vormgeving van de grafiek		x	x			
- Even tussendoor: de backend		x	x			
- Lichtblauw: de tickers op een rij		x	x			
Lichtgroen: voeg een ticker toe		x	x			
- Donkerblauw: naam, waarde, geldigheid en tijd		x	x			
- Geel (2): de inhoud van de grafiek		x	x			
- Paars (1): de knoppen boven de grafiek		x	x			
- Paars (2): de backend van de knoppen boven de grafiek		x	x			
- Roze: indicatoren		x	x			
Afsluiting		x	x			
worden tegelijkertijd een document en een uitleg. Hierdoor staat voor elkaars uitleg een manier van omgaan die voor ons ook interessant.						
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias						
- Activatiefunctie: de invoer verandert naar een uitvoer						
-- Het belang van activatiefuncties						
- Enkele laag: samen staan neuronen sterk						
- Tussentlagen: dit is waar de magie gebeurt						
- De laatste laag: hier neemt het neurale netwerk zijn beslissing						
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk						
- Backpropagation: een neuraal netwerk leert van zijn fouten						
- De wiskunde achter backpropagation						
Conclusie/samenvatting						
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	
Loops		x	x			
Electron		x	x			
9 Praktijk: zo werkt ons product		x	x			
Introductie & overzicht		x	x			
AI-gedeelte		x	x			
- Gedefinieerde variabelen		x	x			
- Trainingsfase I		x	x			
- Trainingsfase II		x	x			
- Testfase		x	x			
- NEAT-configuratie		x	x			
GUI-gedeelte		x	x			
- Rood: de balance		x	x			
- Oranje: de tijd		x	x			
Geel (1): de vormgeving van de grafiek		x	x			
- Even tussendoor: de backend		x	x			
- Lichtblauw: de tickers op een rij		x	x			
Lichtgroen: voeg een ticker toe		x	x			
- Donkerblauw: naam, waarde, geldigheid en tijd		x	x			
- Geel (2): de inhoud van de grafiek		x	x			
- Paars (1): de knoppen boven de grafiek		x	x			
- Paars (2): de backend van de knoppen boven de grafiek		x	x			
- Roze: indicatoren		x	x			
Afsluiting		x	x			
worden tegelijkertijd een document en een uitleg. Hierdoor staat voor elkaars uitleg een manier van omgaan die voor ons ook interessant.						
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias						
- Activatiefunctie: de invoer verandert naar een uitvoer						
-- Het belang van activatiefuncties						
- Enkele laag: samen staan neuronen sterk						
- Tussentlagen: dit is waar de magie gebeurt						
- De laatste laag: hier neemt het neurale netwerk zijn beslissing						
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk						
- Backpropagation: een neuraal netwerk leert van zijn fouten						
- De wiskunde achter backpropagation						
Conclusie/samenvatting						
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	
Loops		x	x			
Electron		x	x			
9 Praktijk: zo werkt ons product		x	x			
Introductie & overzicht		x	x			
AI-gedeelte		x	x			
- Gedefinieerde variabelen		x	x			
- Trainingsfase I		x	x			
- Trainingsfase II		x	x			
- Testfase		x	x			
- NEAT-configuratie		x	x			
GUI-gedeelte		x	x			
- Rood: de balance		x	x			
- Oranje: de tijd		x	x			
Geel (1): de vormgeving van de grafiek		x	x			
- Even tussendoor: de backend		x	x			
- Lichtblauw: de tickers op een rij		x	x			
Lichtgroen: voeg een ticker toe		x	x			
- Donkerblauw: naam, waarde, geldigheid en tijd		x	x			
- Geel (2): de inhoud van de grafiek		x	x			
- Paars (1): de knoppen boven de grafiek		x	x			
- Paars (2): de backend van de knoppen boven de grafiek		x	x			
- Roze: indicatoren		x	x			
Afsluiting		x	x			
worden tegelijkertijd een document en een uitleg. Hierdoor staat voor elkaars uitleg een manier van omgaan die voor ons ook interessant.						
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias						
- Activatiefunctie: de invoer verandert naar een uitvoer						
-- Het belang van activatiefuncties						
- Enkele laag: samen staan neuronen sterk						
- Tussentlagen: dit is waar de magie gebeurt						
- De laatste laag: hier neemt het neurale netwerk zijn beslissing						
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk						
- Backpropagation: een neuraal netwerk leert van zijn fouten						
- De wiskunde achter backpropagation						
Conclusie/samenvatting						
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	
Loops		x	x			
Electron		x	x			
9 Praktijk: zo werkt ons product		x	x			
Introductie & overzicht		x	x			
AI-gedeelte		x	x			
- Gedefinieerde variabelen		x	x			
- Trainingsfase I		x	x			
- Trainingsfase II		x	x			
- Testfase		x	x			
- NEAT-configuratie		x	x			
GUI-gedeelte		x	x			
- Rood: de balance		x	x			
- Oranje: de tijd		x	x			
Geel (1): de vormgeving van de grafiek		x	x			
- Even tussendoor: de backend		x	x			
- Lichtblauw: de tickers op een rij		x	x			
Lichtgroen: voeg een ticker toe		x	x			
- Donkerblauw: naam, waarde, geldigheid en tijd		x	x			
- Geel (2): de inhoud van de grafiek		x	x			
- Paars (1): de knoppen boven de grafiek		x	x			
- Paars (2): de backend van de knoppen boven de grafiek		x	x			
- Roze: indicatoren		x	x			
Afsluiting		x	x			
worden tegelijkertijd een document en een uitleg. Hierdoor staat voor elkaars uitleg een manier van omgaan die voor ons ook interessant.						
- Neuron: de kleinste eenheid van het netwerk						
-- Functie van het gewicht en de bias						
- Activatiefunctie: de invoer verandert naar een uitvoer						
-- Het belang van activatiefuncties						
- Enkele laag: samen staan neuronen sterk						
- Tussentlagen: dit is waar de magie gebeurt						
- De laatste laag: hier neemt het neurale netwerk zijn beslissing						
- Verliestfunctie & gradient descent: zo leert een neuraal netwerk						
- Backpropagation: een neuraal netwerk leert van zijn fouten						
- De wiskunde achter backpropagation						
Conclusie/samenvatting						
7 NEAT						
Inleidend NEAT-stukje		x	x	13-12	Stach	
De versimpelde werking van NEAT	Afgerond	x	x		Sander	
De wiskunde en biologie achter NEAT		x	x		Stach & Sander	
Een evolutionair algoritme in plaats van gradient descend?		x	x		Stach	
8 Programmeerbenodigdheden		x	x	13-12	Sander & Noah	
Introductie		x	x		Stach	
Python		x	x		Sander & Noah	
- Libraries		x	x		Stach, Noah & Sander	
Classes en variabelen		x	x		Noah	

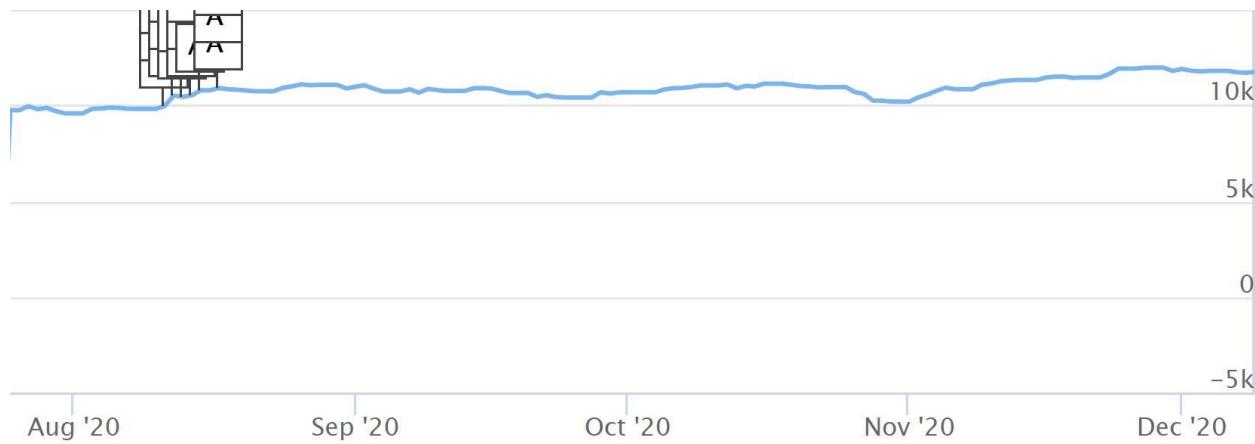
Onderwerp	Status	Feedback	Verwerking	Toegewezen	Link	Opmerkingen
		SR NV SV MV	Laatste check	Naar Word		
10 Evaluatie en conclusies						
Introductie		x				
Vergeschiedenis		x				
Resultaten		x				
- 1 januari 2019 - 1 januari 2020 (voor corona)		x				
-- EUR/USD-pair (EURUSD=X)		x				
-- AEX-index (^AEX)		x				
-- Microsoft (MSFT)		x				
-- GameStop (GME)		x				
-- VIX-index (^VIX)		x				
-- Just Eat Takeaway (TKWY.AS)		x				
- 1 januari 2020 - nu (met corona)		x				
-- EUR-USD pair (EURUSD=X)		x				
-- Ahold Delhaize (ADRN.Y)		x				
-- Tesla (TSLA)		x				
-- Microsoft (MSFT)		x				
-- Shell (RYDAF)		x				
-- Air France-KLM (AF.PA)		x				
-- VIX-index (^VIX)		x				
-- Just Eat Takeaway (TKWY.AS)		x				
Conclusie		x				
- Hamvraag: is het beter dan een ... ?		x				
- Bespreken maatschappelijke impact		x				
Discussie		x				
- Terugblik op de hypothese		x				
- Validiteit en betrouwbaarheid		x				
Aanbevelingen voor vervolgonderzoek		x				
11 Bronnenlijst						
Bronnenlijst in APA-formaat		Afgerond	x	x	14-12	Stach, Noah & Sander
12 Appendices						
I Logboek						
II Taakverdeling						
III Prestaties						
IV Compleet overzicht van alle transacties van Stach en een grafiek met zijn rendement over de tijd						
V Grafieken hoofdstuk 3						
VI Grafieken hoofdstuk 4						
VII Vergrote versie soorten neurale netwerken						
IX Meer screenshots van ons programma						

Appendix III: prestatiegrafiek

In deze grafiek is de gewerkte tijd in minuten van Stach, Noah en Sander uitgezet over de tijd.



Appendix IV: compleet overzicht van alle transacties van Stach en een grafiek met zijn rendement over de tijd



Een grafiek met het totale saldo van Stachs virtuele beleggingspotje.

