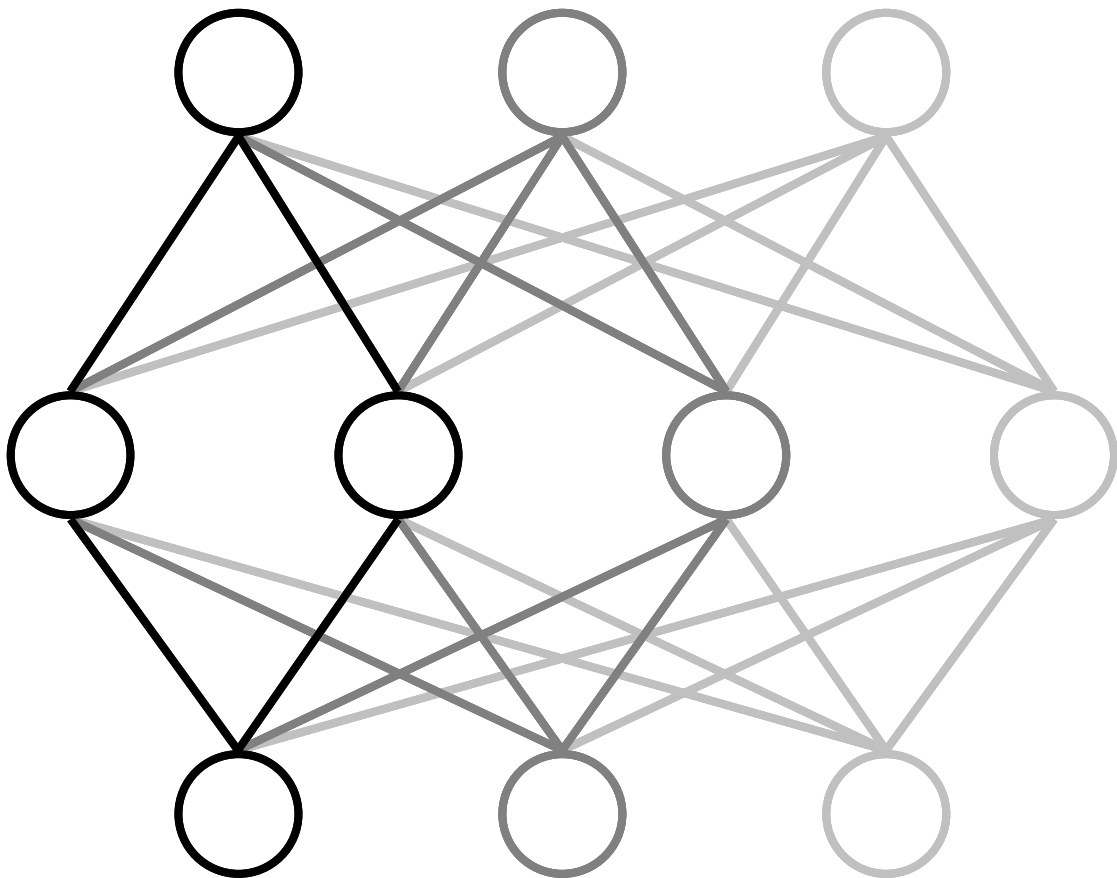


EVALUATIE VAN TWEE TYPEN NEURALE NETWERKEN IN EEN DYNAMISCHE OMGEVING

Gerbrand Kamphuis



Universiteit Utrecht

MSc Scriptie Cognitive Artificial Intelligence

Begeleider Dr. I. Th. C. Hooge

Utrecht, 12 december 2009

Inhoud

| | |
|---|----|
| Voorwoord | 4 |
| Samenvatting | 5 |
| 1 Achtergrond en doel van het onderzoek | 7 |
| 1.1 Introductie | 7 |
| 1.2 Basisbegrippen Kunstmatige Intelligentie | 7 |
| 1.3 Literatuuronderzoek | 11 |
| 1.4 Onderzoeksvraag | 14 |
| 2 Onderzoeksopzet | 15 |
| 2.1 Introductie | 15 |
| 2.2 Ontwikkeling van dynamische omgeving | 17 |
| 2.2.1 Concept | 17 |
| 2.2.2 Beschrijving | 17 |
| 2.3 Ontwikkeling van 'creatures' | 19 |
| 2.3.1 Concept | 19 |
| 2.3.2 Beschrijving | 20 |
| 2.4 Ontwikkeling van neurale netwerken | 27 |
| 2.4.1 Concept | 27 |
| 2.4.2 Beschrijving | 28 |
| 2.5 Ontwikkeling van leermethoden | 33 |
| 2.5.1 Concept | 33 |
| 2.5.2 Beschrijving | 34 |
| 2.6 Ontwikkeling dataverwerking en analyse methodologie | 38 |
| 2.6.1 Concept | 38 |
| 2.6.2 Beschrijving | 38 |
| 2.7 Testen en aanpassen | 41 |
| 2.7.1 Inleiding | 41 |
| 2.7.2 De startsituatie | 42 |
| 2.7.3 Aanpassingen | 48 |
| 2.7.4 'Leuke' Bugs | 49 |
| 3 Uitvoering onderzoek | 51 |
| 3.1 Wereld 1 | 52 |
| 3.2 Wereld 2 | 54 |
| 3.3 Wereld 3 | 56 |
| 3.4 Wereld 4 | 58 |
| 3.5 Wereld 5 | 60 |
| 4 Resultaten | 61 |
| 4.1 Introductie | 61 |
| 4.2 Onderzoekresultaten | 62 |
| 5 Conclusie en discussie | 74 |
| Referenties | 78 |

VOORWOORD

Binnen het vakgebied kunstmatige intelligentie nemen neurale netwerken een belangrijke plaats in omdat ze zowel de mogelijkheid bieden om de werking van onze hersenen beter te begrijpen als om kunstmatige intelligentie te ontwikkelen. Aangezien ik geïnteresseerd ben in zowel bewustzijn als intelligentie heb ik mijn onderzoek op neurale netwerken gericht. Het onderzoek was bovendien leuk omdat de neurale netwerken in mijn onderzoek als creatures in een omgeving leefden waardoor je echt kon zien dat er werd geleerd en omdat er tijdens de ontwikkeling soms onverwachte maar interessante resultaten tevoorschijn kwamen. Verder bood het onderzoek me de gelegenheid om me verder te verdiepen en te bekwamen in de programmeertaal D. Dit onderzoek is zinvol geweest, niet alleen omdat ik veel heb geleerd, maar ook omdat de onderzoeksresultaten duidelijk zijn en de ontwikkelde onderzoeksopzet mogelijkheden biedt voor vervolgonderzoek, door mijzelf of anderen. Bij deze wil ik Ignace Hooge danken voor zijn begeleiding en steun aan mijn onderzoek. Ook dank aan het bedrijf MeinMein van mijn broer voor de alpha versie van het programma dat het raamwerk vormde voor de ontwikkelde onderzoeksopzet, en de technische ondersteuning van Meindert Kamphuis hierbij.

Gerbrand Kamphuis

SAMENVATTING

Onderzoeksdoel

Deze MSc scriptie geeft verslag van het onderzoek naar de vraag of de leersnelheid van een neurale netwerk dat stapsgewijs wordt opgebouwd met zowel input, hidden als output neuronen verbetert in vergelijking tot een niet stapsgewijs opgebouwd netwerk. Hiervoor is een nieuw netwerk model ontworpen, het Incremental Neuraal Netwerk (INN), dat wordt vergeleken met een standaard neurale netwerk, het Sequential Neuraal Netwerk (SNN). Het basisverschil tussen deze netwerken is, dat het SNN altijd een volledig netwerk is, d.w.z. gelijk in aantal neuronen en onderlinge verbindingen, terwijl het INN met een klein netwerk begint dat vervolgens wordt uitgebreid. Gebaseerd op de eigenschappen van beide netwerken is de hypothese dat het INN betere resultaten oplevert dan het SNN.

Onderzoeksopzet

Om de netwerken te kunnen evalueren worden ze gebruikt als hersenen van 'creatures' die kunnen observeren, bewegen, eten en voortplanten. Hun omgeving bestaat uit een wereld van een bepaalde omvang, met muren en voedsel, waarin de creatures, eventueel in interactie met andere creatures, de genoemde acties ondernemen. Voor dit onderzoek zijn vijf werelden ontworpen. Het leerdoel van de creatures verschilt per wereld en wordt complexer voor opeenvolgende werelden. De creatures moeten een aantal werelden sequentieel doorlopen om succesvol de opeenvolgende doelen te kunnen halen; ze hebben de resultaten - geleerd gedrag - in wereld 1 nodig om het doel in wereld 2 te kunnen behalen, etc. In tabel 1 zijn de belangrijkste kenmerken en doelen van de werelden kort samengevat. Het leerproces van de creatures in deze werelden verloopt volgens een evolutionair proces, d.w.z. de populatie van creatures verbetert onder invloed van natuurlijke selectie.

| | Kenmerken | Doel | Subdoel(en) |
|-----------------|--|--|-----------------------|
| Wereld 1 | 3 food blokjes willekeurig geplaatst 1 creature | Voedsel vinden | Doelgericht bewegen |
| Wereld 2 | 3 food blokjes willekeurig geplaatst 1 creature | Voedsel eten | Als in wereld 1 |
| Wereld 3 | 3 food blokjes willekeurig geplaatst 1 creature | Zo lang mogelijk leven | Als in wereld 1 en 2 |
| Wereld 4 | 5 food blokjes willekeurig geplaatst 1 creature 3 andere creatures aanwezig | Nakomelingen maken | Als in wereld 1 t/m 3 |
| Wereld 5 | Grote wereld Veel food blokjes willekeurig geplaatst Initiële populatie van 50 creatures | Populatie zelfstandig in stand houden, ofwel niet uitsterven | Als in wereld 1 t/m 4 |

Tabel 1 Samenvatting van kenmerken en doelen van werelden in dit onderzoek

Onderzoeksresultaten

Uit de onderzoeksresultaten blijkt dat het Incremental Neuraal Netwerk in wereld 1, 2 en 4 significant minder generaties nodig heeft om het gestelde leerdoel te behalen dan het Sequential Neuraal Netwerk. Deze resultaten bevestigen de onderzoekshypothese dat het INN sneller evolueert in een dynamische omgeving dan het SNN. Uit het onderzoek blijkt verder dat de grotere leersnelheid van het INN niet wordt veroorzaakt door het verschil in grootte van het netwerk, want ook in wereld 4, waar beide netwerken even groot zijn, is het INN sneller. Daarnaast laten de resultaten van wereld 5 zien dat er, behalve het verschil in leersnelheid, geen verschil is tussen de functionaliteit van de netwerken; beide netwerken leiden namelijk tot een stabiele zichzelf in stand houdende populatie. Hieruit kan de conclusie worden getrokken dat het ontwikkelde concept, het stapsgewijze opbouwen van een netwerk met zowel input, hidden en output neuronen, een positief effect heeft op de leersnelheid.

Verder onderzoek

Naast de onderzoeksresultaten heeft het onderzoek ook geresulteerd in een nieuwe onderzoeksomgeving. Door opeenvolgende iteraties van testen en aanpassen van de onderzoeksopzet en onderdelen ervan is een testomgeving ontwikkeld waarmee het mogelijk was de hierboven genoemde vraagstelling te testen. De in dit rapport beschreven onderzoeksopzet en de broncode biedt anderen niet alleen de mogelijkheid om het onderzoek te herhalen, maar ook om het te gebruiken voor verder onderzoek.

De ontworpen onderzoeksopzet is echter niet alleen bruikbaar voor het onderzoeken van neurale netwerken, maar biedt ook mogelijkheden voor ander onderzoek, zoals het vergelijken van leermethoden en het bestuderen van het gedrag van creatures. Zo biedt de onderzoeksopzet de mogelijkheid om twee populaties in dezelfde omgeving met elkaar te laten concurreren. Door de verschillende soorten creatures verschillende typen netwerken te laten gebruiken kan ook het verschil in eigenschappen tussen deze netwerken in een dergelijk vervolgonderzoek worden betrokken.