40 Transacties

Datum	Type	Aandeel	Volume	Koers	Totaal
17 aug	Aankoop effecten	ABN AMRO BANK N.V.	130	8,104	-1.053,52
17 aug	Aankoop effecten	PayPal Holdings	7	161,650	-1.131,55
17 aug	Aankoop effecten	AEX	2	560,510	-1.121,02
17 aug	Aankoop effecten	Microsoft Corp	4	176,374	-705,50
17 aug	Verkoop effecten	AEX Volatility Index	170	23,632	+4.017,44
17 aug	Verkoop effecten	ING Groep	173	6,946	+1.201,66
17 aug	Aankoop effecten	ING Groep	174	6,946	-1.208,60
17 aug	Aankoop effecten	ING Groep	80	6,944	-565,52
17 aug	Aankoop effecten	AEX Volatility Index	170	23,500	-3.995,00
17 aug	Verkoop effecten	Flow Traders	134	33,860	+4.537,24
15 aug	Aankoop effecten	Flow Traders	134	31,500	-4.221,00
14 aug	Verkoop effecten	Flow Traders	127	33,240	+4.221,48
14 aug	Aankoop effecten	JUST EAT TAKEAWAY	19	100,500	-1.909,50
14 aug	Verkoop effecten	Heineken	24	79,300	+1.903,20
14 aug	Aankoop effecten	Flow Traders	127	32,060	-4.074,16
14 aug	Verkoop effecten	AEX	7	569,950	+3.989,65
13 aug	Aankoop effecten	AEX	7	569,950	-3.989,65
13 aug	Verkoop effecten	Tesla	3	1.319,525	+3.958,57
12 aug	Aankoop effecten	Microsoft Corp	2	173,219	-346,44
12 aug	Verkoop effecten	AEX	7	570,520	+3.993,64
12 aug	Aankoop effecten	Tesla	3	1.170,568	-3.511,70
11 aug	Aankoop effecten	ING Groep	73	7,108	-518,88
11 aug	Aankoop effecten	AMX	3	814,480	-2.443,44
11 aug	Verkoop effecten	Tesla	2	1.208,338	+2.416,68
11 aug	Aankoop effecten	AEX	2	568,470	-1.136,94
11 aug	Verkoop effecten	AMX	2	814,650	+1.629,30
11 aug	Aankoop effecten	Heineken	24	77,320	-1.855,68
11 aug	Aankoop effecten	Sligro Food Group	71	13,960	-991,16
11 aug	Dividend	AMX			+2,00
11 aug	Dividend	AEX			+5,00
11 aug	Dividend	Tesla			+2,00
11 aug	Dividend	ING Groep			+21,00
11 aug	Verkoop effecten	Citi S&P 500 Indicatie	1	2.851,927	+2.851,93
11 aug	Dividend	Citi S&P 500 Indicatie			+1,00
14 jul	Aankoop effecten	Citi S&P 500 Indicatie	1	2.773,656	-2.773,66
14 jul	Aankoop effecten	ING Groep	21	6,430	-125,03
14 jul	Aankoop effecten	AMX	2	772,300	-1.544,60
14 jul	Aankoop effecten	AEX	5	578,850	-2.894,25
14 jul	Aankoop effecten	Tesla	2	1.316,799	-2.633,60
13 jul	Banktransactie				+10.000,00

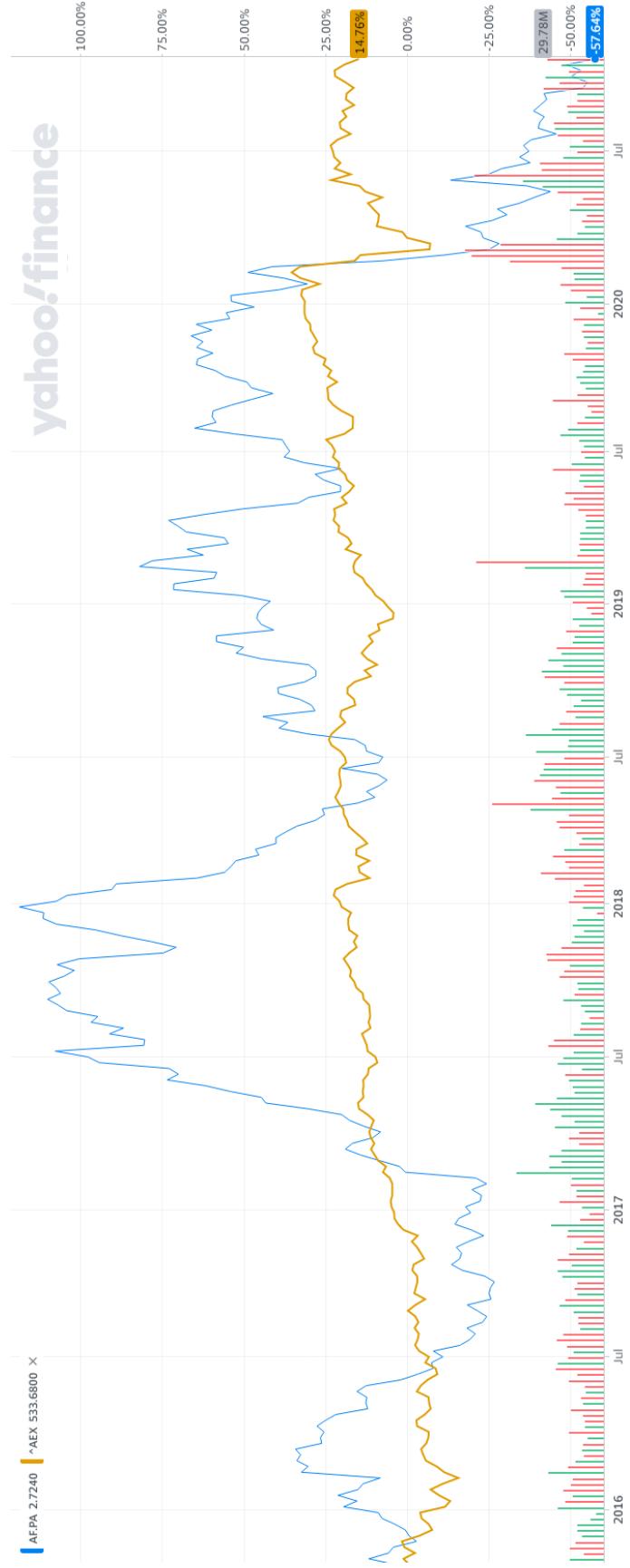
40 Transacties

De virtuele beleggingen van Stach.

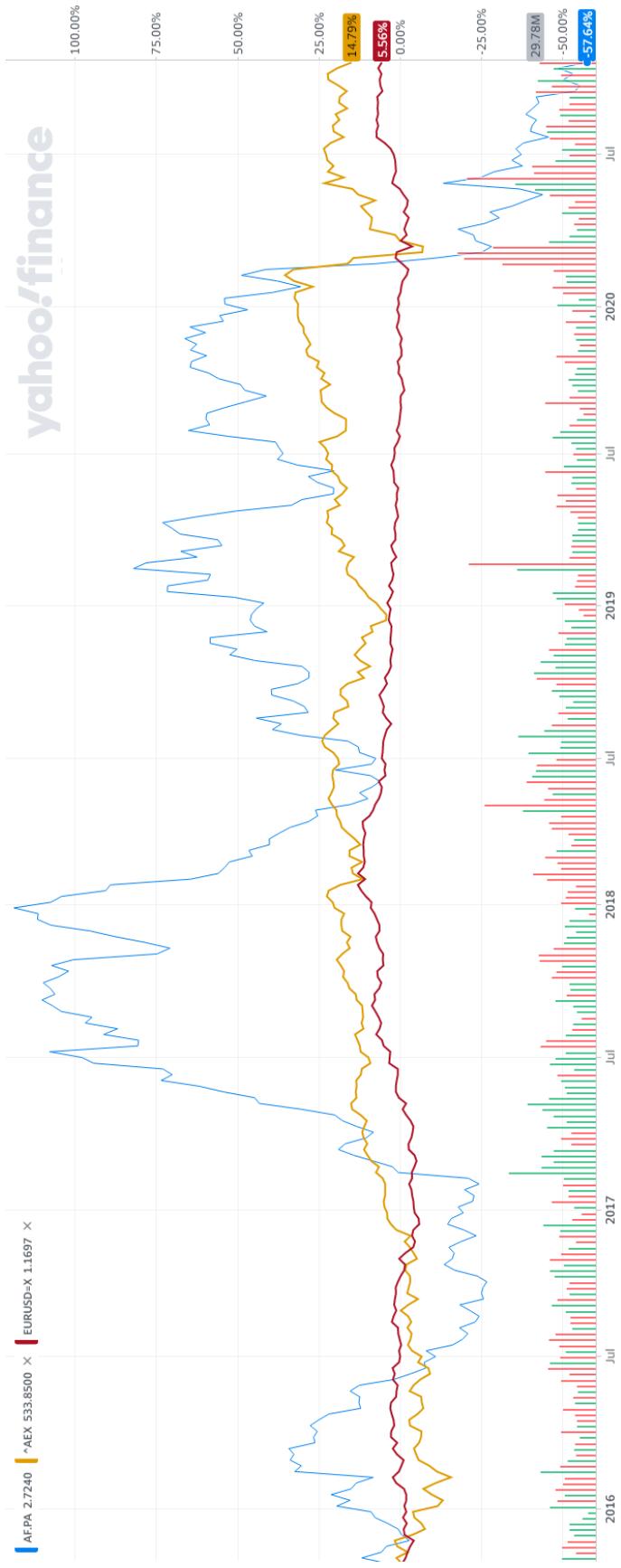
Appendix V: vergrote grafieken hoofdstuk 1

- Pagina 164: Waarde van de AEX-index en de prijs van het aandeel Air France-KLM
- Pagina 165: Waarde van de AEX-index, de prijs van het aandeel Air France-KLM en de waarde van het EUR/USD-pair.

Deze grafiek toont de waarde van de AEX-index (geel) en de prijs van een aandeel Air France-KLM (blauw). Bron gegevens: (Yahoo Finance, 2020).

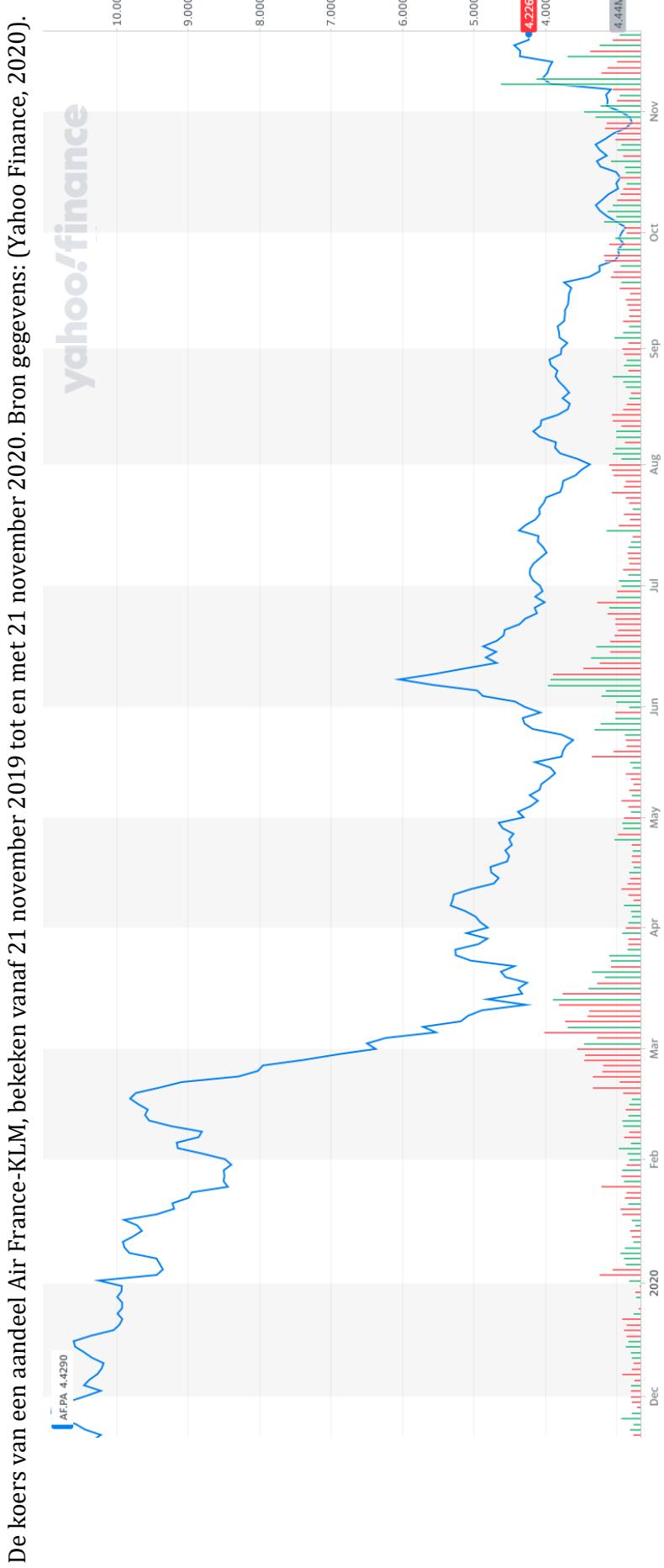


Deze grafiek toont de waarde van de AEX-index (geel), de prijs van een aandeel Air France-KLM (blauw) en de waarde van het EUR/USD-pair (rood).
Bron gegevens: (Yahoo Finance, 2020).

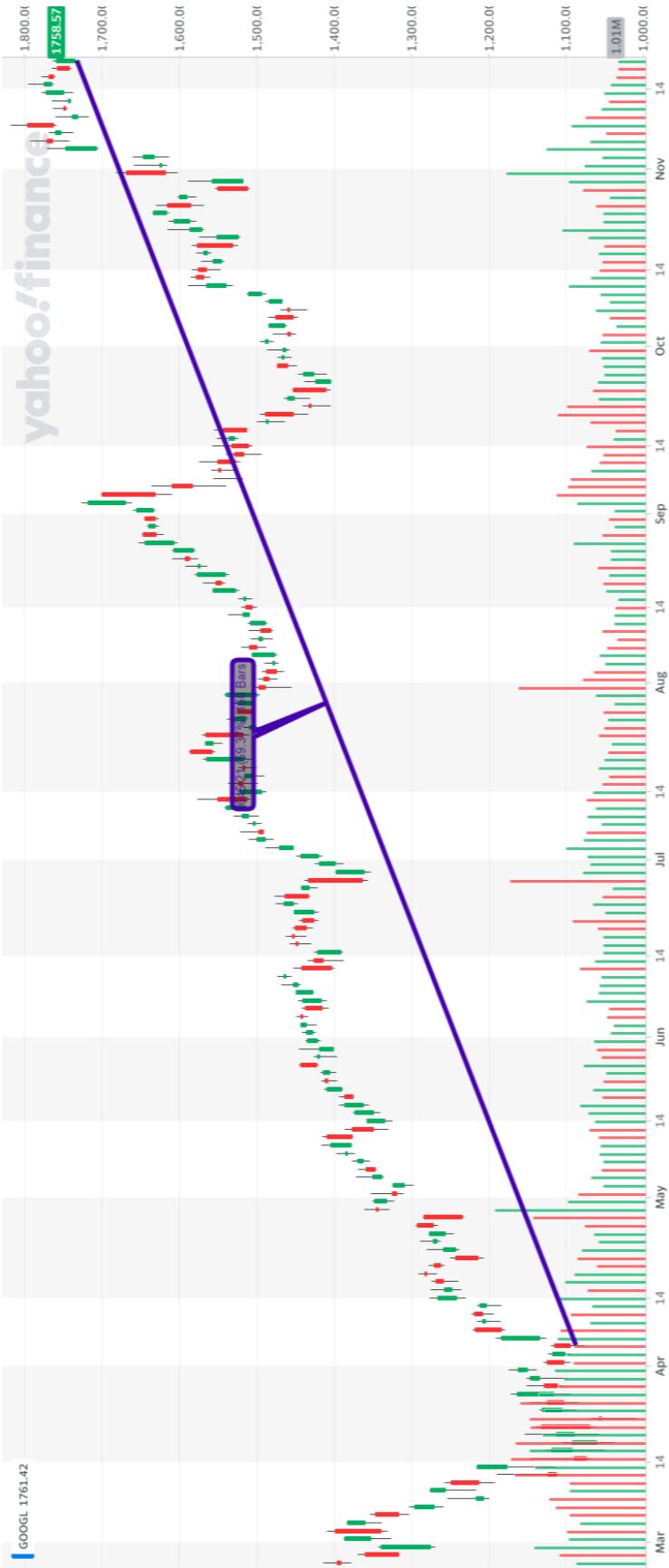


Appendix VI: vergrote grafieken hoofdstuk 2

- Pagina 167: De koers van een aandeel Air France-KLM (21 november 2019 t/m 21 november 2020)
- Pagina 168: De koers van een aandeel in Alphabet met een trendlijn (maart 2020 t/m november 2020)
- Pagina 169: De koers van een aandeel Air France-KLM met een trendlijn (september 2019 t/m november 2020)
- Pagina 170: Koersgrafiek van een computerbedrijf Intel met quadrant lines. (eind 2019 tot eind 2020)
- Pagina 171: Grafiek van de koers van een aandeel Air France-KLM met een SMA-indicatorlijn. (december 2019 t/m december 2020)
- Pagina 172: Grafiek van de koers van een aandeel Air France-KLM met een SMA-indicatorlijn met onzekerheidsmarge. (december 2019 t/m december 2020)
- Pagina 173: Grafiek van de koer van een aandeel Air-France KLM met een SMA- en een EMA-indicatorlijn. (december 2019 t/m december 2020)
- Pagina 174: Grafiek van de koers van een aandeel Ari France-KLM met een MACD-indicatorlijn. (december 2019 t/m december 2020)
- Pagina 175: Grafiek van de koers van een aandeel Air France-KLM met een RSI(14,C)-indicator. (december 2019 t/m december 2020)



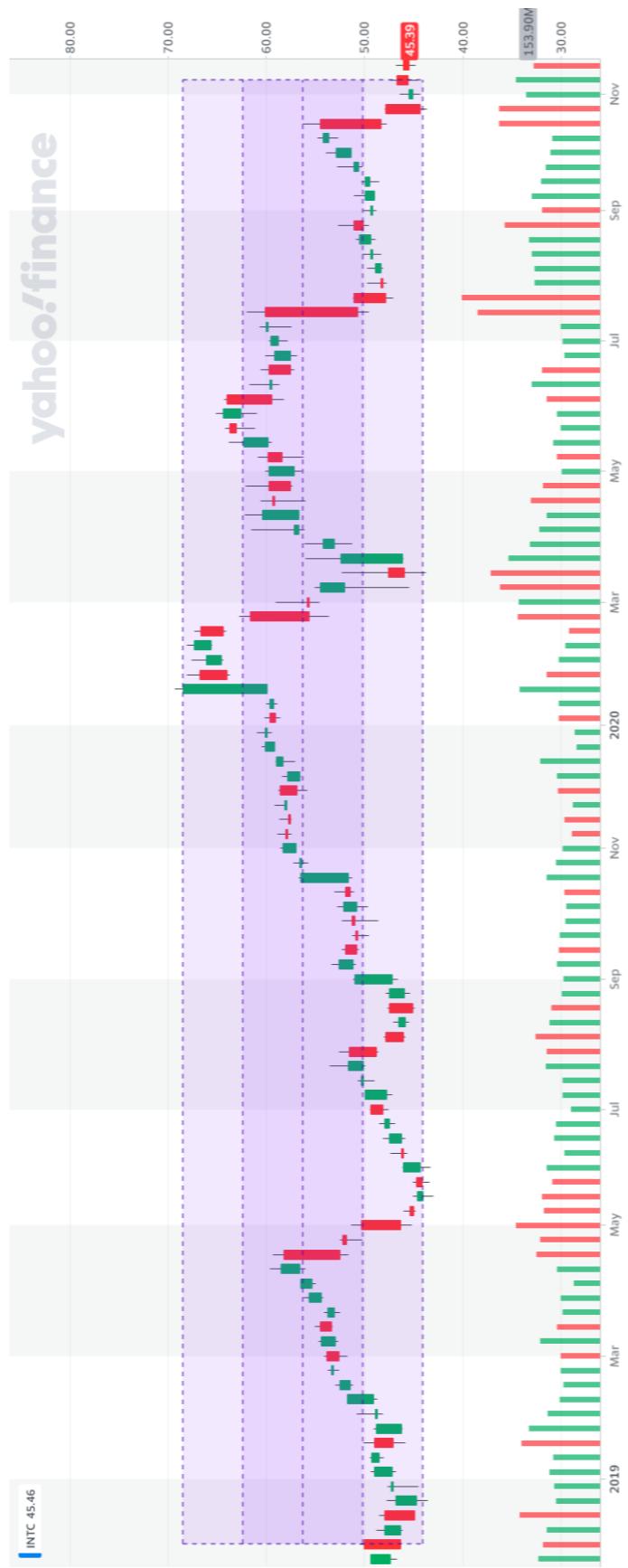
De koers van een aandeel in Alphabet, van maart 2020 tot en met november 2020. In de grafiek is een trendlijn getekend. Bron gegevens: (Yahoo Finance, 2020).



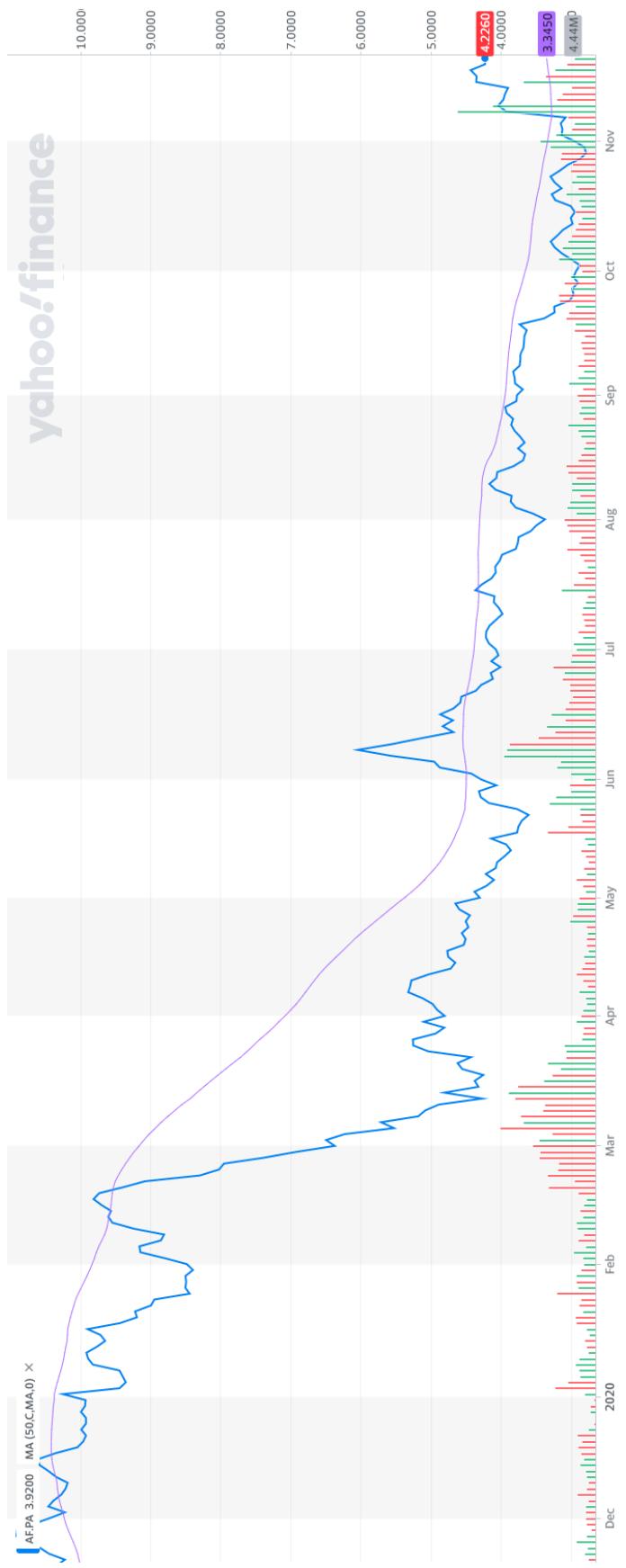
De koers van een aandeel Air France-KLM van september 2019 tot en met november 2020. In de grafiek is een trendlijn getekend.
Bron gegevens: (Yahoo Finance, 2020).