1 ACHTERGROND EN DOEL VAN HET ONDERZOEK

1.1 Introductie

Het onderzoeksgebied Kunstmatige Intelligentie (KI) richt zich op het creëren en begrijpen van intelligentie. KI wordt onder andere gebruikt voor data mining, spraakherkenning, logistiek, medische diagnostiek, industriële robotica, banking en (internet) search engines. Hiervoor worden verschillende middelen gebruikt, zoals zoekalgoritmen, statistische classificatie methoden en neurale netwerken. Dit onderzoek richt zich op neurale netwerken. Neurale netwerken zijn op het biologische brein geïnspireerde systemen die tot doel hebben om de werking van het brein beter te begrijpen en intelligentie te simuleren voor specifieke toepassingen.

In dit hoofdstuk worden achtergrond en doel van het onderzoek beschreven. Hiervoor wordt eerst ingegaan op een aantal basisbegrippen uit de kunstmatige intelligentie. Daarna wordt in hoofdstuk 2 de onderzoeksopzet beschreven en in hoofdstuk 3 het uitgevoerde onderzoek.. De onderzoeksresultaten komen in hoofdstuk 4 aan de orde, waarna in hoofdstuk 5 de conclusies worden gepresenteerd gevolgd door een discussie van de resultaten.

1.2 Basisbegrippen Kunstmatige Intelligentie

Kunstmatige Intelligentie

De term 'kunstmatige intelligentie' (KI) werd het eerst gebruikt door John McCarthy in 1956, die als definitie gaf: '*the science and engineering of making intelligent machines*'. Dit is echter niet de enige definitie. Nu, ruim 50 jaar later, bestaat er nog steeds geen eenduidige definitie van KI.

In [Russell] worden, na analyse van verscheidene gebruikte definities, de volgende vier categorieën van definities onderscheiden:

- Systemen die denken zoals mensen.
- Systemen die zich gedragen als mensen.
- Systemen die rationeel denken.
- Systemen die zich rationeel gedragen.

KI houdt zich dus voornamelijk bezig met het creëren en het begrijpen van intelligentie, waarbij het niet noodzakelijk menselijke intelligentie betreft. Het onderzoeksgebied van KI is zeer breed, met uiteenlopende onderwerpen waaronder bijvoorbeeld cognitieve

neurowetenschap, logica, natuurkunde, natuurlijke taalverwerking, perceptie, motoriek en zoekalgoritmen.

Neurale Netwerken

Neurale netwerken zijn onder te verdelen in biologische en kunstmatige neurale netwerken. Biologische neurale netwerken vormen het zenuwstelsel van dieren en spelen een rol bij het verwerken van informatie, zoals zintuiglijke waarnemingen, en het uitvoeren van handelingen, zoals voortbeweging. Bij complexere diersoorten kunnen ze ook leiden tot emotionele en cognitieve processen. Voor een gedetailleerde beschrijving van biologische neurale netwerken, zie bijvoorbeeld [Gazzaniga].

Kunstmatige neurale netwerken zijn, in meer of mindere mate, gebaseerd op biologische neurale netwerken. De functionaliteit is om verschillende redenen vaak sterk versimpeld, voornamelijk omdat kunstmatige neurale netwerken (computationeel) duurder worden naarmate ze gedetailleerder zijn en voor veel toepassingen en onderzoek volstaan kan worden met een relatief eenvoudig model.

Kunstmatige neurale netwerken zijn, net als biologische neurale netwerken, opgebouwd uit neuronen die op een bepaalde manier met elkaar zijn verbonden. Neuronen krijgen bepaalde input en geven op basis daarvan een bepaalde output. Bij biologische neuronen gebeurt dit via elektrochemische signalen en dit wordt bij kunstmatige neuronen in meer of mindere mate gesimuleerd. De verbindingen tussen neuronen bepalen de input en output; de output van het ene neuron is de input van een of meer andere neuronen.

Er zijn verschillende typen kunstmatige neurale netwerken, met verschillende complexiteit en biologische plausibiliteit, bijvoorbeeld het Perceptron [Rojas], Kohonen netwerk [Kohonen], Hodgkin-Huxley model [Hodgkin] en Spiking Neuraal Netwerk [Vreeken] [Izhikevich]. Deze netwerken worden voor verschillende onderzoeksdoeleinden gebruikt, zoals bijvoorbeeld voor onderzoek naar de werking van hersenen, neuronen of intelligentie. Maar er wordt ook veel onderzoek verricht dat is gericht op de kunstmatige neurale netwerken zelf, bijvoorbeeld om netwerken te maken die snel leren of om computationeel goedkope maar toch slimme netwerken te ontwikkelen.

In dit verslag wordt de term 'neurale netwerken' verder exclusief gebruikt om te verwijzen naar kunstmatige neurale netwerken.

Leeralgoritmen

Neurale netwerken worden gebruikt om bepaalde data, of 'input', te verwerken tot een gewenste uitkomst, of 'output'. Bij zeer kleine netwerken kunnen de verschillende waarden, zoals gewichten van verbindingen, zo worden gekozen dat het netwerk de gewenste eigenschappen vertoont. Maar het aantal waarden in een netwerk neemt exponentieel toe

met het aantal neuronen en verbindingen, waardoor het al zeer snel te complex wordt voor een mens om de waarden te kunnen kiezen. Daarom zijn er leeralgoritmen ontwikkeld die de waarden automatisch veranderen. Dit leren kan worden onderverdeeld in twee categorieën, namelijk 'supervised' of 'unsupervised'.

Supervised leren wordt zo genoemd omdat tijdens het leerproces informatie wordt verschaft aan het leeralgoritme over de kwaliteit (d.w.z. de gewenste eigenschappen) van het netwerk. Deze leeralgoritmen werken met behulp van input-output paren; er zijn verscheidene input waarden waar de gewenste output waarden van bekend zijn. Wanneer een van deze inputs aan het netwerk wordt gepresenteerd, kan de verkregen output vergeleken worden met de gewenste output. Het verschil tussen de gewenste en echte output wordt dan gebruikt om de waarden of zelfs de architectuur van het netwerk te veranderen. Een aantal voorbeelden van supervised leeralgoritmen zijn Backpropagation [LeCun], Architecture Learning [Fahlman] [Aran] en Spike Timing [Kasinski].

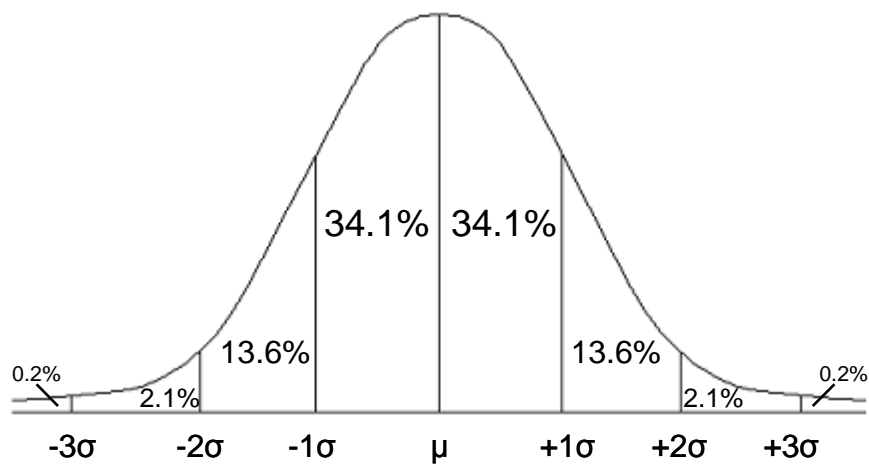
Bij unsupervised leren zijn de gewenste outputs bij bepaalde inputs niet bekend; er zijn dus geen input-output paren beschikbaar. Het netwerk krijgt verschillende waarden als input waarin het zelfstandig een patroon probeert te vinden. Unsupervised leren wordt bijvoorbeeld toegepast als er over data wordt beschikt waarvan niet duidelijk is of er een bepaald patroon in zit. Voorbeelden van unsupervised leeralgoritmen zijn Competitive Learning [Rumelhart] en Hebbian Learning [Gerstner].

Evolutionaire algoritmen

Evolutionaire algoritmen zijn gebaseerd op natuurlijke evolutie. In plaats van een leeralgoritme toe te passen op een enkel netwerk, worden er meerdere netwerken gebruikt die samen een 'populatie' vormen. De samenstelling van deze populatie verandert doordat 'slechte' individuen (d.w.z. netwerken die niet de gewenste eigenschappen bezitten) worden vervangen door nieuwe individuen. Deze nieuwe individuen worden opgebouwd uit de goede individuen van de populatie door seksuele of asexuele reproductie te simuleren. De belangrijkste methoden die bij reproductie worden gebruikt zijn 'crossover' en 'mutatie'. Bij crossover worden delen van twee individuen (ouders) samengevoegd tot een nieuw individu (kind), die dus eigenschappen bevat van beide ouders. Door mutatie worden individuen willekeurig een klein beetje veranderd, waardoor nieuwe eigenschappen kunnen ontstaan. Door crossover en mutatie veranderen populaties in de loop der tijd onder invloed van natuurlijke selectie. Hoewel er een hele populatie van netwerken moet worden gesimuleerd, kan evolutionair leren toch leiden tot een sneller leerproces (zie bijvoorbeeld [Bullinaria]) dan een leer methode met een enkel netwerk.

Normaalverdeling

De initiële waarden van nieuw gegenereerde netwerken (zie hoofdstuk 2.4) en de veranderingen van deze waarden (zie hoofdstuk 2.5) worden bepaald door middel van een zogenaamde 'normaalverdeling'. De normaalverdeling is een veel gebruikte verdeling van waarden, die is weergegeven in figuur 1. De verdeling wordt bepaald door een gemiddelde (μ) en een standaard deviatie (σ). Het gemiddelde geeft het midden van de verdeling aan, en de standaard deviatie hoezeer de waarden in de verdeling van het gemiddelde verschillen. Ofwel, 34.1% van de waarden in een normaalverdeling met $\mu = 0$ en $\sigma = 1$ zitten tussen de 0 en 1, 13.6% tussen de 1 en 2, etc. Wanneer een willekeurige waarde uit een normaalverdeling met $\mu = 0$ en $\sigma = 1$ wordt getrokken dan zit deze waarde met 34.1% + 34.1% = 68.2% kans tussen de -1 en 1.



Figuur 1 Schematische weergave van een normaalverdeling

1.3 Literatuuronderzoek

Binnen KI worden neurale netwerken gebruikt en wordt er veel onderzoek verricht om deze te verbeteren of om nieuwe netwerken te ontwikkelen. Het doel van dit onderzoek is hieraan een bijdrage te leveren. Hierbij is uitgegaan van het idee dat het mogelijk zou moeten zijn om de leersnelheid van neurale netwerken te verbeteren door een netwerk stapsgewijs op te bouwen door het toevoegen van input, hidden en output neuronen, waardoor taken opgesplitst kunnen worden in simpelere deeltaken.

Om te achterhalen of dit idee al eerder is uitgewerkt en het onderzoek verder richting te geven, is een literatuuronderzoek verricht. De resultaten hiervan worden hieronder beschreven.

Een in de literatuur veel beschreven type neuraal netwerk is het Feedforward neuraal netwerk. De standaard manier om zulke netwerken te trainen is met behulp van het backpropagation algoritme [LeCun]. Dit algoritme heeft echter bepaalde tekortkomingen, waarvan het meest belangrijke de benodigde tijd is. Wat hier een grote rol bij speelt is het zogenaamde 'Moving Target' probleem [Fahlman]. Een neuron kan niet tot zijn optimale waarde komen omdat alle andere neuronwaarden ook constant veranderen, waardoor het niet duidelijk is in welke richting de waarde van het neuron zou moeten veranderen. Een voorbeeld van dit probleem is het 'herd effect' [Fahlman]. Wanneer een netwerk wordt getraind op twee aparte deeltaken, dan hebben neuronen de neiging om eerst de ene taak volledig te leren, en daarna pas de andere taak. Maar bij het leren van die andere taak verleert het netwerk deels weer de eerste taak. Dit proces schommelt heen en weer totdat een deel van de neuronen zich toespitst op de ene taak en een andere deel op de andere taak. De 'herd' van neuronen heeft dus moeite om zich op te splitsen in kleinere groepen. Een ander probleem is het bepalen van de grootte van het netwerk, ook wel het 'bias/variance dilemma' genoemd [Geman]. Als het netwerk te klein is (weinig neuronen bevat) dan kan het alleen de meest algemene dingen leren. Als het netwerk te groot is (veel neuronen bevat) dan leert het alleen de specifieke trainingdata; het generaliseert dan juist te weinig. In de eerste situatie wordt de training van het netwerk nooit voltooid omdat het niet voldoende leert. In de tweede situatie wordt de training wel voltooid, maar presteert het netwerk alleen goed op de trainingsdata en slecht op alle nieuwe data.

Het concept 'Architecture learning' (zie bv. [Fahlman]) probeert deze problemen op te lossen door via algoritmen netwerken te vergroten of te verkleinen. Sommige onderzoeken die volgens dit concept werken, richten zich op het aanpakken van het bepalen van de gewenste

grootte van een netwerk, terwijl andere zich meer richten op het verbeteren van de leersnelheid. Zo wordt er in [Fahlman] het 'Cascade-Correlation' algoritme beschreven waarmee een netwerk opgebouwd wordt door hidden lagen, elk bestaand uit een enkel hidden neuron en zijn bijbehorende verbindingen, toe te voegen aan het netwerk. Wanneer een neuron wordt toegevoegd worden de inkomende verbindingen van de andere neuronen vastgezet (hun waarden kunnen niet meer veranderen). Uit het onderzoek bleek dat het algoritme 10 tot 50 maal sneller dan backpropagation was.

Een andere methode, het 'Dynamic Node Creation' algoritme, wordt beschreven in [Ash]. Daarbij worden hidden neuronen toegevoegd aan dezelfde hidden laag van het netwerk als, volgens het algoritme, de training van het netwerk niet snel genoeg meer verloopt. Elke keer dat een neuron is toegevoegd wordt het netwerk opnieuw getraind.

Het 'Grow and Learn' algoritme, beschreven in [Alpaydin], is nog een methode om de architectuur van een netwerk op te bouwen. Daarbij worden hidden neuronen toegevoegd aan het netwerk tijdens de training. Periodiek wordt het netwerk kleiner gemaakt door hidden neuronen te verwijderen die volgens dit algoritme niet noodzakelijk zijn, waardoor wordt voorkomen dat het netwerk te groot wordt.

Het algoritme dat in [Nabhan] is beschreven bouwt een nieuw netwerk op door uit een oud netwerk eerst verschillende nieuwe netwerken te maken, waaraan neuronen en lagen zijn toegevoegd. Het algoritme kiest dan het netwerk dat beter presteert dan de anderen, en hetzelfde proces wordt dan opnieuw toegepast op dit netwerk.

In [Setiono] wordt een algoritme beschreven dat 'cross validation' gebruikt voor het toevoegen van hidden neuronen aan een één-laags Feedforward neuraal netwerk. Alleen als het nieuwe netwerk met het toegevoegde neuron beter presteert op de trainingdata en een statistische 'cross validation' test dan het vorige netwerk, wordt de training voortgezet met dit nieuwe netwerk.

Nog een methode is 'EPNet', die beschreven is in [Yao1] en [Yao2]. Dit algoritme evolueert de architectuur van een neuraal netwerk: door genetische operatoren worden hidden neuronen of verbindingen aan het netwerk toegevoegd of verwijderd. Uit dit onderzoek blijkt het algoritme tot zeer goede resultaten te leiden, maar ook veel tijd in beslag te nemen. Ook blijkt dat EPNet tot netwerk architecturen leidt die nieuw en origineel zijn. Dit komt doordat het algoritme via een evolutionaire proces verloopt.

'MOST', het algoritme dat wordt beschreven in [Aran], bouwt een netwerk op door hidden neuronen of hidden layers zowel toe te voegen als te verwijderen. Er worden geen waarden vastgezet. Uit dit onderzoek bleek dat het algoritme bijna optimale oplossingen creëert, maar ook dit algoritme blijkt veel tijd in beslag te nemen.

Alle besproken onderzoeken manipuleren op de een of andere manier de architectuur van het netwerk. Daar hebben ze verschillende redenen voor. Bij sommigen ligt de focus op het automatisch creëren van de architectuur van een netwerk [Aran], zodat niet van tevoren het aantal benodigde neuronen hoeft te worden geschat. Anderen hopen een betere leersnelheid te behalen [Fahlman]. Bij het veranderen van de architectuur van het netwerk beperken zij zichzelf allemaal tot de hidden neuronen, in een enkele hidden laag dan wel verspreid over meerdere hidden lagen. In geen van de hiervoor genoemde onderzoeken worden input of output neuronen aan een netwerk toegevoegd of weggehaald.

Tegen deze achtergrond is er bij dit onderzoek voor gekozen om te onderzoeken of de leersnelheid van een neuronaal netwerk dat stapsgewijs wordt opgebouwd met input, hidden én output neuronen verbetert in vergelijking tot een niet stapsgewijs opgebouwd netwerk. Om dit te onderzoeken is een nieuw netwerkmodel ontworpen, het Incremental Neuraal Netwerk (INN). Het INN is een aanpassing van, en uitbreiding op, het concept 'Architecture Learning'. Aan dit netwerk kunnen neuronen worden toegevoegd, maar niet worden weggehaald. In tegenstelling tot de hiervoor genoemde modellen zijn de neuronen die worden toegevoegd niet gelimiteerd tot hidden neuronen, maar kunnen ook input en output neuronen worden toegevoegd. De architectuur van dit netwerk verandert niet automatisch op basis van een algoritme, maar neuronen worden toegevoegd wanneer gewenst door de onderzoeker. De focus van het onderzoek ligt immers niet op het automatisch creëren van een netwerk, maar op het verbeteren van de leersnelheid.

1.4 Onderzoeksvraag

Uit het literatuuronderzoek blijkt dat het stapsgewijze opbouwen met hidden neuronen van een neuraal netwerk vaak voordelen oplevert, zoals een verbeterde leersnelheid. Dit onderzoek richt zich nu op de vraag of de leersnelheid ook verbetert door het stapsgewijs uitbreiden van het netwerk met zowel input, hidden als output neuronen.

Daarvoor wordt het voor dit onderzoek ontwikkelde Incremental Neuraal Netwerk (INN) en het functioneren hiervan geëvalueerd door het te vergelijken met een standaard neuraal netwerk, namelijk het 'Sequential Neuraal Netwerk' (SNN), wat een specifieke implementatie is van het type 'Feedforward Neuraal Netwerk'. Het basisverschil tussen deze twee neurale netwerken is dat het SNN altijd een volledig netwerk is, d.w.z. gelijk in aantal neuronen en onderlinge verbindingen, terwijl het INN als een klein netwerk begint dat vervolgens stap voor stap wordt vergroot. De in het onderzoek gebruikte neurale netwerken worden verder uitvoerig beschreven in hoofdstuk 2.4.

Om het INN goed te kunnen evalueren worden de twee netwerken vergeleken in een dynamische omgeving onder invloed van evolutie. Dit heeft twee redenen: Ten eerste resulteert dit in een complexe en moeilijke leertaak. De verschillen tussen de twee netwerken zullen namelijk alleen duidelijk zichtbaar zijn als ze de kans krijgen om tot het uiterste te gaan. Ten tweede is de dynamische omgeving met evolutie een versimpelde versie van de echte wereld, waardoor de resultaten kunnen worden gegeneraliseerd naar de echte wereld.

Gebaseerd op de hiervoor beschreven eigenschappen van beide netwerken is de hypothese dat het INN betere resultaten oplevert dan het SNN. Het criterium dat wordt gehanteerd om deze stelling te toetsen is het aantal generaties dat nodig is om succes te behalen in een dynamische omgeving.

Om deze evaluatie te kunnen uitvoeren is het nodig te beschikken over:

- Een dynamische omgeving.
- Een implementatie van bovengenoemde neurale netwerken.
- Leermethodes die het evolutie proces simuleren.

In het volgende hoofdstuk, waarin wordt ingegaan op de onderzoeksopzet, wordt dit verder beschreven.

2 ONDERZOEKSOPZET

2.1 Introductie

Om te onderzoeken of het Incremental Neuraal Netwerk beter is dan het Sequential Neuraal Netwerk is er een dynamische wereld gecreëerd waarin deze worden getraind door evolutie te simuleren. Deze simulatie bestaat hieruit dat een populatie van creatures met hersenen die bestaan uit een van de neurale netwerken in een wereld worden geplaatst waarin ze acties kunnen ondernemen die van invloed zijn op het voortbestaan van de populatie.

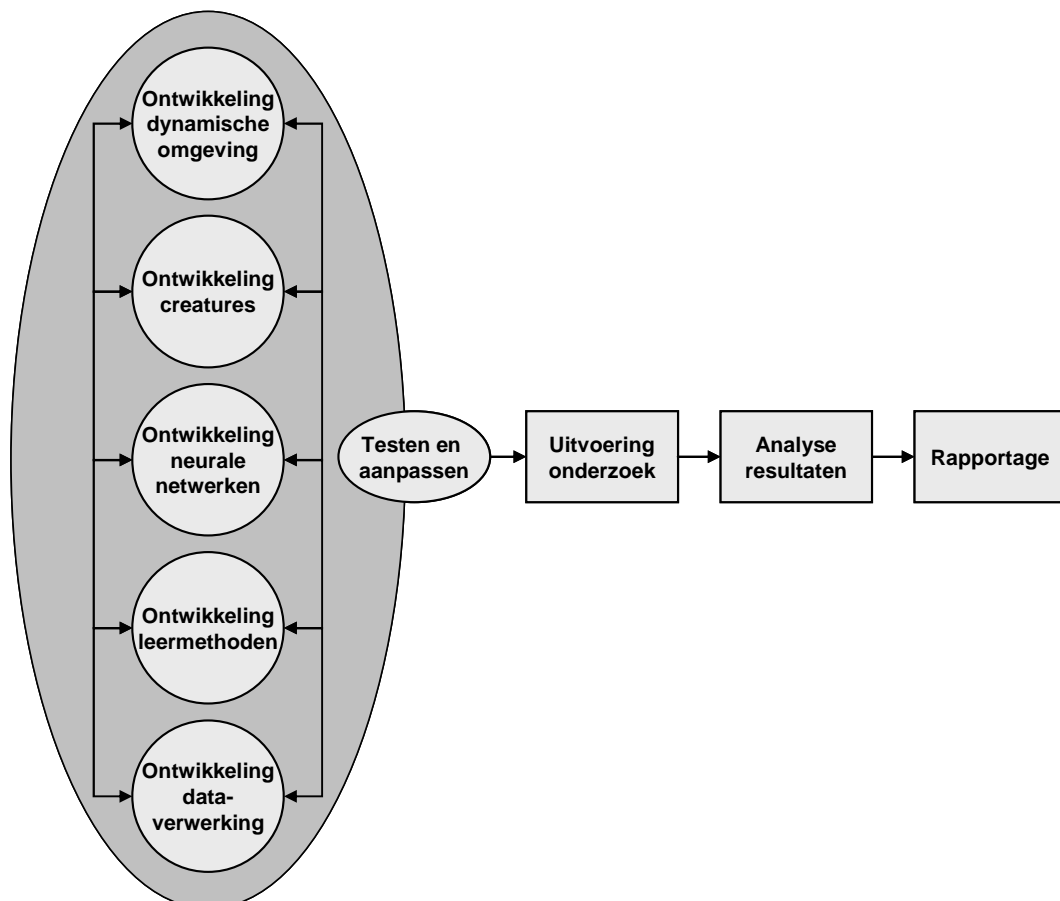
Er zijn door andere onderzoekers al eerder dynamische omgevingen ontworpen, vooral in het onderzoeksgebied 'Artificial Life', dat zich bezig houdt met het simuleren van leven. Voorbeelden van onderzoeken op dit gebied zijn 'Game of Life' [Gardner], 'Flocking' [Reynolds] en 'Artificial Ants' [Dorigo], vaak in redelijk simpele omgevingen. Meer dynamische omgevingen zijn bijvoorbeeld 'Core Wars', 'VENUS' en 'Tierra' [Yaeger 1]. Polyworld [Yaeger 1] [Yaeger 2] komt dicht in de buurt van de voor dit onderzoek gewenste soort dynamische omgeving, maar bleek om de volgende redenen toch minder geschikt:

- Om verschillende typen neurale netwerken met elkaar te vergelijken is er behoefte aan een uitdagende omgeving waarin een netwerk wordt getraind. Deze mogelijkheid biedt Polyworld niet, omdat daarin een willekeurig gegenereerde populatie van creatures direct tot een zichzelf in stand houdende populatie leidt.
- De voor dit onderzoek gewenste evolutionaire leermethoden worden toegepast op nodes van een neuraal netwerk, d.w.z. op een neuron en zijn bijbehorende waarden. Het evolutieproces in Polyworld is echter alleen toepasbaar op genen die zijn gecodeerd als lijsten van enen en nullen (voor meer over genen zie hoofdstuk 2.3).
- Dit onderzoek richt zich op het evolueren van een geheel netwerk, waarbij de informatie in de genen altijd tot hetzelfde neuraal netwerk leidt. In Polyworld echter staat in de genen bepaalde informatie waaruit een neuraal netwerk probabilistisch wordt gegenereerd, ofwel de architectuur van neurale netwerken wordt geëvolueerd.
- Dit onderzoek richt zich op twee netwerken van het type Feedforward Neuraal Netwerk, die niet tijdens hun leven leren. Neurale netwerken in Polyworld leren echter tijdens hun leven als creature in de omgeving, waarbij Hebbian Learning [Gerstner] wordt toegepast op Recurrent Neurale Netwerken.
- In een realistische omgeving nemen objecten een deel van de omgeving in. In Polyworld echter kunnen verschillende objecten dezelfde plek in de omgeving innemen. Dit is zelfs een vereiste voor verscheiden acties.