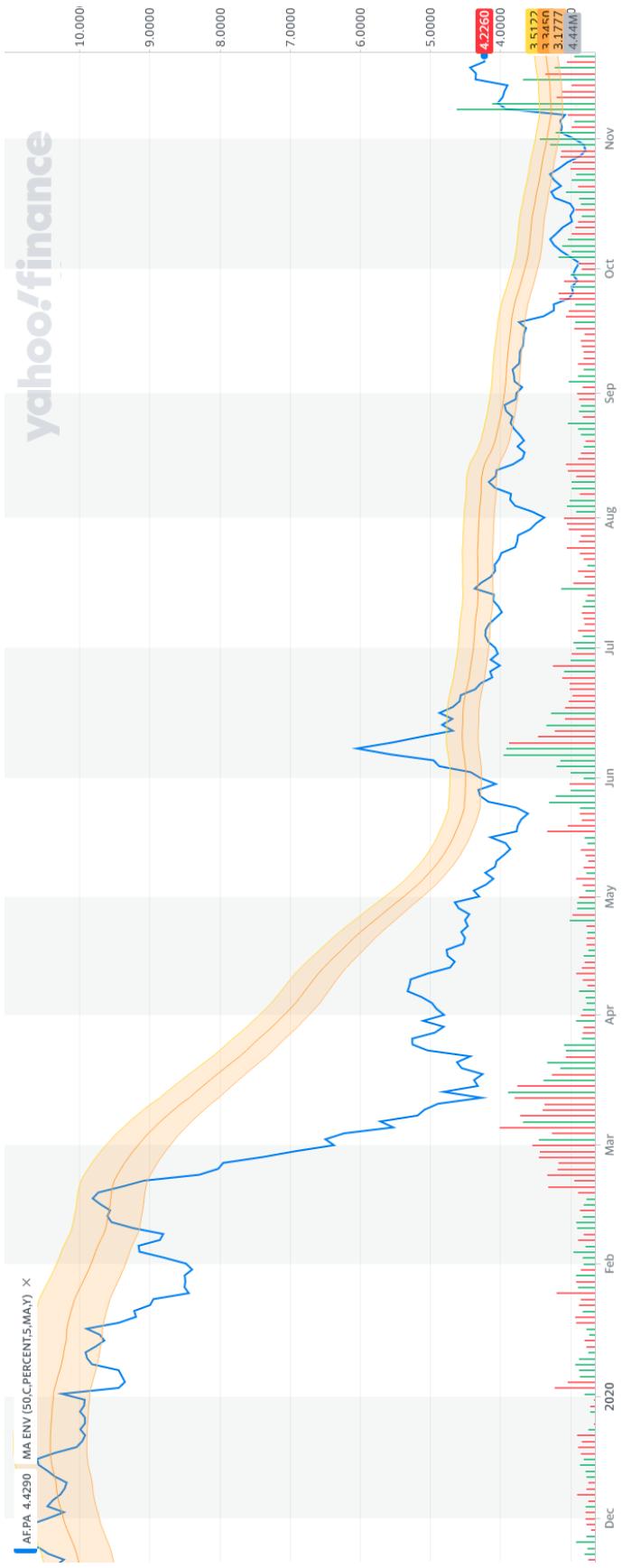
De koersgrafiek van computerbedrijf Intel (INTC) van eind 2019 tot eind 2020. In de grafiek zijn quadrant lines getekend. Bron gegevens: (Yahoo Finance, 2020).



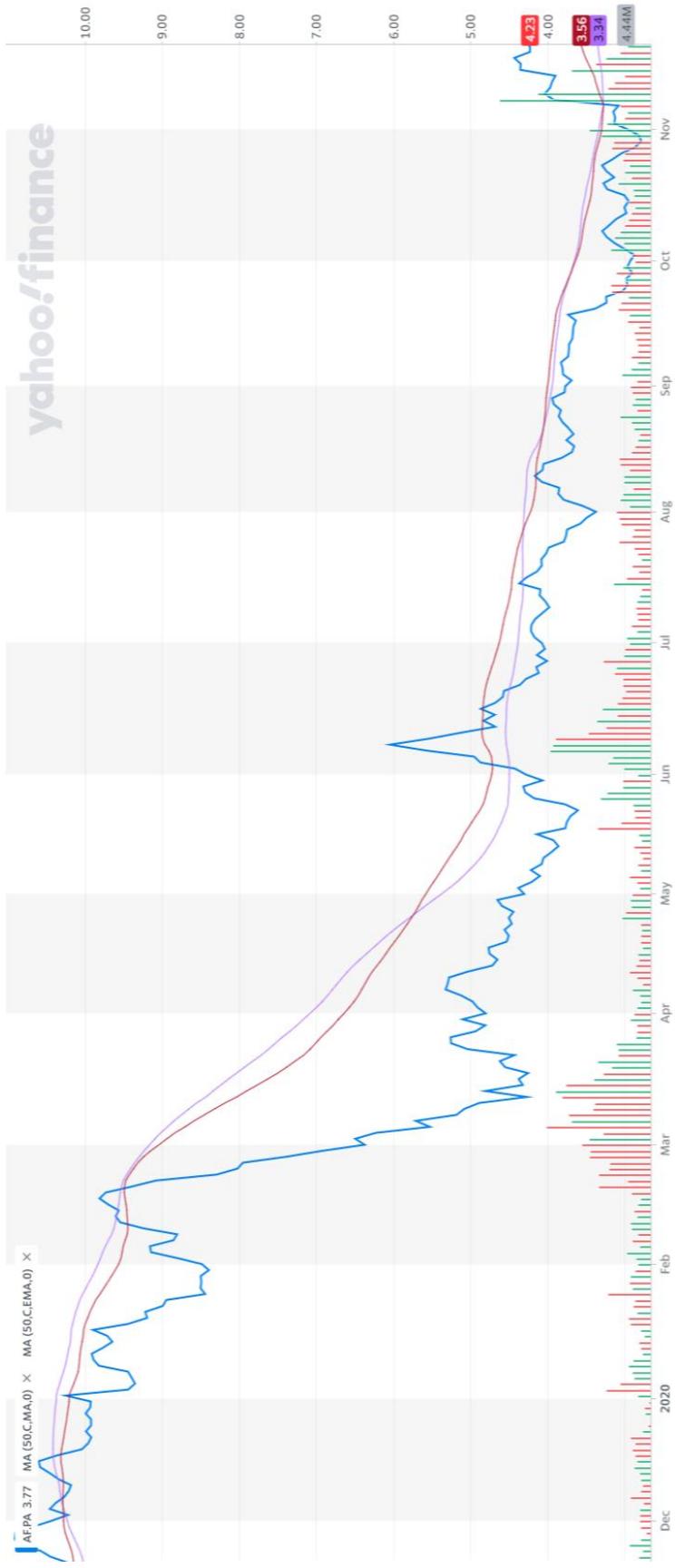
Graafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA-indicatorlijn getekend. Bron gegevens: (Yahoo Finance, 2020).



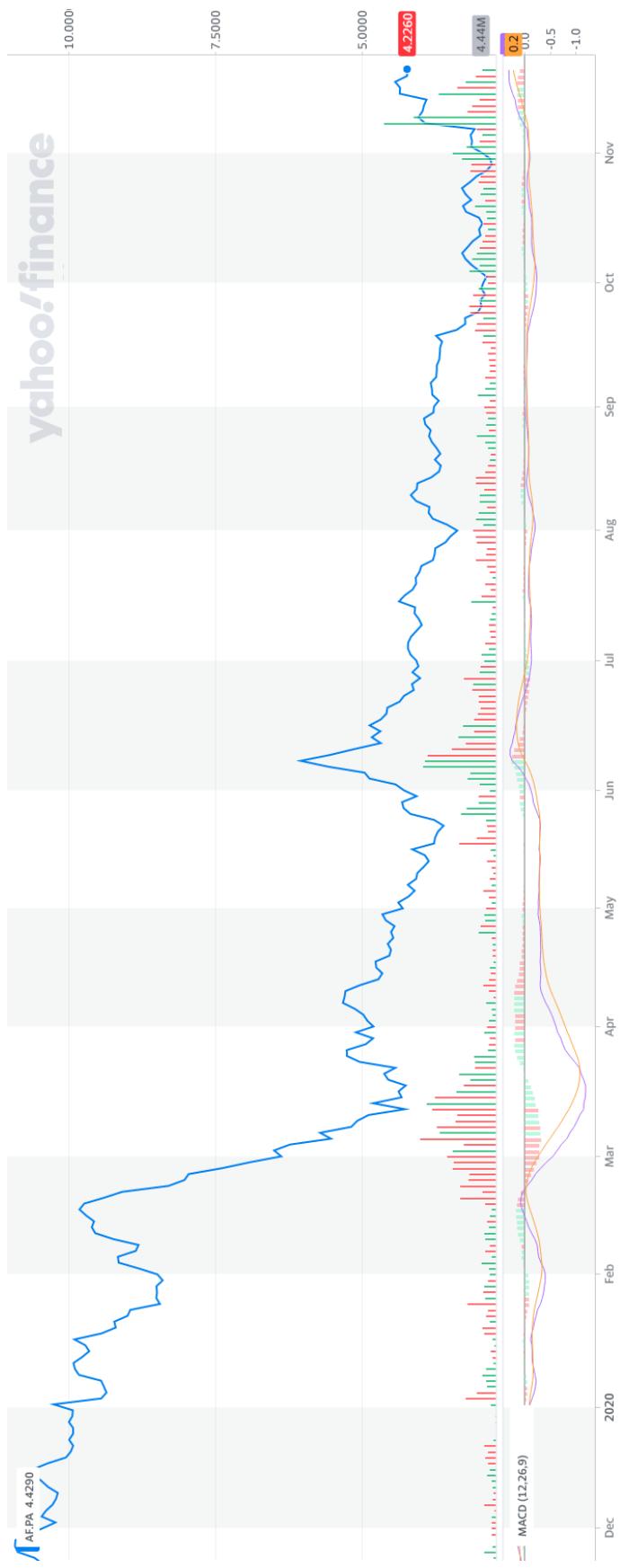
Grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA-indicatorlijn met onzekerheidsmarge getekend. Bron gegevens: (Yahoo Finance, 2020).