- In dit onderzoek is het de bedoeling dat elke actie een keuze is van de creatures, d.w.z. elke actie is een gekozen actie. Creatures in Polyworld bewegen echter altijd. Zij kiezen niet voor een beweegactie, maar alleen *hoe snel* ze bewegen.
- Een belangrijk element van handelingen, of acties, is dat ze een tijdsduur hebben. Acties in Polyworld hebben echter geen tijdsduur.

Omdat geen van de beschikbare dynamische omgevingen aan de vereisten van het voorgenomen onderzoek voldeden, is er een ontwikkeld. Niet alleen de omgeving, maar ook de creatures met hersenen die bestaan uit de gekozen neurale netwerken en de leermethoden voor deze creatures moesten worden ontwikkeld, alsook methoden voor dataverwerking en analyse. Daarop aansluitend kunnen in het onderzoek verscheidene stappen worden onderscheiden, die in figuur 2 schematisch zijn weergegeven. Vooral in de beginfase was er sprake van een iteratief proces waarbij voor het onderzoek nodige systemen en onderdelen werden ontwikkeld, getest en aangepast totdat de vereiste kwaliteit werd behaald om het onderzoek uit te voeren.

Het programma dat het raamwerk vormt voor de ontwikkeling van deze onderdelen is geleverd door het bedrijf MeinMein en technische ondersteuning hierbij is geleverd door Meindert Kamphuis. Omdat de onderzoeksopzet binnen dit raamwerk is ontwikkeld, is deze ook van praktisch nut voor het bedrijf.



Figuur 2 Schematische weergave van onderzoeksopzet

2.2 Ontwikkeling van dynamische omgeving

2.2.1 Concept

Om de neurale netwerken te kunnen evalueren op hun functioneren in een dynamische omgeving, moet een dynamische omgeving beschikbaar zijn. Een omgeving is dynamisch als het aan veranderingen onderhevig is, d.w.z. als eigenschappen van de omgeving en objecten van of in deze omgeving veranderen, bijvoorbeeld ontstaan, verdwijnen en verplaatsen. Binnen het kader van dit onderzoek is een omgeving ontworpen met een beperkt aantal dynamische aspecten. Deze omgeving is tweedimensionaal en is discreet. De omgeving is niet driedimensionaal gekozen, zoals in [Reynolds], omdat dit het onderzoek te complex zou maken. In een driedimensionale omgeving zouden creatures (zie hoofdstuk 2.3) namelijk meer zintuigen en acties nodig hebben omdat er meer richtingen mogelijk zijn. Ook is ervoor gekozen om de omgeving discreet te maken door het uit velden te laten bestaan, in tegenstelling tot bijvoorbeeld Polyworld [Yaeger 1] [Yaeger 2] waar de omgeving continu is. Dit is gedaan omdat een continue omgeving de complexiteit van de omgeving niet sterk vermeerdert, maar het creëren ervan wel erg complex is.

De dynamiek in de omgeving komt tot stand door veranderingen in de tijd, van de ruimte zelf en van de daarin geplaatste creatures. Door het tijdsverloop veranderen ruimtelijk objecten van plaats of eigenschap. Creatures zijn onderhevig aan tijd, niet alleen omdat hun acties een bepaalde tijd duren, maar ook omdat ze constant energie nodig hebben om te blijven bestaan. Naast de tijd als dynamische factor is ook het feit dat creatures elkaar en de omgeving beïnvloeden een factor die de dynamiek vergroot.

De tijd als dynamische factor betekent dat een creature, of meer precies zijn neurale netwerk, niet kan worden beoordeeld op zijn prestatie op één bepaald punt in zijn leven, maar dat zijn gehele leven in beschouwing moet worden genomen om een oordeel te kunnen vellen. Het gaat er immers niet om of een stap naar links of rechts op een bepaald moment positief of negatief is, maar of het totaal van acties tot een positief of negatief resultaat leidt, naar het oordeel van de onderzoeker.

2.2.2 Beschrijving

De dynamische omgevingen waarin creatures zich bevinden worden in dit onderzoek 'werelden' genoemd. Zoals nader wordt uitgelegd in hoofdstuk 3 zijn er verschillende werelden ontworpen die op bepaalde punten verschillen, maar elke wereld is opgebouwd uit dezelfde basiseigenschappen en objecten.

De basiseigenschappen van de in het onderzoek gebruikte werelden zijn:

- Een wereld is opgebouwd uit velden.
- Een wereld is x bij y aantal velden groot; ongeveer als een dambord, maar hoeft niet vierkant te zijn.
- Op elk veld kan maar 1 object aanwezig zijn.
- Een wereld is eindig en wordt omgeven door een muur.
- Een wereld verandert in de tijd.
- Het verloop van tijd wordt geteld in iteraties; elke iteratie is een stapje vooruit in tijd.

De objecten die in de ontworpen werelden voorkomen, zijn:

- Wall blokjes
 - Wall blokjes zijn objecten die muren en obstakels voorstellen.
 - Wall blokjes zijn statisch; ze kunnen niet opzij worden geduwd, opgegeten worden of verdwijnen.
- Food blokjes
 - Food blokjes zijn objecten die voedsel bevatten dat door de creatures kan worden opgenomen waardoor ze van energie worden voorzien.
 - Food blokjes bevatten een bepaalde hoeveelheid voedsleenheden.
 - De maximale hoeveelheid voedsel per blokje is 1000 voedsel eenheden.
 - De beschikbare hoeveelheid voedsel neemt af met 1 eenheid per iteratie.
 - Als het aantal voedsleenheden van een blokje gelijk is aan 0, dan verdwijnt het.

In het eerste ontwerp van de werelden waren meer objecten opgenomen zoals gevaren, lava blokjes die creatures doden, en water blokjes die samen met food blokjes een ecosysteem vormden. Deze objecten zijn uiteindelijk niet in het onderzoek meegenomen. Zie voor verdere informatie hierover hoofdstuk 2.7.

2.3 Ontwikkeling van 'creatures'

2.3.1 Concept

Om neurale netwerken in een dynamische omgeving te kunnen trainen en evalueren, moeten ze deze omgeving op de een of andere manier kunnen observeren en beïnvloeden. Dit wordt bereikt door de netwerken een 'vehikel' te geven waarover ze controle hebben. Datgene wat een vehikel observeert is dan de input voor het neurale netwerk en de output van het neurale netwerk bepaalt de handelingen die het vehikel onderneemt. Het neurale netwerk kan dus worden beschouwd als de hersenen van een levend wezen. Het leerproces waarmee deze neurale netwerken worden getraind is een simulatie van het natuurlijke evolutieproces.

Om een omgeving te kunnen beïnvloeden moet een vehikel 'fysiek' aanwezig zijn in deze omgeving. Daarom heeft het een 'lichaam' nodig dat bepaalde eigenschappen bezit. Om natuurlijke selectie te simuleren moeten deze lichamen dood kunnen gaan door honger of ouderdom. Met een dergelijk lichaam kan gedrag van het vehikel als een maatstaf worden genomen voor het evalueren van het betrokken neurale netwerk.

Het vehikel heeft niet alleen een lichaam nodig, maar ook een manier om de omgeving te observeren en daarop te reageren. Het heeft dus bepaalde 'zintuigen' nodig. Verder moet een vehikel bepaalde handelingen kunnen verrichten, om zijn omgeving te kunnen beïnvloeden; ze moeten 'acties' kunnen uitvoeren. In de volgende tabel zijn de acties die de creatures in dit onderzoek kunnen ondernemen, summier beschreven.

| | Vereiste Inspanning | Benodigde Tijdsduur | Redenering |
|---------------------|----------------------------|----------------------------|--|
| Bewegen | Laag | Kort | Een creature heeft de vrijheid nodig om zijn omgeving te verkennen. Hierom moet bewegen worden gepromoot en niet afgestraft. |
| Eten | Middel | Middel | Eten verzamelen en consumeren kost tijd en moeite. |
| Reproduceren | Zeer Hoog | Lang | Het maken en voortbrengen/opvoeden van nakomelingen kost veel inspanning en tijd. |

Tabel 2 Samenvatting van kenmerken en doelen van werelden in dit onderzoek

Aangezien het leerproces via een evolutionair geïnspireerd systeem verloopt, is er iets nodig waarop de evolutionaire leermethoden kunnen worden toegepast. Dit 'iets' moet alle informatie bevatten die de evolutionaire leermethoden nodig hebben en het moet uniek zijn

voor elk vehikel. Een veelgebruikte en handige analogie uit de natuurlijke wereld zijn de 'genen'.

Deze vier basis onderdelen (lichaam, zintuigen, acties, genen) vormen samen een vehikel, waarmee een neurale netwerk kan functioneren in een omgeving zoals een levend wezen met hersenen in een natuurlijke omgeving. Vanwege deze analogie met de natuur, wordt dit vehikel in dit onderzoek een 'creature' genoemd.

2.3.2 Beschrijving

De verschillende delen waaruit de creatures in dit onderzoek zijn opgebouwd worden hier achtereenvolgens beschreven.

a. Lichaam

Om in een dynamische omgeving te kunnen bestaan en er invloed op te kunnen uitoefenen, hebben de neurale netwerken, of hersenen, een lichaam nodig. In dit onderzoek heeft het lichaam de volgende eigenschappen en vaardigheden.

- Voorkant/Achterkant
 - Creatures hebben een voorkant, achterkant, linker zijkant en rechter zijkant.
 - Acties worden uitgevoerd op het veld aan de voorkant van een creature.
- Oriëntatie
 - Aangezien de omgeving, zoals beschreven in hoofdstuk 2.2, uit velden bestaat soortgelijk aan een dambord, zijn er 4 richtingen mogelijk.
 - De oriëntatie is gelijk aan de richting waarnaar de voorkant van een creature wijst.
 - Deze richtingen zijn het noorden, oosten, zuiden of westen.
- Maximum leeftijd
 - Lichamen zijn niet onsterfelijk. Ze worden oud en sterven. Het verouderingsproces zelf wordt in dit onderzoek buiten beschouwing gelaten, maar er is wel een maximum leeftijd ingesteld. Als een creature deze leeftijd haalt, dan sterft het. Dit is gedaan om de evolutie te simuleren door resources (voedsel, plek in de wereld) vrij te maken voor nieuw geboren creatures.
 - De maximum leeftijd is 5000 iteraties.
- Huidige leeftijd
 - De huidige leeftijd geeft aan hoe oud een creature is. Wanneer een creature wordt geboren staat dit getal op 0. De leeftijd neemt elke iteratie met 1 toe.