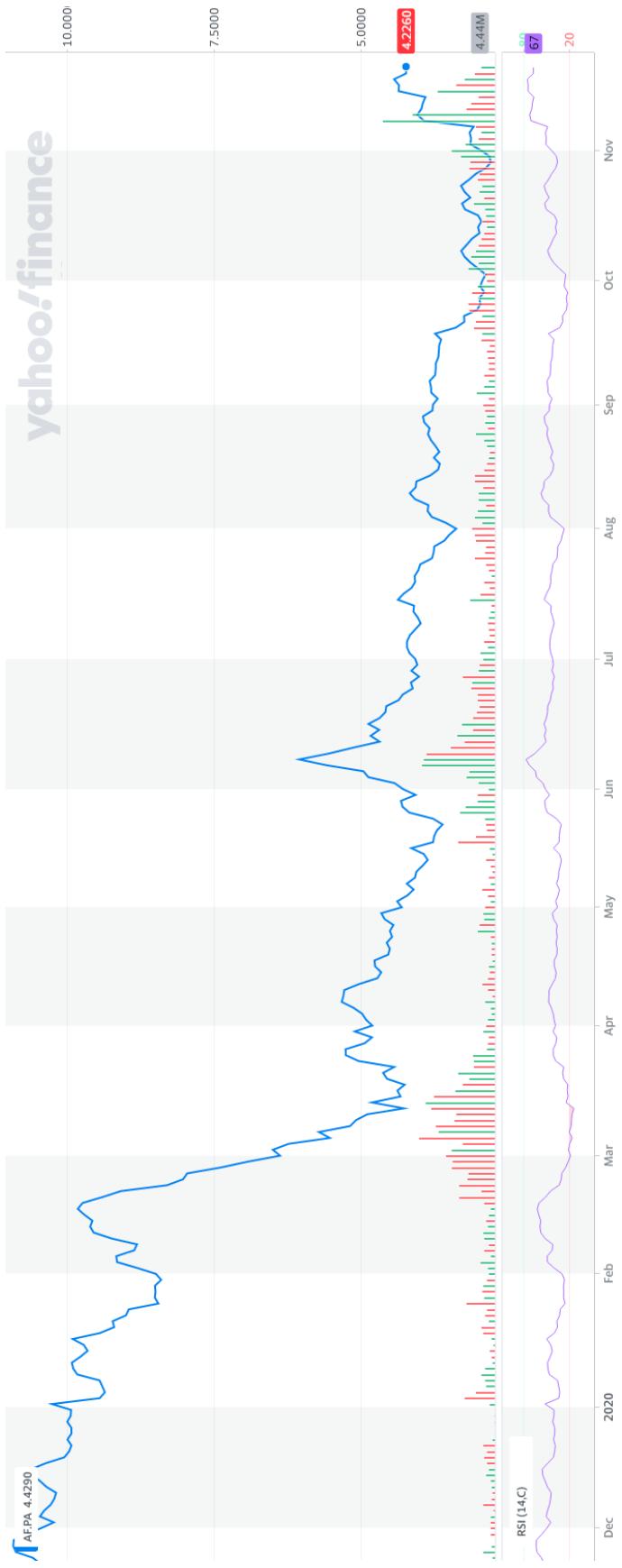
Grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020. In de grafiek is een SMA- en een EMA-indicatorlijn getekend. Bron gegevens: (Yahoo Finance, 2020).



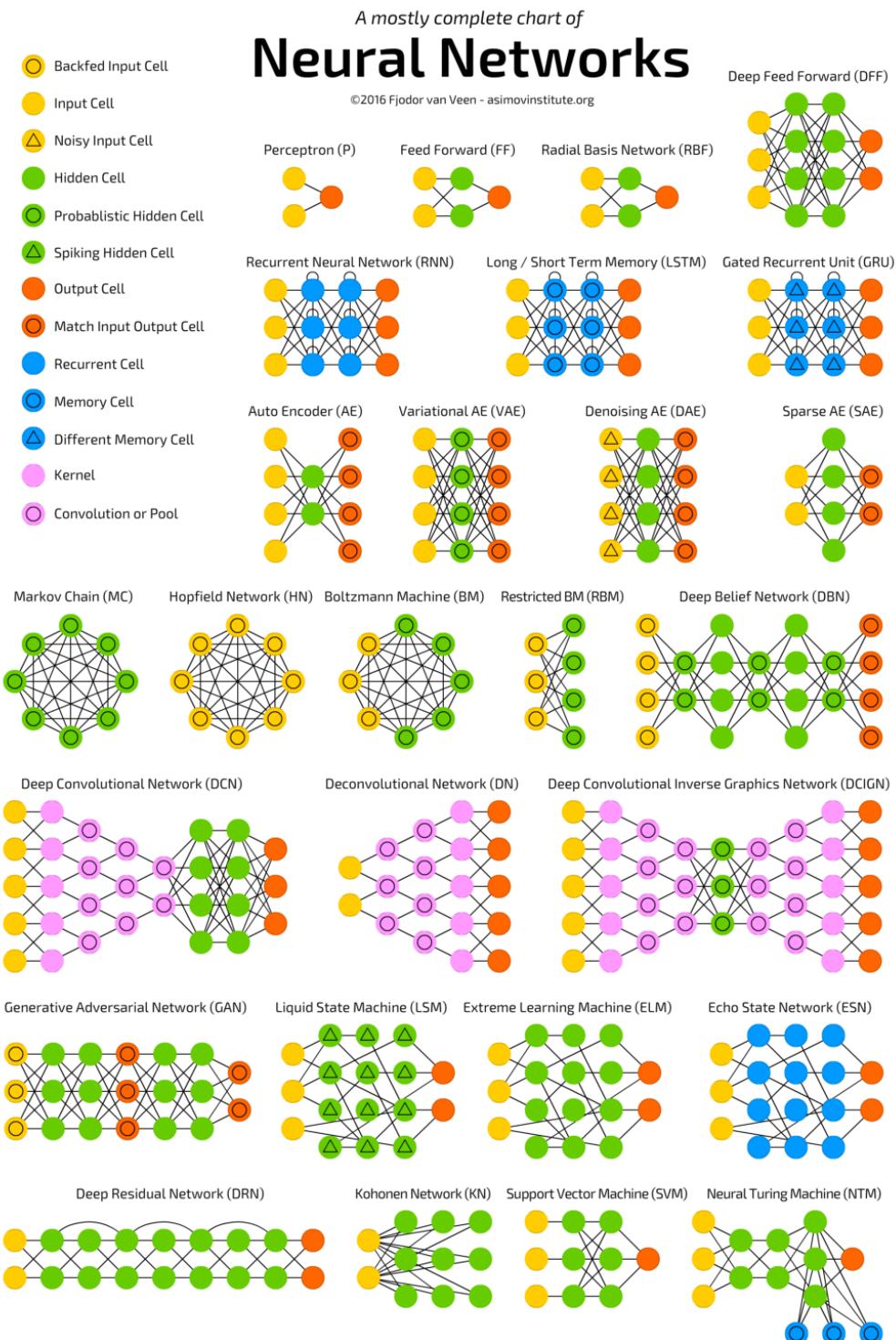
Grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020 met daaronder een MACD-indicator. Bron gegevens: (Yahoo Finance, 2020).



Grafiek van de koers van een aandeel Air France-KLM van december 2019 tot en met december 2020 met daaronder een RSI(14,C)-indicator. Bron gegevens: (Yahoo Finance, 2020).



Appendix VII: vergrote versie soorten neurale netwerken (towardsdatascience.com)



Appendix VIII: verschillende activatiefuncties met uitleg

Elk netwerk heeft zijn eigen eisen en behoeften als we het hebben over activatiefuncties. Daarom zijn er veel verschillende soorten. Het verschil tussen de activatiefuncties ligt voornamelijk bij het bereik van de functie en de helling. Maar ook worden er soms activatiefuncties gekozen voor hun efficiëntie. Hieronder bespreken wij de meest gebruikte activatiefuncties. Ook zijn bij elke functie de grafiek van de functie en de grafiek van de afgeleide gegeven.

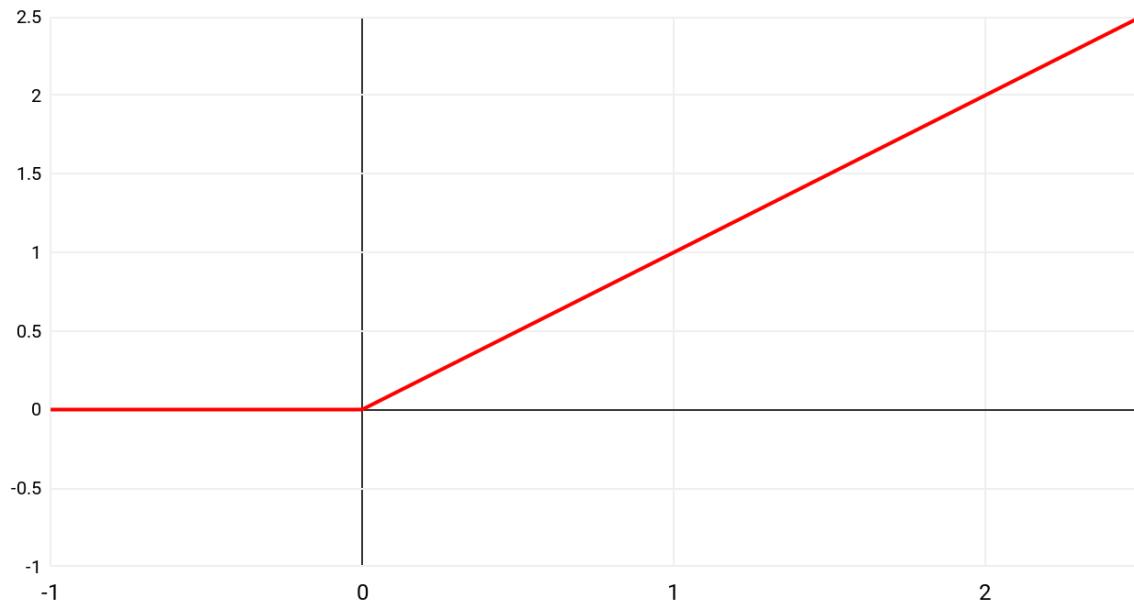
VIII.1 ReLU

ReLU is momenteel de meest gebruikte activatiefunctie (MissingLink.ai, sd). Het is de afkorting van *Rectified Linear Unit*, en is gedefinieerd als:

$$f(x) = \max(0, x) \quad (\text{VIII.1})$$

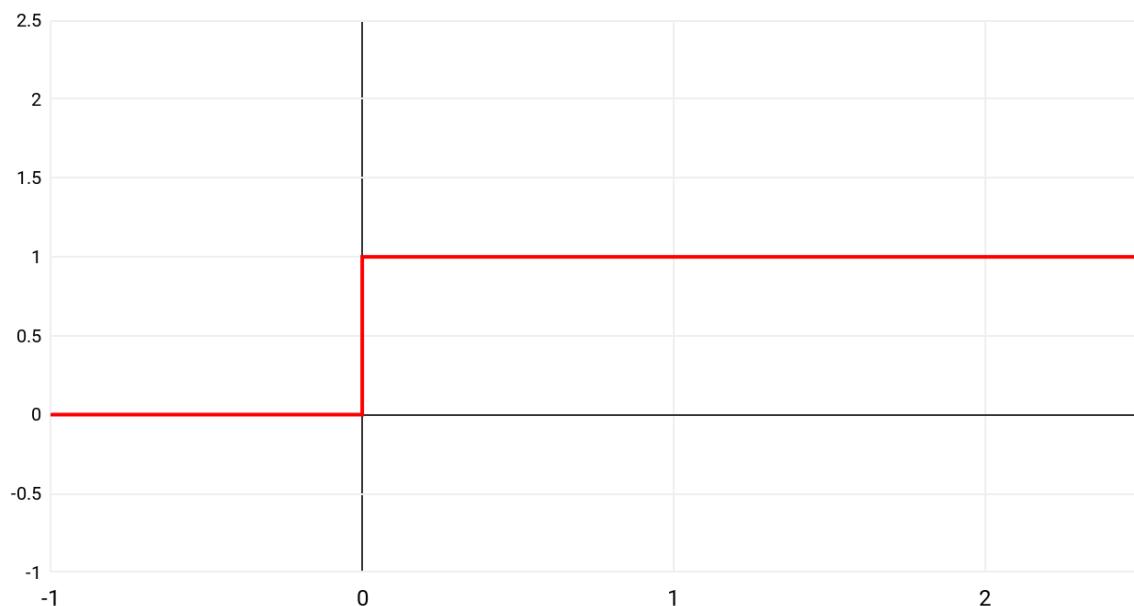
De *max* neemt simpelweg de hoogst mogelijke waarde aan. In dit geval is dat de hoogste waarde tussen 0 en x . Dit zorgt ervoor dat de uitkomst nooit lager dan 0 zal worden. Deze functie is efficiënt, al ziet de functie er redelijk lineair uit, dat is hij niet. Het grootste voordeel van deze functie is dat het niet alle neuronen tegelijkertijd activeert. Dit betekent dat backpropagation (zie hoofdstuk 4.2.8) mogelijk is voor positieve getallen. Echter, deze functie is niet perfect, omdat de helling van de functie 0 kan worden. Dit gebeurt bij negatieve x -waarden. Als de helling 0 is, is backpropagation niet meer mogelijk (Anand, 2019). Dit kan voor zogenaamde *dode neuronen* zorgen en die zijn schadelijk voor de prestaties van het netwerk. (Rout, 2020)

ReLU



Figuur VIII.a: De activatiefunctie ReLU.

Afgeleide van ReLU



Figuur VIII.b: De afgeleide van de activatiefunctie ReLU.

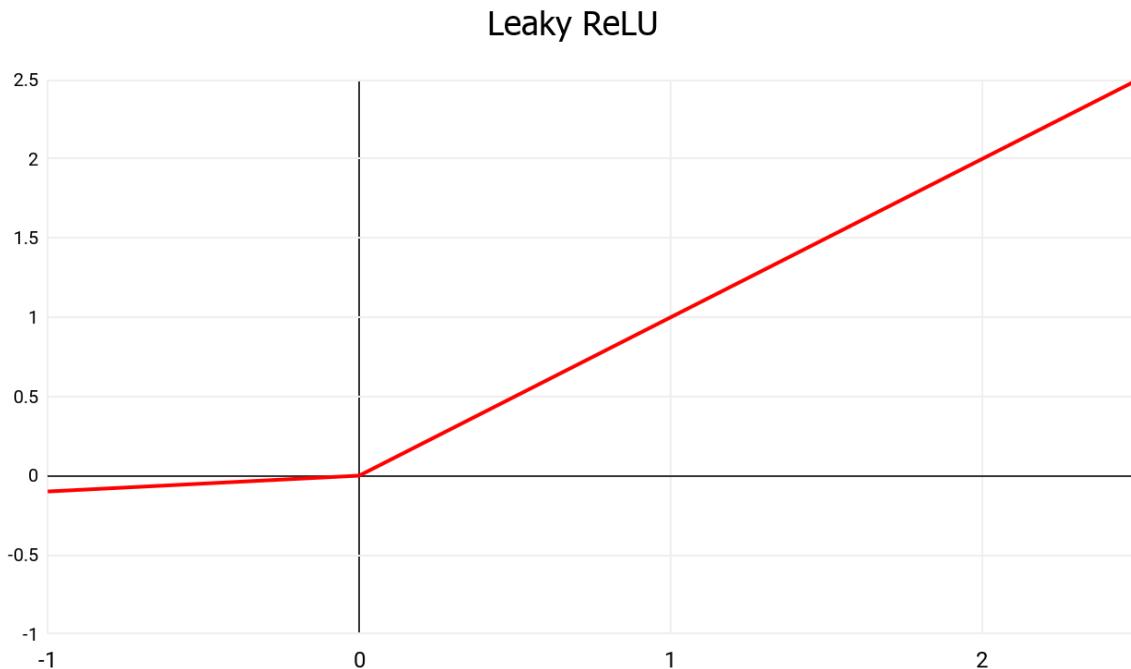
VIII.2 Leaky ReLU

De *Leaky ReLU-functie* is simpel gezegd een verbeterde versie van ReLU. Deze functie lost het probleem met de negatieve waarden op. Leaky ReLU wordt gedefinieerd als (Rout, 2020):

$$f(x) = \max(\alpha \cdot x, x) \quad (\text{VIII.2})$$

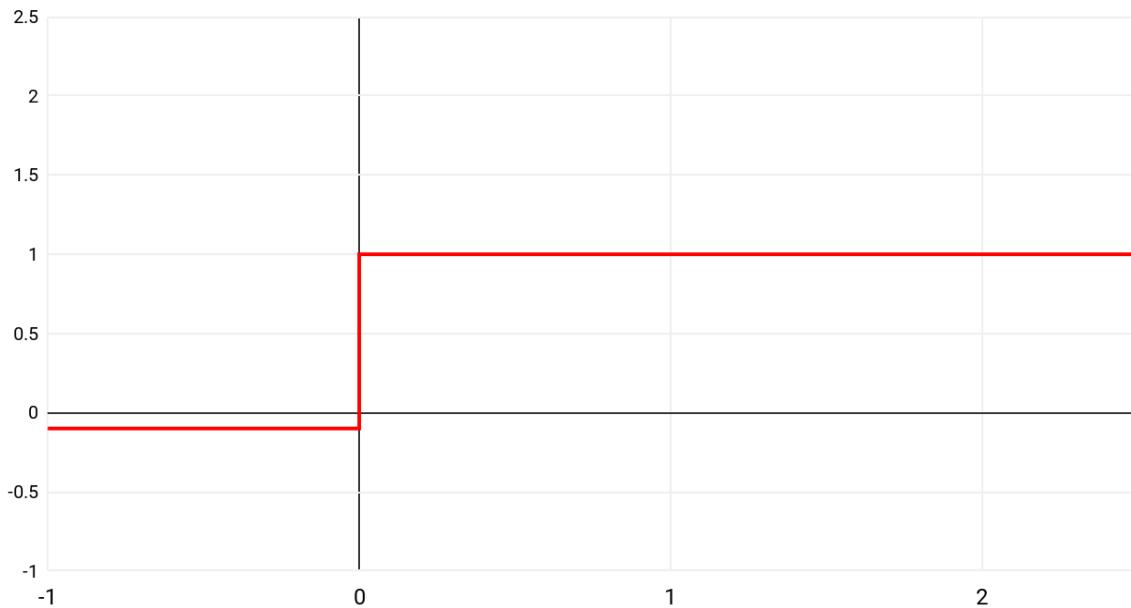
De alfa heeft hier een constante waarde (meestal rond de 0,01) die aangepast kan worden wanneer nodig. Bij deze functie bestaat er een helling voor negatieve getallen. Dit voorkomt dat neuronen in een dode staat terechtkomen. Deze functie heeft zich nog niet bewezen in de praktijk en veel mensen blijven toch ReLU gebruiken. (MissingLink.ai, sd)

In complexere netwerken kan de alfa ook in een variabele omgezet worden die het netwerk kan aanpassen. Dit zorgt ervoor dat het netwerk nog accuratere resultaten krijgt. (Anand, 2019)



Figuur VIII.c: De activatiefunctie Leaky ReLU.

Afgeleide van Leaky ReLU



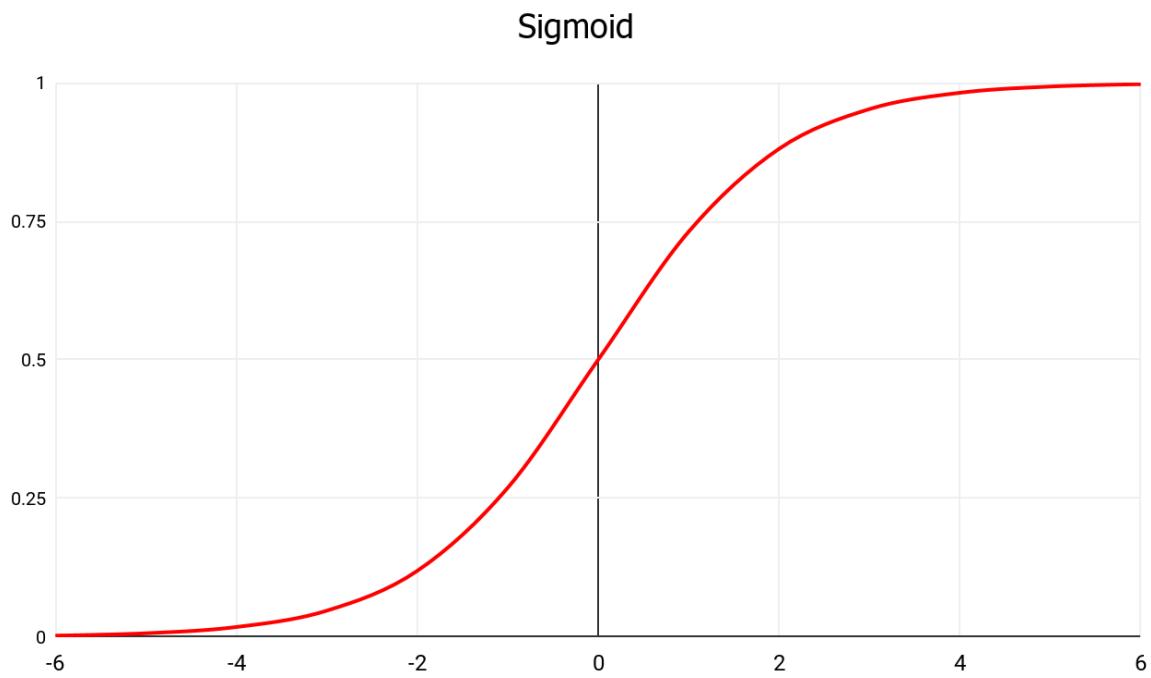
Figuur VIII.d: De afgeleide van de activatiefunctie Leaky ReLU.