- Maximum energie niveau
 - Hoe lang een lichaam kan functioneren zonder opnieuw voedsel te consumeren wordt gesimuleerd door een energiesysteem. Elke eenheid voedsel dat wordt gegeten wordt omgezet naar een (even grote) interne energie eenheid. Deze energie is nodig om in leven te blijven en om acties te ondernemen. Lichamen zijn echter niet oneindig groot. Daarom kunnen ze maar een maximale hoeveelheid energie bevatten.
 - De maximum hoeveelheid energie is in dit onderzoek 1000 eenheden.
- Huidig energie niveau
 - Het huidige energie niveau geeft aan over hoeveel energie een creature op een bepaald moment beschikt. Dit is minimaal 1 en maximaal 1000. Als de creature eet terwijl het huidige energie niveau 1000 is dan wordt het energie niveau niet hoger en gaat dat eten dus verloren. Als het energieniveau 0 is dan sterft de creature van honger.
- Leefkosten
 - Om te simuleren dat een lichaam en hersenen energie verbruiken om te leven en te werken, heeft een lichaam leefkosten. Dit kost een creature elke iteratie 1 energie eenheid.

Sommige van deze eigenschappen hingen in het begin van dit onderzoek af van andere eigenschappen, op basis van de informatie opgeslagen in de genen van de creatures, zoals maximum leeftijd en maximum energieniveau. Zo was de snelheid van acties (aantal iteraties dat een actie duurt) afhankelijk van maximum energieniveau. Om praktische redenen zijn echter al deze eigenschappen statisch gemaakt.

Verder konden de creatures in het begin van het onderzoek nog andere eigenschappen hebben, zoals een kleur en een houding, maar uiteindelijk zijn deze niet meer gebruikt in het onderzoek. Voor meer informatie hierover zie hoofdstuk 2.7.

b. Zintuigen

Voor het observeren van de omgeving bezitten creatures zintuigen. Elke creature heeft vijf zintuigen die functioneel hetzelfde zijn, maar elk een andere observatierichting hebben. Van deze zintuigen worden er vier gebruikt om de omgeving mee te observeren en het laatste zintuig is naar binnen gericht.

De vier zintuigen die de omgeving observeren hebben elk een bepaalde observatierichting, namelijk naar voren, naar achteren, naar links of naar rechts. De richting van deze zintuigen is relatief ten opzichte van de oriëntatie van de creatures. Dit betekent dat als een creature naar het noorden georiënteerd staat, de observatierichting van zijn linkerzintuig westelijk is

en die van zijn rechterzintuig oostelijk, maar als zijn oriëntatie zuidelijk is dan is de observatierichting van zijn linkierzintuig oostelijk en die van zijn rechterzintuig westelijk, etc. Omdat creatures in dit onderzoek niet over de eigenschap beschikken om tijdens hun leven te leren - bijvoorbeeld door hun neurale netwerk aan te passen - hebben ze in feite geen geheugen en hebben ze dus geen informatie over hun eigen lichaam. Om te kunnen beslissen welke actie een creature het best kan ondernemen, is informatie over zichzelf echter heel belangrijk. Daarvoor dient het laatste zintuig. Met dat zintuig observeert de creature zichzelf en beschikt het over evenveel informatie als een andere creature dat hem observeert. Als hij bijvoorbeeld observeert dat zijn energieniveau laag is, dan kan hij op basis daarvan besluiten om voedsel te gaan eten. Een ander creature kan dat energieniveau ook zien en concluderen dat de creature op zoek zou moeten gaan naar voedsel. Door dit laatste zintuig krijgt de creature dus informatie over zijn eigen lichaam en omdat de toestand van zijn lichaam is bepaald door zijn vorige acties, kan de creature dus in feite terug kijken in de tijd. Hoewel dit laatste zintuig een unieke observatierichting heeft, is het type en het detail van informatie dat het observeert gelijk aan die van de andere zintuigen.

Elk zintuig verzamelt vijf informatie-elementen. Het eerste informatie-element wordt verzameld door het veld direct naast de creature waar te nemen:

- Hoe lang geleden op dit veld een creature aanwezig is geweest:
 - Dit wordt bepaald door 'feromonen' waar te nemen, die langzaam in sterkte afnemen.
 - Elke creature laat 'feromonen' achter op elk veld waarop het aanwezig is.
 - Er is geen onderscheid tussen de feromonen van de creature zelf of van andere creatures.

Dan kijkt het zintuig in een rechte lijn steeds verder van de creature af totdat het een object ziet. De informatie-elementen dat het zintuig dan verzamelt, is:

- De afstand tot het object als het object een wall blokje is.
- De afstand tot het object als het object een food blokje is.
- De afstand tot het object als het object een creature is.
- Hoeveel er in het object zit:
 - Hoeveel energie als het object een creature is,
 - hoeveel voedsel als het object een food blokje is en
 - leeg als het een wall blokje is.

c. Acties

De manieren waarop de creatures de omgeving, en zichzelf, kunnen beïnvloeden worden in dit onderzoek 'acties' genoemd. Ze kunnen in totaal zes acties ondernemen; vier om te

bewegen, één om voedsel te eten en één om nakomelingen te voort te brengen. Elke actie heeft een aantal kenmerken: (i) het kost een bepaalde hoeveelheid energie, (ii) het duurt een bepaalde tijd om te voltooien en (iii) het heeft een bepaald resultaat. Acties kunnen ook mislukken; dan wordt het gewenste resultaat niet behaald, zijn de energiekosten lager en is de tijdsduur van de actie korter.

Alle acties die creatures ondernemen, kosten energie en ze moeten voldoende energie hebben om die acties te kunnen ondernemen. Als ze geen energie meer hebben, sterven ze direct. De basis energiekosten van acties is de hoeveelheid energie die nodig is om met een bepaalde actie te beginnen. Die energie wordt direct verbruikt en als er dan geen energie meer over is, sterft de creature ook direct.

Elke actie heeft een bepaalde tijdsduur waarin geen andere acties kunnen worden begonnen. Het resultaat van een actie ontstaat geleidelijk verspreid over deze tijdsduur. Acties kunnen mislukken als er niet aan bepaalde voorwaarden wordt voldaan, bijvoorbeeld als een creature de actie 'eat' uitvoert terwijl er geen voedsel aanwezig is.

Als een actie is mislukt en dus geen resultaat oplevert, dan is er altijd een vaste rustperiode van acht iteraties voordat er weer een actie kan worden ondernomen.

Overzicht van acties in dit onderzoek:

- Move Forward
 - Basis energiekosten: 1 eenheid
 - Tijdsduur: 8 iteraties
 - Voorwaarde: Het veld direct voor de creature moet leeg zijn.
 - Resultaat: De creature verplaatst zich 1 veld naar voren.
- Move Backward
 - Basis energiekosten: 1 eenheid
 - Tijdsduur: 8 iteraties
 - Voorwaarden: Het veld direct achter de creature moet leeg zijn.
 - Resultaat: De creature verplaatst zich 1 veld naar achteren.
- Turn Left
 - Basis energiekosten: 1 eenheid
 - Tijdsduur: 8 iteraties
 - Voorwaarden: Geen
 - Resultaat: De creature draait een kwartslag naar links.
- Turn Right
 - Basis energiekosten: 1 eenheid
 - Tijdsduur: 8 iteraties

- Voorwaarden: Geen
- Resultaat: De creature draait een kwartslag naar rechts.
- Eat
 - Basis energiekosten: 5 eenheden
 - Tijdsduur: 16 iteraties
 - Voorwaarden: Er moet een 'food' blokje aanwezig zijn in het veld direct voor de creature.
 - Resultaat: De hoeveelheid energie die de creature krijgt is maximaal 20 maal de basis energiekosten (= 100). Als er minder voedsel aanwezig is dan krijgt de creature ook minder. De hoeveelheid voedsel in het food blokje neemt evenredig af en het food blokje verdwijnt als het geen voedsel meer bevat.
- Mate
 - Basis energiekosten: 10 eenheden
 - Tijdsduur: 40 iteraties
 - Voorwaarden:
 - Er moet een andere creature aanwezig zijn in het veld direct voor de creature.
 - Er moet minstens één leeg veld zijn rondom de twee 'ouders'.
 - De andere creature moet dezelfde basis energiekosten (10) leveren.
 - Daarnaast moeten beide creatures 150 energie eenheden bezitten om de mogelijke nakomeling van een starthoeveelheid energie te voorzien.
 - Als niet aan bovenstaande voorwaarden wordt voldaan, mislukt de actie en komen beide creatures in de faalstand te staan, die 8 iteraties duurt.
 - Resultaat: Er wordt een nieuwe creature, de nakomeling, in één van de velden rondom de twee ouders geplaatst. De genen van de nakomeling zijn samengesteld uit de genen van beide ouders. De hoeveelheid energie van de nakomeling is de extra energie die beide ouders overdragen, ofwel $2 \times 150 = 300$.
- Push
 - Push is een actie die niet echt kan worden gekozen door een creature maar die optreedt zodra er een object in de weg staat bij de 'move forward' en 'move backward' acties (zie boven). De beweegactie wordt dan omgezet in een push actie, waarbij de creature zelf niet beweegt.
 - Energie kosten: 10 eenheden
 - Tijdsduur: 24 iteraties

- Voorwaarden:
 - Het veld waar naartoe het object wordt geduwd moet leeg zijn.
 - Het object moet duwbaar zijn (bv. wall blokjes zijn niet duwbaar).
- Resultaat: Het object wordt verplaatst naar het veld waarnaar het werd geduwd.

d. Genen

In dit onderzoek zijn creatures onderhevig aan een nagebootst evolutie proces; de neurale netwerken leren door middel van evolutionaire algoritmen. Dit soort evolutionaire algoritmen worden vaak toegepast op 'genen' waarop het neurale netwerk op een bepaalde manier staat gecodeerd [Belew]. Deze codering is meestal een lijst van enen en nullen. Hierop wordt dan met een bepaalde kans mutatie toegepast, waarbij een één in een nul verandert en andersom. En op twee genen, bijvoorbeeld van twee ouders, kan crossover worden toegepast om een nieuw gen te maken, waarbij crossover betekent dat de lijst van enen en nullen op een of meer plaatsen doorgeknipt wordt en de delen worden uitgewisseld. Dit is een versimpelde versie van wat er gebeurt met DNA in een natuurlijk evolutie proces. Deze genen kunnen een neurale netwerk compleet specificeren of slechts gedeeltelijk [Yao 1]. Als het netwerk compleet is gespecificeerd, dan bepalen de genen het gehele neurale netwerk d.w.z. zowel zijn gehele architectuur (hoeveel neuronen er zijn, welk type het zijn en op welke manier ze met elkaar zijn verbonden) als de bijbehorende waarden (de bias van de neuronen en de gewichten van de verbindingen). Als het netwerk gedeeltelijk is gespecificeerd in de genen, dan wordt deze informatie gebruikt om probabilistisch een netwerk te genereren dat nooit precies hetzelfde is. In de genen staat dan, bijvoorbeeld, hoeveel neuronengroepen er zijn, uit hoeveel neuronen een groep bestaat en hoeveel verbindingen er bestaan tussen deze groepen, maar niet welk neuron met welk ander neuron is verbonden en wat de gewichten zijn van deze verbindingen.

De manier waarop een neurale netwerk wordt gecodeerd op de genen kan effect hebben op de effectiviteit van de gekozen evolutionaire leermethoden. Zo wordt er gewaarschuwd in [Belew] dat, wanneer er wordt gekozen voor een codering tot een lijst van enen en nullen, er vooral bij crossover problemen zijn. Als bijvoorbeeld twee waarden in combinatie met elkaar een bepaald positief effect hebben op het functioneren van het netwerk en ze op plekken in de genen worden gecodeerd die ver van elkaar af liggen, dan is de kans groot dat deze waarden bij crossover van elkaar worden gescheiden en dit effect verloren gaat. Hierdoor zal het langer duren voor een evolutionair proces om tot een bepaald resultaat te komen. Om deze problemen te voorkomen, is er in dit onderzoek voor gekozen om het neurale netwerk niet te coderen tot een lijst van enen en nullen, maar lijsten bij te houden van het

gehele neurale netwerk (de neuronen, hun bias, verbindingen en bijbehorende gewichten) en zijn er leermethoden gekozen die hier goed op opereren (zie hoofdstuk 2.5). Ofwel, in dit onderzoek bestaan de genen van een creature uit:

- Het aantal neuronen in de input laag van het netwerk.
- Het aantal neuronen in de hidden laag van het netwerk.
- Het aantal neuronen in de output laag van het netwerk.
- De bias van elke hidden neuron.
- De bias van elke output neuron.
- De verbindingen tussen de input en de hidden laag.
- De verbindingen tussen de hidden en de output laag.
- De gewichten van alle verbindingen.