VIII.3 Sigmoid

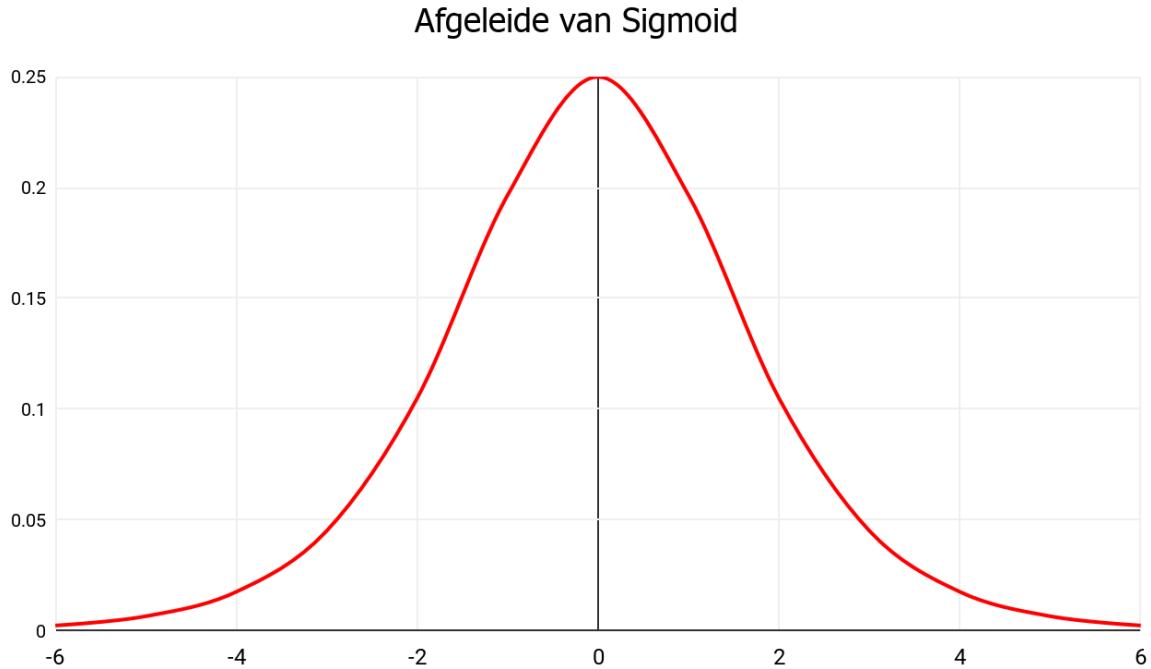
De *sigmoidfunctie* is na ReLU de meest gebruikte activatiefunctie. Deze functie wordt gedefinieerd als:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (\text{VIII.3})$$

Deze functie wordt vaak met een sigma (σ) aangeduid, vermoedelijk door zijn naam. Het grootste voordeel van deze functie is dat de helling bergvormig is. Dit is in sommige specifieke gevallen erg handig. Bovendien zorgt de sigmoid ervoor dat de uitkomst niet lineair is, wat backpropagation mogelijk maakt. Deze functie is niet perfect, omdat de helling bij erg hoge of erg lage getallen bijna 0 is. Dit vertraagt het leerproces. Dat de helling naar 0 nadert, wordt het *verdwijnendgradiëntprobleem* (Wikipedia-bijdragers, 2020) genoemd. (Anand, 2019)



Figuur VIII.e: De activatiefunctie sigmoid.



Figuur VIII.f: De afgeleide van de activatiefunctie sigmoid.

VIII.4 Tanh

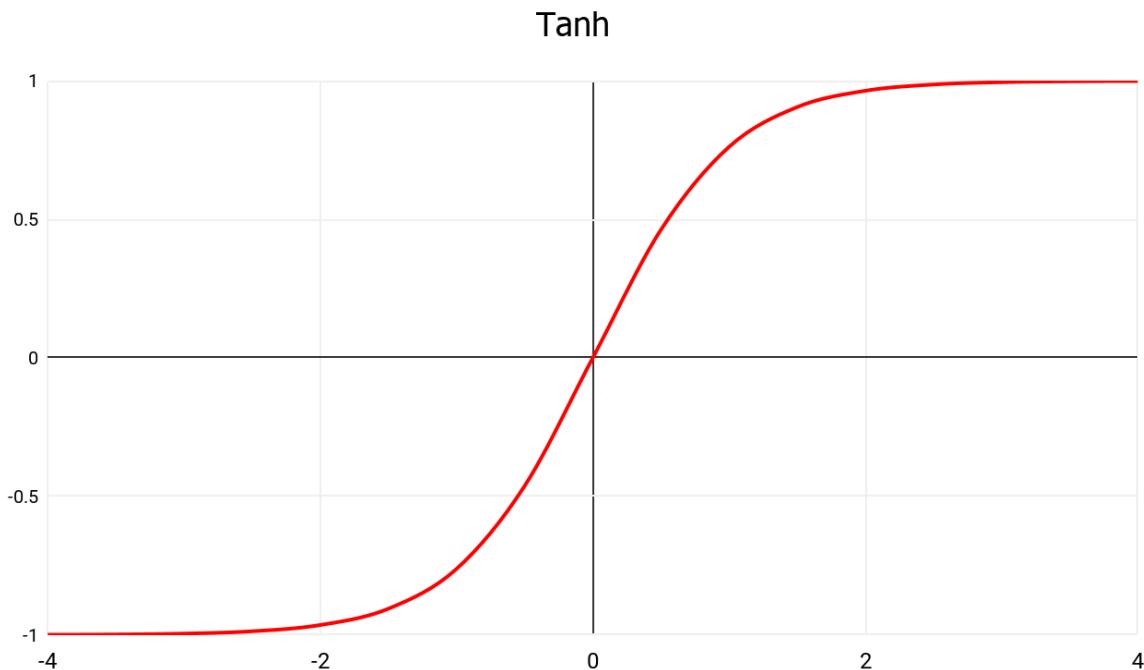
Deze functie lijkt erg op de sigmoidfunctie. *Tanh* werd oorspronkelijk gedefinieerd als:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (\text{VIII.4})$$

Maar vaker wordt deze functie opgeschreven als:

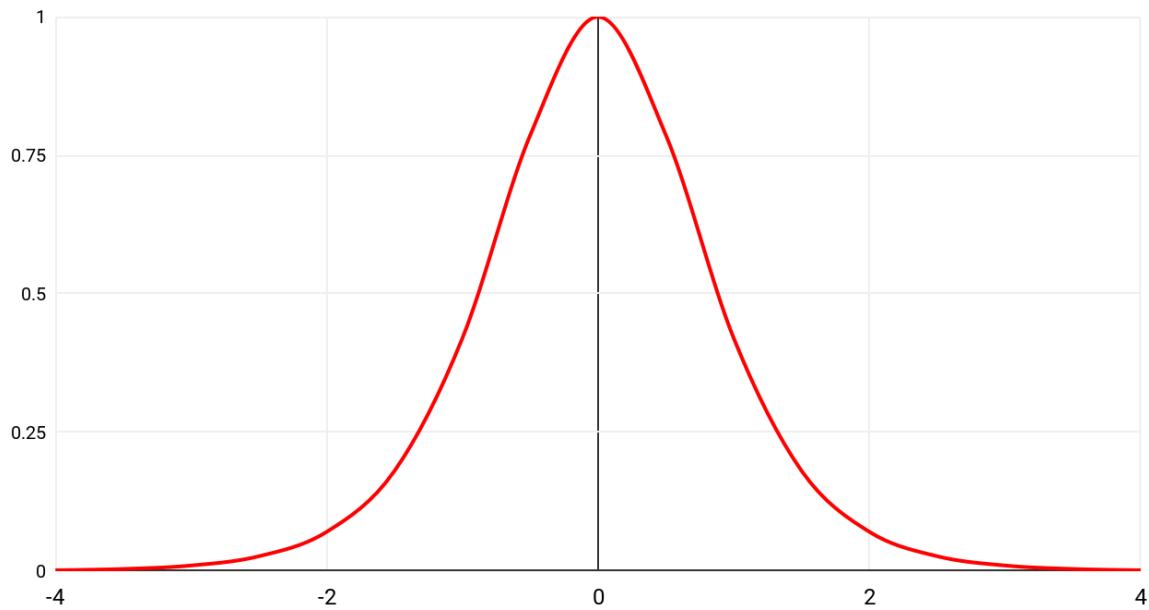
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{VIII.5})$$

Deze functie wordt gezien als de verbeterde sigmoidfunctie en is puntsymmetrisch in $(0,0)$. De helling van de afgeleide van deze functie is, net zoals sigmoid, bergvorming. Tanh heeft een steilere helling dan de sigmoidfunctie. Maar ondanks deze verbeteringen heeft de functie ook last van het verdwijnengradiente probleem. (Rout, 2020)



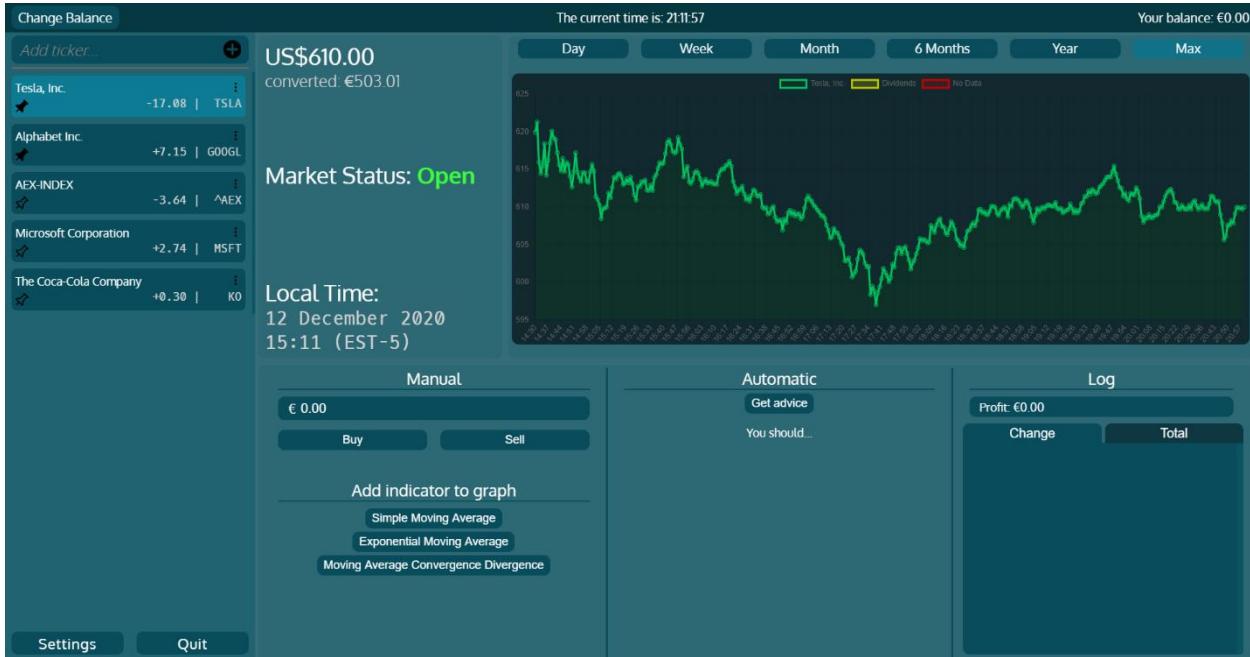
Figuur VIII.g: De activatiefunctie Tanh.