De genen bevatten geen informatie over het lichaam van de creature; het lichaam van elke creature is hetzelfde.

In tegenstelling tot veel andere onderzoeken in een dynamische omgeving beschrijven de genen in dit onderzoek dus niet slechts de architectuur van het neurale netwerk waarmee een neuraal netwerk kan worden gegenereerd, maar een geheel neuraal netwerk. Andere onderzoeken in een niet-dynamische omgeving gebruiken wel vaak een gehele neuraal netwerk beschrijving gecodeerd tot een lijst van enen en nullen (bijvoorbeeld bij onderzoek waar leermethoden worden onderzocht met behulp van bepaalde standaard datasets, zoals in [Yao 2]).

In het begin van dit onderzoek stond er veel meer informatie in de genen, zoals maximum energie capaciteit, aantal zintuigen, etc., maar dit bleek bij nader inzien te complex om mee te laten evolueren, zodat uiteindelijk alleen het neuraal netwerk deel van de genen is gebruikt. Voor meer informatie, zie hoofdstuk 2.7.

2.4 Ontwikkeling van neurale netwerken

2.4.1 Concept

Zoals in hoofdstuk 1.4 is beschreven, richt dit onderzoek zich op het vergelijken van een stapsgewijs opbouwend neurale netwerk met een standaard neurale netwerk, respectievelijk het nieuw ontwikkelde Incremental Neuraal Netwerk (INN) en het Sequential Neuraal Netwerk (SNN). Beide zijn gebaseerd op het Feedforward Neuraal Netwerk. Het belangrijkste verschil is dat aan het INN input, hidden en output neuronen kunnen worden toegevoegd, terwijl het SNN niet van omvang verandert. Zoals eerder is beschreven in hoofdstuk 1.3, is de groei van het INN niet beperkt tot het toevoegen van alleen hidden neuronen, zoals bij andere modellen, maar kunnen er ook input en output neuronen aan het INN worden toegevoegd. Het Sequential Neuraal Netwerk doorloopt de werelden (zie hoofdstuk 3) zonder van architectuur te veranderen; alleen de waarden van het netwerk veranderen door middel van de gebruikte leermethoden (zie hoofdstuk 2.5). De architectuur van het voor dit onderzoek ontwikkelde Incremental Neuraal Netwerk verschilt echter wel per wereld. Er worden namelijk in elke wereld delen aan het netwerk toegevoegd totdat het in wereld 4 dezelfde architectuur heeft als het Sequential Neuraal Netwerk. Het toevoegen van neuronen gebeurt niet volgens een bepaald algoritme, maar wordt handmatig bepaald.

Het vastzetten van neuronen voordat nieuwe neuronen worden toegevoegd, zoals in [Fahlman], is ook in dit INN opgenomen. Het concept volgens [Fahlman] is dat een vastgezette neuron bepaalde kennis elementen bevat, en nieuw toegevoegde neuronen deze elementen kunnen gebruiken bij opbouw van meer complexe kennis constructies. Bij het INN zoals het wordt gebruikt in dit onderzoek is dit te vergelijken met simpel gedrag leren (voedsel vinden) en hierop voortbouwen tot complexer gedrag (voedsel vinden én eten). Het vastzetten van neuronen in het INN werkt niet zo simpel als in [Fahlman]. Ten eerste wordt een neuron nooit volledig vastgezet, waardoor er een zeer kleine kans blijft bestaan dat zijn waarden veranderen. Daardoor kan ook simpel gedrag, over lange tijd, veranderen als dit voordeel oplevert. Ten tweede kunnen er verbindingen bij komen met al vastgezette neuronen (doordat ook input en output neuronen kunnen worden toegevoegd), wat niet bij [Fahlman] voorkomt. Dit zou geleerde kennis, zoals voedsel vinden, snel kunnen worden verstoord. Daarom krijgen deze verbindingen een lagere kans om te veranderen dan normaal. De precieze kansen hiervan en hun werking staan beschreven in hoofdstuk 2.5.

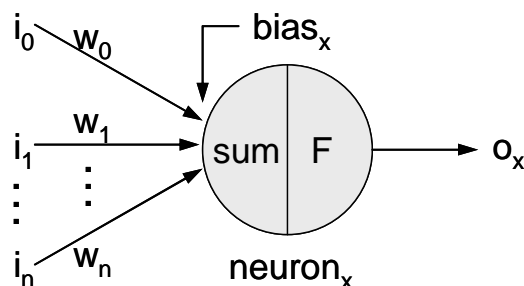
De netwerken leren niet tijdens hun leven als creature (zie hoofdstuk 2.3). Dit is omdat er in [Nolfi] voor wordt gewaarschuwd dat leren tijdens leven in combinatie met evolutionaire algoritmen zeer complex is en gemakkelijk leidt tot minder voortgang in het leren. Bovendien zou dit extra variabelen introduceren waardoor het onderzoek onnodig complexer zou worden.

2.4.2 Beschrijving

De twee gebruikte neurale netwerken in dit onderzoek, gebaseerd op het Feedforward Neuraal Network, zijn het Sequential Neuraal Network (SNN) en het Incremental Neuraal Network (INN). Deze worden hier achtereenvolgens beschreven.

Feedforward Neuraal Network

Het Feedforward Neuraal Network wordt ook wel een 'Multi-layer perceptron' genoemd: de neuronen in het netwerk zijn namelijk perceptronen die met elkaar verbonden zijn. In figuur 3 is schematisch een dergelijk neuron weergegeven. Het bestaat uit het neuron zelf (neuron_x), verbindingen die bepaalde input leveren aan het neuron ($i_0 \dots i_n$), gewichten van deze verbindingen ($w_0 \dots w_n$), de bias waarde van het neuron (bias_x), een combinatiefunctie (sum), een activatiefunctie (F) en de output van het neuron (o_x).

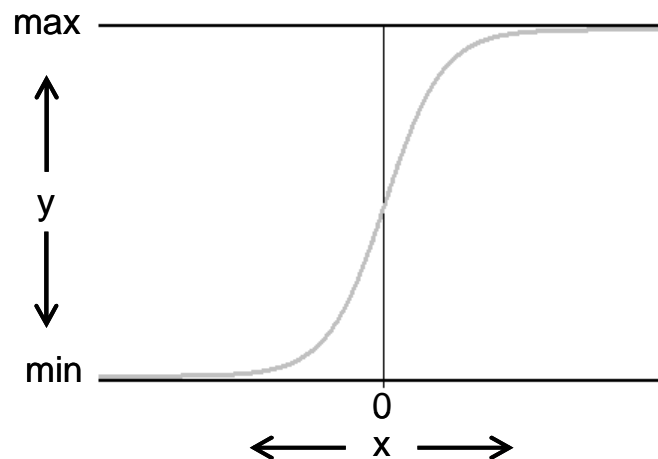


Figuur 3 Schematische weergave van een neuron

De output van het neuron wordt op de volgende manier bepaald:

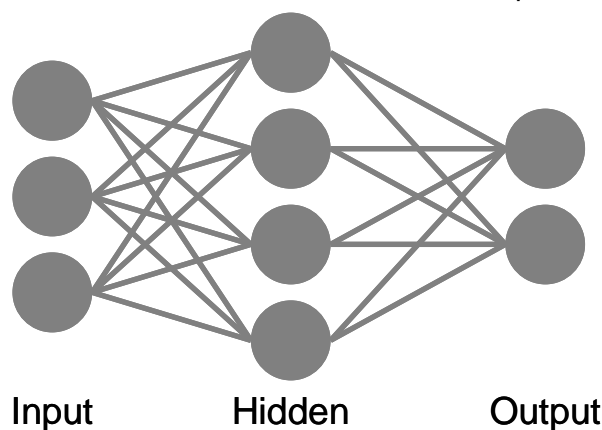
- De combinatiefunctie; deze berekent de totale input van het neuron.
 - **sum** = $\sum (i_n \times w_n) + \text{bias}$
 - Dit is de 'weighted sum' van de inputs. Elke input wordt vermenigvuldigd met het bijbehorende gewicht van de verbinding en gesommeerd, waarna de bias erbij op wordt geteld.
 - Er zijn andere combinatiefuncties mogelijk, maar de 'weighted sum' is het meest gebruikt voor Feedforward Neurale Netwerken.
- De activatiefunctie; deze berekent de activiteit van het neuron.
 - Er zijn verschillende soorten activatiefuncties, waarvan twee van de meest gebruikte van het type 'sigmoid' functies zijn.

- De hyperbolische tangens $F(\text{sum}) = \tanh(\text{sum})$ die varieert van -1 tot 1.
- De logistische functie $F(\text{sum}) = 1 / (1 + e^{-\text{sum}})$ die varieert van 0 tot 1.
- Deze functies hebben een distinctieve 'S' vorm, zoals wordt weergegeven in figuur 4. Hoe hoger (lager) de x, hoe dichter de y het max (min) nadert.
- De output van het neuron is dan $\text{out} = F(\text{sum})$



Figuur 4 Schematische weergave van een sigmoid functie

Een Feedforward Neuraal Network is opgebouwd uit een aantal opeenvolgende 'layers', of lagen, bestaande uit deze neuronen. De verschillende opeenvolgende lagen zijn een input laag, één of meer hidden lagen en een output laag. Neuronen binnen een laag zijn niet met elkaar verbonden; elke neuron in een laag is verbonden met elke neuron in de opvolgende laag (zie figuur 5). De neuronen in de input laag worden input neuronen genoemd, neuronen in de hidden laag(en) hidden neuronen en neuronen in de output laag output neuronen.



Figuur 5 Schematische weergave van een 3-laags Feedforward Neuraal Network

De output van een input neuron is gelijk aan zijn input. Deze input is een waarde die aan het netwerk wordt gegeven. Het aantal waarden waarop een netwerk wordt uitgevoerd bepaalt daarom het aantal benodigde input neuronen.

Voor de neuronen in de hidden en output laag wordt dezelfde combinatiefunctie gebruikt, namelijk 'weighted sum', maar verschillende activatiefuncties. De hidden neuronen gebruiken de hyperbolische tangens, terwijl de output neuronen het 'softmax' algoritme gebruiken:

- Softmax bestaat uit de volgende stappen, waar n = neuron n , sum_n = weighted sum van neuron n en o_n = output van neuron n .
 - $i_n = e^{\text{sum}_n}$
 - $\text{totaal} = \sum i_n$
 - $o_n = \text{totaal} / i_n$
- Ofwel, de output waarden variëren van 0 tot 1, maar zijn relatief aan de input van de andere neuronen in dezelfde laag.

Sequential Neuraal Netwerk

Het Sequential Neuraal Netwerk is een Feedforward Neuraal Netwerk dat de verschillende delen, of werelden, in dit onderzoek sequentieel doorloopt (zie hoofdstuk 3) zonder dat zijn architectuur verandert. Zijn waarden, de gewichten van verbindingen en bias van neuronen, veranderen uiteraard wel. De initiële waarden van een nieuw gegenereerd netwerk (gewichten van verbindingen en bias van neuronen) zijn normaal verdeeld met $\mu = 0$ en $\sigma = 0.5$ (zie hoofdstuk 1.2).

De architectuur van het Sequential Neuraal Netwerk is als volgt:

- 1 input laag bestaand uit 25 neuronen.
 - 5 informatie elementen per zintuig, 5 zintuigen; $5 \times 5 = 25$ (zie hoofdstuk 2.3)
- 1 hidden laag bestaand uit 25 neuronen.
- 1 output laag bestaand uit 6 neuronen.
 - 6 mogelijke acties (zie hoofdstuk 2.3)

De input neuronen geven de verzamelde informatie van de zintuigen door aan de rest van het netwerk. De informatie is gecodeerd in waarden tussen de 1 en -1. De informatie elementen per zintuig, zoals beschreven in (hoofdstuk 2.3) zijn:

- Hoe lang geleden op het veld een creature aanwezig is geweest (feromoon).
 - $2 \times 0.5^{N \times 0.01} - 1$ waarbij N = het aantal iteraties sinds hier een creature aanwezig is geweest.
 - Ofwel nabij 1 is kort geleden en nabij -1 is lang geleden.
- De afstand tot het object als het object een wall blokje is.
 - $2 \times 0.5^{N \times 0.2} - 1$ waarbij N = de afstand tussen de creature en het object.
 - Ofwel nabij 1 is dichtbij en nabij -1 ver weg.

- De afstand tot het object als het object een food blokje is.
 - Hetzelfde als bij het wall blokje.
- De afstand tot het object als het object een creature is.
 - Hetzelfde als bij het wall blokje.
- Hoeveel het object bevat (voedsel of energie).
 - $\tanh(N * 1 - 2)$ waarbij N = de hoeveelheid gedeeld door de maximum hoeveelheid.
 - Ofwel nabij 1 is veel en nabij -1 is weinig.

De output neuron die de hoogste output waarde heeft, is de winnaar. De winnaar van de zes output neuronen is de actie die zal worden uitgevoerd. De output neuronen 1 t/m 6 staan respectievelijk voor de acties 'move forward', 'move backward', 'turn left', 'turn right', 'eat' en 'mate'.

Het Sequential Neuraal Netwerk heeft de standaard mutatie kans. Deze kans wordt gebruikt door de mutatie operator. (zie hoofdstuk 2.5)

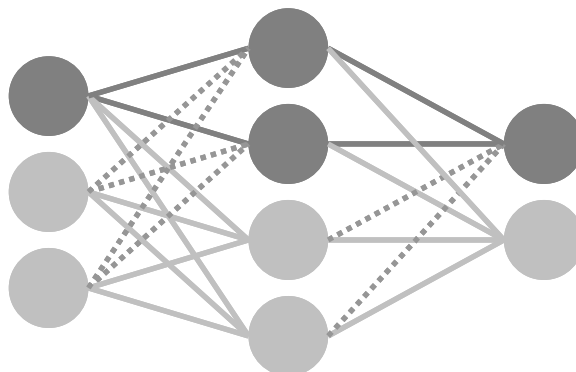
Incremental Neuraal Netwerk

Het Incremental Neuraal Netwerk begint, zoals uitvoerig zal worden beschreven in hoofdstuk 3, klein in wereld 1 en wordt in elke opvolgende wereld groter doordat er neuronen aan het netwerk worden toegevoegd tot het in wereld 4 dezelfde architectuur heeft als het Sequential Neuraal Netwerk. Wanneer een neuron wordt toegevoegd, worden ook de benodigde verbindingen met gewichten toegevoegd zodat, zoals boven uitgelegd bij het Feedforward Neuraal Netwerk, elke neuron in een laag is verbonden met elke neuron in de opvolgende laag.

Het toevoegen van input neuronen gebeurt per informatie-element. Zoals eerder beschreven verzamelen zintuigen vijf informatie-elementen (zie hoofdstuk 2.3). Als een informatie-element wordt toegevoegd aan het netwerk, betekent dit dat *voor elk zintuig* een neuron voor dit element wordt toegevoegd. Dus als bijvoorbeeld het element 'hoeveelheid dat een object bevat' wordt toegevoegd, wordt er voor elk van de vijf zintuigen een neuron toegevoegd, ofwel vijf neuronen.

Voordat het Incremental Neuraal Netwerk wordt uitgebreid met nieuwe neuronen (lichtgrijze neuronen in figuur 6), krijgen de bestaande neuronen (donkergrijs), gewichten en bias waarden een zeer beperkte kans om te muteren (donkergrijs). Aan deze 'vastgezette' neuronen worden de nieuwe neuronen met hun verbindingen, gewichten en bias waarden toegevoegd. De verbindingen van nieuwe neuronen *naar* vastgezette neuronen krijgen een middelmatige kans om te muteren (gestreept grijs). De verbindingen van vastgezette

neuronen *naar* nieuwe neuronen en van nieuwe neuronen *naar* nieuwe neuronen krijgen de standaard mutatie kans (lichtgrijs). Deze kansen worden gebruikt door de mutatie operator en zijn beschreven in hoofdstuk 2.5.



Figuur 6 Schematische weergave van kansen van een INN.

De reden voor het toekennen van verschillende kansen is als volgt. In wereld 2, 3 en 4 worden er neuronennetwerken aan het netwerk toegevoegd (zie hoofdstuk 3). De beginpopulatie in wereld 2 bijvoorbeeld bestaat uit de netwerken van de eindpopulatie van wereld 1, uitgebreid met een aantal nieuwe neuronennetwerken. Dus het vastgezette deel van het netwerk heeft, als creature, succesvol het leerdoel van een eerdere wereld behaald. Dit gedrag, bepaald door de waarden van het netwerk, moet niet verloren gaan. Dit gedrag is namelijk ook nodig voor het behalen van het leerdoel in de nieuwe wereld. Om dit gedrag te behouden wordt het 'oude' netwerk vrijwel vastgezet; doordat het nauwelijks kan muteren zal het niet (snel) verloren gaan.

Om nieuw gedrag te leren worden de nieuwe neuronennetwerken aan het netwerk toegevoegd. Maar deze nieuwe neuronennetwerken moeten niet direct het oude gedrag sterk beïnvloeden. Daarom moeten de gewichten van de verbindingen van nieuwe neuronennetwerken *naar* oude neuronennetwerken niet te sterk muteren. De mutatie kans van die verbindingen is dan ook middelmatig.

De nieuw toegevoegde neuronennetwerken zijn bedoeld om nieuw gedrag te leren. Hiervoor moeten ze toegang hebben tot alle mogelijke informatie (zowel van vastgezet naar nieuw als van nieuw naar nieuw). Deze neuronennetwerken en verbindingen krijgen dus de standaard mutatie kans.

Door het toevoegen van input neuronennetwerken krijgt het netwerk meer informatie over zijn omgeving; het kan meer dingen observeren. Het toevoegen van hidden neuronennetwerken geeft het netwerk de ruimte om snel nieuw gedrag te leren of oud gedrag aan te passen. Het toevoegen van output neuronennetwerken geeft het netwerk de mogelijkheid om nieuwe acties uit te voeren.

2.5 Ontwikkeling van leermethoden

2.5.1 Concept

Er zijn verschillende leermethoden om een neurale netwerk te trainen voor een bepaalde taak. De keuze hangt af van het type neurale netwerk, de taak die het netwerk moet uitvoeren en de voorkeur van de onderzoeker.

Aangezien dit onderzoek zich richt op de evaluatie van de *evolutie* van creatures in een dynamische omgeving, is ervoor gekozen om het leerproces van de neurale netwerken te laten verlopen door middel van een evolutionaire leeralgoritme. Maar moet dit evolutionair leeralgoritme 'supervised' of 'unsupervised' verlopen (zie hoofdstuk 1.2)?

Unsupervised leren lijkt op een probleem te stuiten. Evolutionaire algoritmen kunnen alleen unsupervised leren laten optreden als een populatie van neurale netwerken zich in een omgeving bevindt en zichzelf bovendien in stand houdt. Wanneer neurale netwerken opereren in een bepaalde omgeving waarin ze onder bepaalde omstandigheden verdwijnen of samen met een ander netwerk een nieuw netwerk maken, treedt er namelijk natuurlijke selectie op. Zolang een populatie zichzelf niet in stand kan houden, kan dit echter niet optreden.

Ook supervised leren lijkt in eerste instantie problematisch voor dit onderzoek. Dit heeft te maken met het feit dat neurale netwerken worden onderzocht die als hersenen van creatures in een dynamische omgeving opereren, waardoor er geen input-output combinaties beschikbaar zijn. De gewenste outputs bij bepaalde inputs zijn namelijk niet te bepalen van een creature dat zich een aanzienlijk aantal iteraties in een omgeving bevindt en tijdens zijn leven tal van acties onderneemt. Het is namelijk niet direct duidelijk of een bepaalde actie op een bepaald tijdstip de gewenste actie is: moet een creature bijvoorbeeld naar rechts lopen, naar links lopen of iets eten?

Maar er is een oplossing. Zolang een populatie van netwerken zichzelf nog niet in stand kan houden, kan een zogenaamd 'steady state' evolutionair leeralgoritme [Yao 1] worden toegepast. Bij een steady state genetisch algoritme treedt er geen natuurlijke selectie op, maar wordt dit bepaald door de onderzoeker. Dit wordt gedaan door een lijst bij te houden van neurale netwerken, die elk een score krijgen toegewezen op basis van het gedrag dat zij vertonen als creature in een omgeving. Een hoge score geeft aan dat het gedrag gewenst was (door de onderzoeker), een slechte score dat het gedrag ongewenst was. Deze score beïnvloedt direct hun kans tot reproduceren: een hogere score geeft een hogere kans op kinderen krijgen, die slecht presterende netwerken in de populatie vervangen. Ofwel, een

steady state evolutionair leeralgoritme stuurt natuurlijke selectie in de richting die gewenst is door de onderzoeker; het is dus een supervised leeralgoritme.

Daarom wordt in dit onderzoek eerst een steady state algoritme gebruikt om een populatie te leren in zichzelf in stand te houden (wereld 1 t/m 4, zie hoofdstuk 3). Hierna wordt het steady state algoritme uitgeschakeld, waarna het leerproces alleen nog onderhevig is aan natuurlijke selectie (wereld 5, zie hoofdstuk 3). Ofwel, eerst verloopt het leerproces via een supervised evolutionair algoritme en daarna via een unsupervised leeralgoritme.

Het evolutionair algoritme, zowel supervised als unsupervised, heeft genetische operatoren nodig waar creatures onderhevig aan zijn wanneer ze zich voortplanten. De standaard operatoren zijn mutatie en crossover [Montana]. Via crossover wordt een nieuw netwerk gemaakt uit twee andere netwerken (zoals twee creatures die een kind krijgen). Door middel van mutatie veranderen netwerken op willekeurig plekken een klein beetje. Mutatie zorgt ervoor dat netwerken altijd spontaan nieuwe dingen kunnen leren; wanneer een neuraal netwerk verandert, verandert het gedrag van een creature ook. Deze verandering kan leiden tot een grotere kans om zich te reproduceren, maar ook tot een kleinere kans.

In [Montana] worden verscheidene variaties op deze standaard genetische operatoren vergeleken. Uit dat onderzoek blijkt dat de variaties die op 'nodes' opereren de beste resultaten geven. Daarom zijn deze operatoren in dit onderzoek toegepast, en worden hieronder in detail beschreven.

2.5.2 Beschrijving

Leeralgoritme

Zolang een populatie van creatures zichzelf niet in stand kan houden (zoals in wereld 1 t/m 4, zie hoofdstuk 3), bepaalt een 'steady state' genetisch algoritme uit welke creatures de populatie bestaat. Dit steady state algoritme houdt een lijst bij van de genen van 100 creatures. Dit wordt de *populatie* van creatures genoemd. De beginpopulatie is *generatie 0*. In *generatie 1* zal de populatie deels zijn veranderd door het genetisch algoritme, etc. De eindpopulatie is de laatste generatie, *generatie n*. Ofwel elke iteratie van het genetisch algoritme verandert de samenstelling van de populatie en neemt de generatie met één toe. Het steady state genetisch algoritme werkt als volgt:

Voor wereld 1 bestaat de beginpopulatie uit willekeurig gegenereerde genen (met bepaalde beperkingen, zie hoofdstuk 2.4). Voor elke opvolgende wereld bestaat de beginpopulatie uit de eindpopulatie van de vorige wereld, dat wil zeggen uit de populatie die het doel van de vorige wereld heeft behaald. Op elke generatie worden de volgende stappen uitgevoerd totdat het leerdoel van een wereld is behaald:

1. De fitness van alle genen in de populatie wordt bepaald door ze één voor één in de wereld te plaatsen als creature en ze na hun dood een score te geven. De bepaling van deze score hangt af van de wereld (zie hoofdstuk 3). Een gen is succesvol als het, als een creature, het doel van de wereld heeft behaald.
2. Als voldoende genen succesvol zijn, stopt de simulatie; de populatie heeft dan het leerdoel van de wereld behaald. De populatie van die generatie is dan de eindpopulatie.
3. De genen worden gerangschikt op basis van hun fitness (hogere fitness is beter) en krijgen een kans toegewezen gebaseerd op deze rangschikking. Er is een sterke voorkeur voor de genen met een hogere fitness; ze krijgen een veel hogere kans.
4. Met behulp van deze kansen wordt er een nieuwe populatie gecreëerd die de huidige populatie zal vervangen:
 - a. De beste 50 genen van deze populatie maken ook deel uit van de nieuwe populatie.
 - b. Er worden 25 nieuwe genen gegenereerd. Dit wordt gedaan door probabilistisch (ofwel, gebaseerd op de toegewezen kansen) twee genen te kiezen uit de populatie, waarmee een nieuw gen wordt gecreëerd door hierop bepaalde genetische operatoren toe te passen (deze operatoren zijn later in dit hoofdstuk beschreven).
 - c. Er worden 25 genen willekeurig (met bepaalde beperkingen, zie hoofdstuk 2.4) gegenereerd.
5. Deze nieuwe populatie doorloopt al deze vijf stappen opnieuw. Bij stap 1 echter hebben de 50 beste genen 50% kans om hun fitness te behouden zonder opnieuw in de wereld te worden geplaatst. Als ze hun fitness behouden krijgen ze opnieuw kans om hun genen door te geven aan de volgende generatie, maar zonder dat ze zich opnieuw hoeven te bewijzen. Als ze wel opnieuw in de wereld worden geplaatst moeten ze zich dus wel opnieuw bewijzen en laten zien dat ze geen one-hit-wonder waren. Met deze methode wordt zowel het doorgeven van goede genen als het verwijderen van slechte genen gestimuleerd.