Afgeleide van Tanh

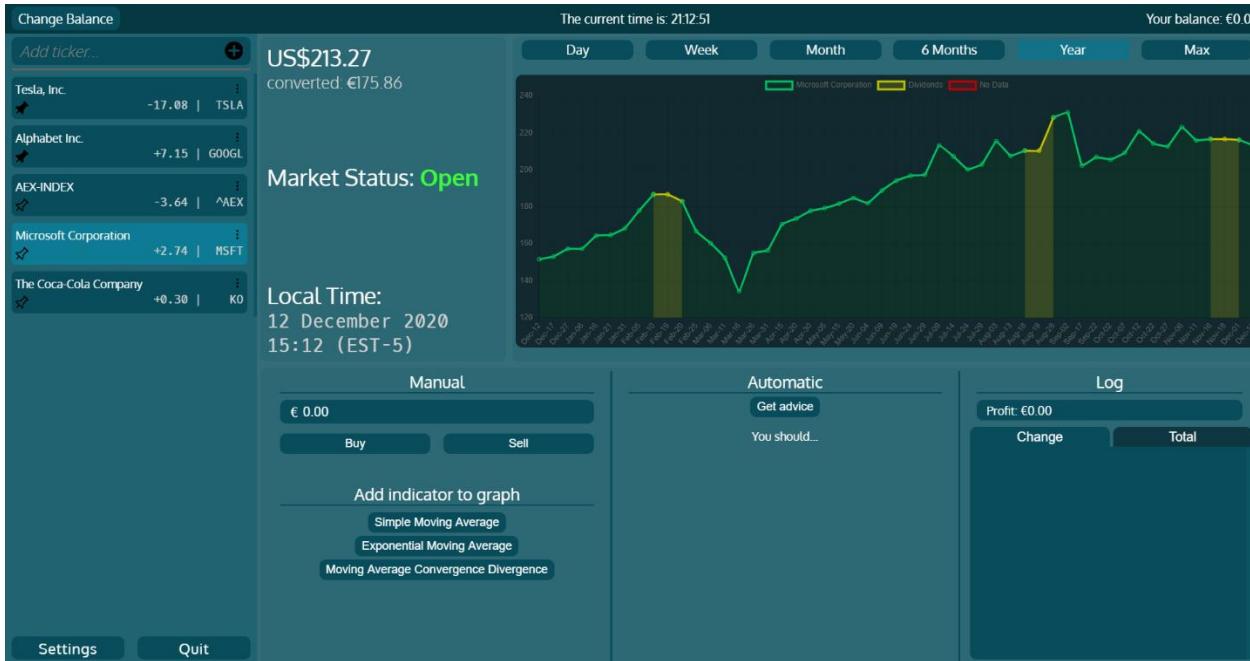


Figuur VIII.h: De afgeleide van de activatiefunctie Tanh.

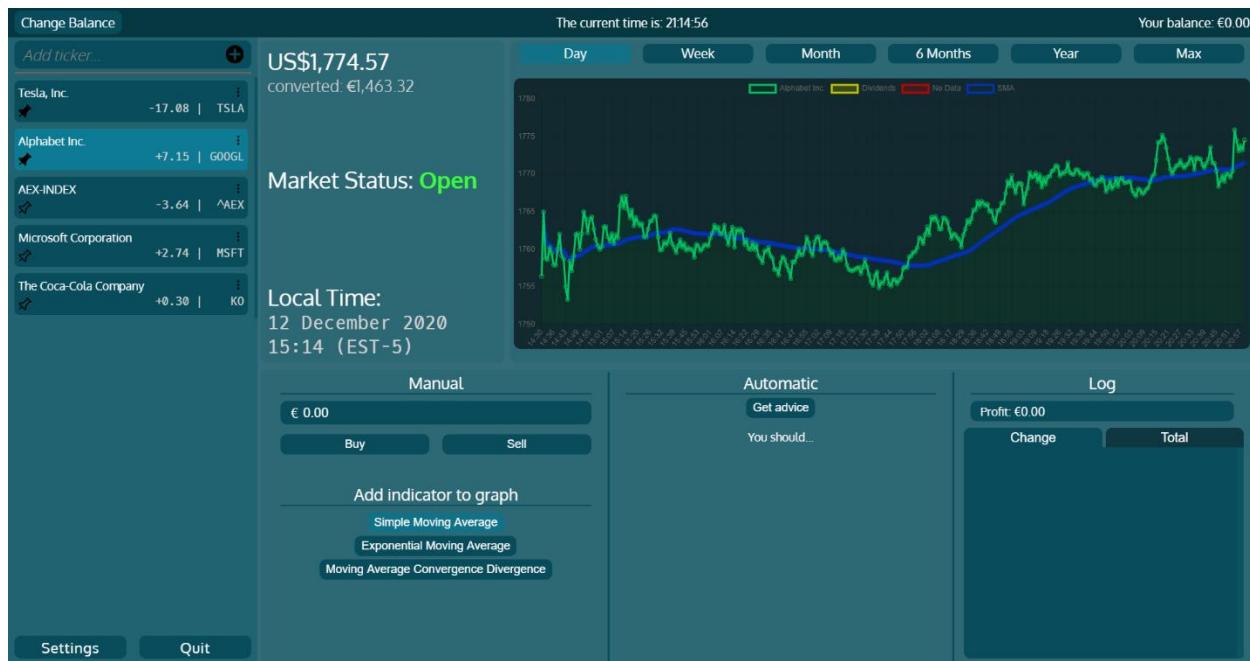
Appendix IX: screenshots van ons programma



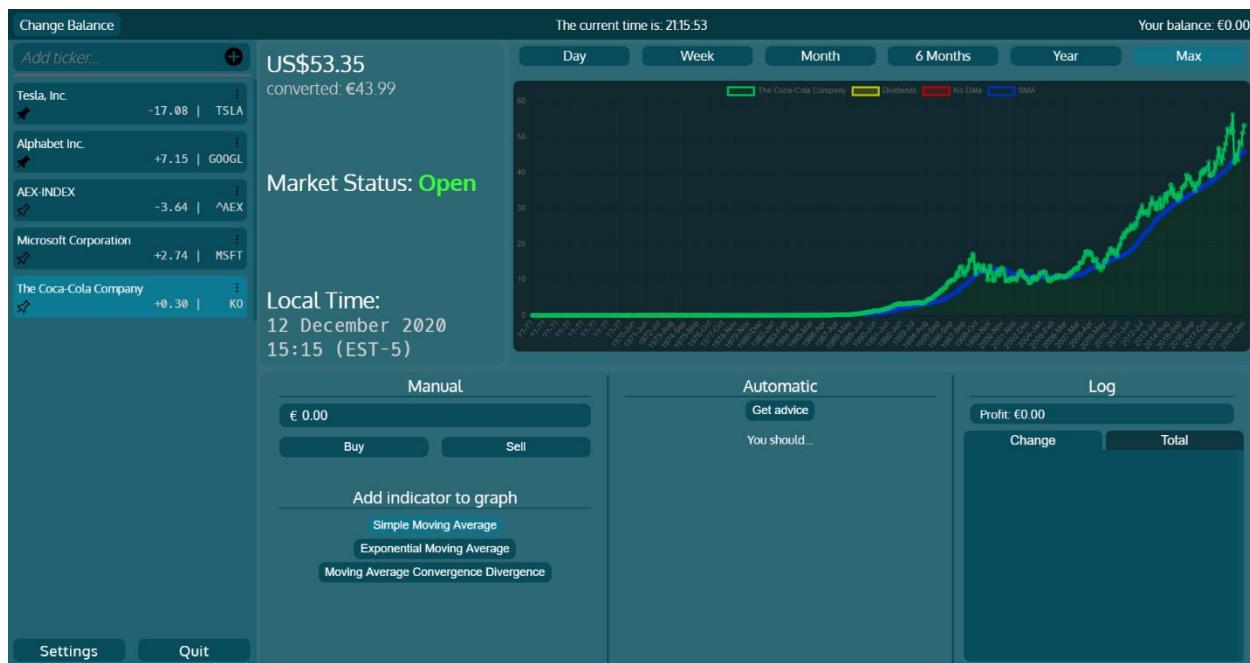
Tesla over zijn volledige levensduur.



Microsoft gedurende een jaar met zichtbare dividenden.



Google gedurende een dag met de SMA-indicator.



De volledige levensduur van Coca-Cola met een SMA-indicator. Hier is ook te zien dat datums voor 1970 niet werken. Dit komt doordat de door ons gebruikte *Unix time* begon in 1970. (Wikipedia-bijdragers, 2020)

Appendix X: Beknopt plan van aanpak PWS

X.1 Wie wij zijn

Stach Redeker, Noah Verkaik, en Sander Vermeulen.

X.2 Gekozen onderwerp

We begonnen met het vage idee om iets met programmeren te doen. Al snel streefden we richting Artificial Intelligence. Alle drie hebben we interesse in dit deel van programmeren, dus gingen we hier wat verder op in. Eerst hadden we het idee om voor een spel een AI-programma te maken. Het perfecte spel leek ons Monopoly (rekening houdend met dat er al AI's voor schaken, go, poker, en meer spellen bestaan), maar toen kreeg een van ons een veel beter idee: een AI maken voor de aandelenmarkt. Bovendien bleek er al iemand een (slecht werkende) Monopoly-AI te hebben gebouwd.

Zo zijn we bij ons huidige onderwerp gekomen en we zijn het met zijn drieën eens dat dit het beste idee is voor ons profielwerkstuk. We willen de aandelenmarkt analyseren en hierbij een computerprogramma schrijven dat zo goed mogelijk voorspellingen kan maken met de huidige gegevens van de markt. Bij dit onderwerp is er invloed vanuit meerdere vakken, waaronder wiskunde D en economie. Hierover later meer.

Dit is een programma dat, zover wij weten, nog niet bestaat. Een aantal bedrijven claimt een AI te bezitten, maar maakt de broncode niet openbaar. Ook IBM werkt aan een beginnetje (wat doen ze niet), maar dat schijnt nog niet goed te werken. Immers, waarom zouden er nog menselijke handelaren bestaan als computers het kunnen doen?

X.3 Waarschijnlijke hoofdvraag

Kan een computerprogramma beter beleggen dan een mens?

X.4 Welk vak en welke begeleider

Wij hebben een wiskundedocent nodig, omdat we hier onder andere met kansberekening en analyse van patronen te maken hebben. Aangezien wij allemaal Wiskunde D in ons vakkenpakket hebben en er ook kansberekeningen behandeld was bij dit vak, lijkt het ons logisch om onze wiskunde-D-docent M. van de Ven-Appel als begeleider te nemen.

Als wij met software gaan werken, is het natuurlijk handig als wij hier begeleiding mee kunnen krijgen. Wij hebben van V.M. Traas doorgekregen dat hij mogelijk experts, die ons kunnen helpen met programmeren, van buiten de school kan regelen.

Daarnaast is het ongetwijfeld handig als een economiedocent ons advies kan geven, omdat beleggen een economisch onderdeel is. We hebben geen economie in ons pakket en dus kunnen we geen officieel 'economie'-PWS maken, maar advies inwinnen kan altijd.

X.5 Waarom met z'n drieën

Wij denken dat het een groot project is. Ten eerste moeten we natuurlijk onderzoek doen naar de aandelenmarkt: (bewezen werkende) patronen bekijken en analyseren. Dan gaan we software schrijven op basis van de verworven gegevens. Tot slot is het natuurlijk mooi als we het kunnen vergelijken met een 'menselijke' belegger. Is een computer efficiënter? Dit alles moet natuurlijk zorgvuldig worden gedocumenteerd.

Elk van ons heeft in een of meerdere van de bovenstaande zowel interesse als ervaring. Dankzij een samenwerking met drie personen kunnen we onze krachten bundelen en er echt iets tofs van maken.

We hebben uitgezocht of het wettelijk is toegestaan om met drie personen een profielwerkstuk te schrijven. Er is geen maximaal aantal samenwerkende leerlingen gegeven, daarom gaan we ervan uit dat het mogelijk is. We zijn ons ervan bewust dat dit neerkomt op $3 \times 80 = 240$ uur werklast.