Aangezien de 25 nieuwe genen (4b) en de 25 willekeurige genen (4c) altijd kans hebben om een volledige mislukking te zijn (omdat mutatie en crossover willekeurig zijn), worden alleen de 50 beste genen (4a) in beschouwing genomen om te bepalen of de populatie het leerdoel van een bepaalde wereld heeft behaald. Het leerdoel van een wereld moet twee generaties achter elkaar worden behaald. De grens tussen wel of niet het leerdoel behaald te hebben is in het onderzoek gesteld op 90%, ofwel 45 van de 50 beste genen moeten het gestelde doel van een wereld hebben behaald.

Bij het Incremental Neuraal Netwerk worden bij 4c de genen niet compleet willekeurig gegenereerd. Alleen de delen van het netwerk die nieuw worden toegevoegd in een wereld zijn willekeurig. De andere delen worden willekeurig gekozen uit de 50 beste genen. Als dit niet op deze manier zou worden gedaan, dan zou de populatie zeer snel de geleerde delen van het netwerk kunnen verliezen. (Bijvoorbeeld als een gen tijdens een bepaalde generatie een slechte fitness krijgt waardoor het niet in de beste 50 komt en het dus verloren gaat).

Genetische Operatoren

Wanneer een nieuw gen wordt gecreëerd uit twee andere genen, wordt dit gedaan door twee genetische operatoren, crossover en mutatie, die beschreven staan in [Montana]. Deze operatoren opereren op de 'nodes' van de neurale netwerken in het gen. Een node in een neuraal netwerk is een neuron, zijn bias, de verbindingen die input leveren aan dit neuron en de gewichten die bij deze verbindingen behoren. Geen enkele input neuron is een node, omdat ze zelf geen input verbindingen en geen bias hebben.

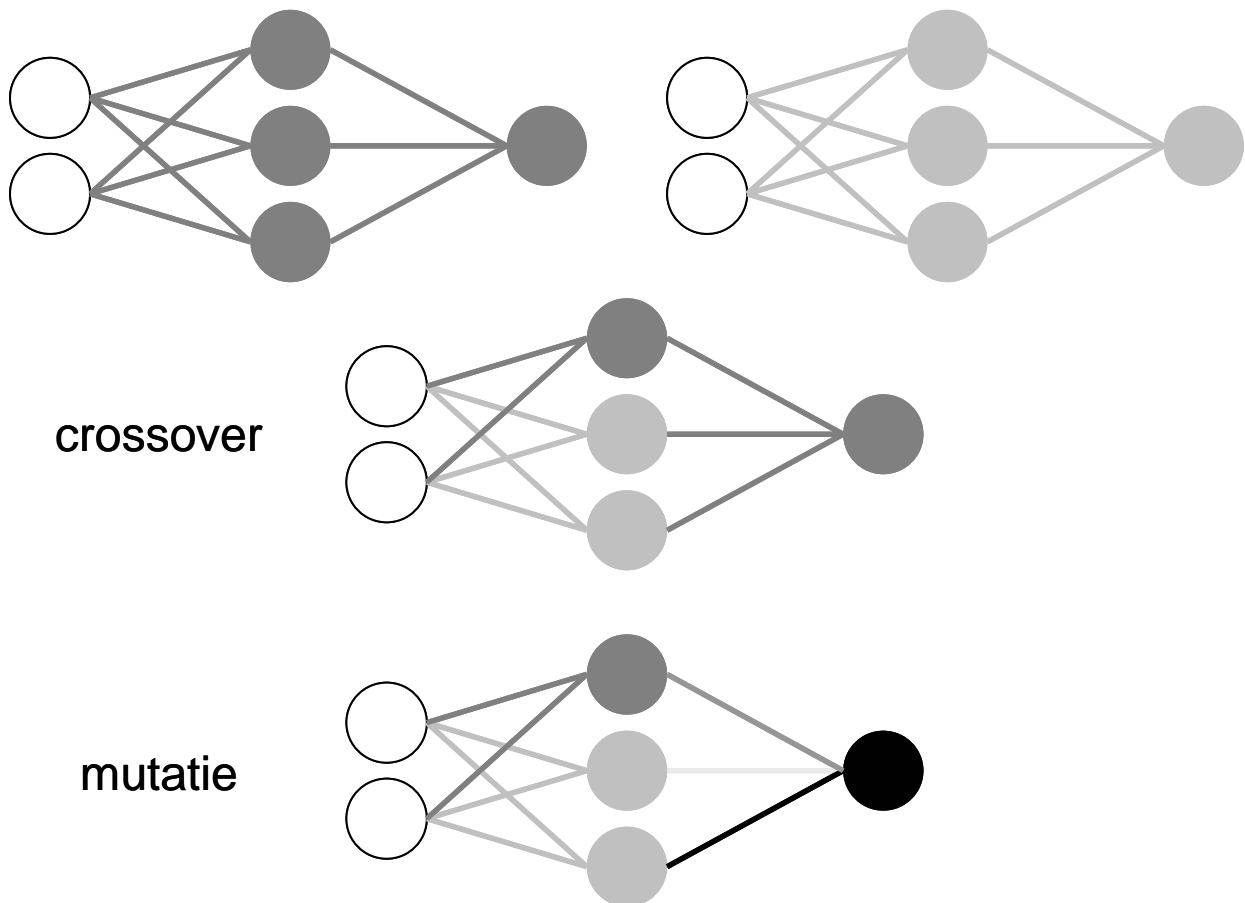
De genetische operatoren worden in sequentie toegepast, d.w.z. eerst crossover en dan mutatie. In figuur 7 zijn ze schematisch weergegeven.

De eerste operator is 'Crossover Nodes'. Deze operator bouwt een nieuw netwerk op uit delen van twee andere netwerken. Elke node in het nieuwe netwerk is van het ene of van het andere netwerk, ofwel 50% kans.

Op dit nieuwe netwerk wordt dan de tweede genetische operator toegepast, namelijk 'Muteer Nodes'. Deze operator verandert een netwerk een beetje. Elke node heeft een bepaalde kans om te worden veranderd. Deze kans verschilt per type netwerk (zie hoofdstuk 2.4). De standaard kans is 0.1, die overeenkomt met de gebruikte kans in [Montana]. Voor het Sequential Neuraal Netwerk is het dus de standaard kans van 0.1. Dit betekent dat elke node in het netwerk een kans van 10% heeft om te muteren. Als een node muteert, dan worden de bias van het neuron en de gewichten van zijn inkomende verbindingen een beetje veranderd.

Het Incremental Neuraal Netwerk heeft verschillende mutatiekansen voor verschillende delen van het netwerk. Een nieuwe node heeft een kans van 0.1; het heeft 10% kans om te muteren en dan worden zijn bias waarde en al de gewichten van zijn inkomende verbindingen een beetje veranderd. Een vastgezette node heeft een kans van 0.01 om te muteren; dit betekent dat het 1% kans heeft dat zijn bias en al de gewichten van inkomende verbindingen van *nieuwe* neuronen een beetje veranderen. De gewichten van zijn inkomende verbindingen van *vastgezette* neuronen veranderen pas als, na de initiële 0.01 kans, nogmaals de 0.01 kans wordt behaald. De veranderingen van de waarden (gewichten en bias) bij mutatie zijn normaal verdeeld met $\mu = 0$ en $\sigma = 0.2$ (zie hoofdstuk 1.2).

Na crossover en mutatie bevat het resulterende neurale netwerk eigenschappen van beide andere netwerken en zal dus waarschijnlijk licht verschillen van de oorspronkelijke netwerken.



Figuur 7 Schematische weergave van de genetische operatoren. De kleur van een neuron stelt zijn bias waarde voor en de kleur van een verbinding stelt zijn gewicht voor. Crossover: De eerste hidden node en de output node zijn van het linker netwerk, de tweede en derde hidden nodes zijn van het rechter netwerk. Mutatie: De output node is gemuteerd; zijn bias en gewichten zijn veranderd.

In het begin van dit onderzoek stond niet alleen informatie over het neurale netwerk in de genen, maar ook andere informatie, waarvoor ook genetische operatoren waren ontwikkeld. In het uiteindelijke onderzoek is hiervan geen gebruik gemaakt. Voor meer detail zie hoofdstuk 2.7.

2.6 Ontwikkeling dataverwerking en analyse methodologie

2.6.1 Concept

Zoals beschreven in hoofdstuk 3, bestaat het onderzoek uit meerdere fasen die meerdere keren worden doorlopen door de verschillende typen netwerken. Het doorlopen van een fase kan, vooral bij de werelden 3 en 4, verscheidene uren kosten. Als er een probleem ontstaat en het computerprogramma crasht of tijdelijk moet worden uitgezet, is het van belang dat een fase daarna kan worden voortgezet zonder geheel opnieuw te moeten beginnen.

Daarom is een periodieke opslag van belangrijke informatie nodig, die kan worden ingeladen door het programma als het doorlopen van de fase wordt voortgezet. Naast deze fail-safe procedure tijdens het doorlopen van een fase, moet er ook informatie worden opgeslagen over de resultaten van elke fase.

Omdat het onderzoek gericht is op het vergelijken van de leersnelheid van verschillende typen netwerken in verscheidene fasen (zie hoofdstuk 1.3), ligt bij de analyse hier natuurlijk ook de nadruk op. Om de verschillende netwerken goed te kunnen vergelijken worden alle fasen meerdere keren doorlopen (zie hoofdstuk 3).

2.6.2 Beschrijving

De voortgang van het leerproces wordt periodiek opgeslagen in bestanden, namelijk elk half uur. De informatie die wordt opgeslagen bestaat uit de genen van de complete populatie creatures van de huidige generatie. Deze genen, zoals beschreven in hoofdstuk 2.3 bestaan uit lijsten waarden die het neurale netwerk specificeren. Elk gen wordt opgeslagen als een apart bestand. Als er een probleem optreedt, kan de doorloop van een fase dus worden voortgezet door de bestanden met de laatst opgeslagen generatie genen in het programma te laden.

Daarnaast wordt er nog veel andere informatie opgeslagen, zoals het iteratie nummer van dat moment, het aantal succesvolle en onsuccesvolle acties dat is uitgevoerd door de creatures, de verhoudingen tussen het aantal verschillende acties, de hoeveelheid voedsel dat is gegeten, de gemiddeld behaalde leeftijd van de creatures, e.d.

Tijdens de doorloop van een fase wordt bovendien van elke generatie de volgende informatie opgeslagen:

- De fitness die alle genen tijdens hun leven als creature hebben behaald.
- Het aantal creatures dat het doel van de wereld heeft behaald.

- Het tijdstip waarop deze informatie is opgeslagen.

De opgeslagen informatie is gebruikt voor het testen en aanpassen (zie hoofdstuk 2.7) van de onderdelen van de onderzoeksopzet en voor de analyse van de eindresultaten.

Om de onderzoeksvraag te kunnen beantwoorden, richt de analyse van de resultaten zich op (a) het aantal creatures dat het doel van een wereld heeft bereikt per generatie en (b) het totale aantal generaties dat het duurde voordat het leerdoel van een wereld werd bereikt. Het eerste is gebruikt om het doorlopen van een fase door de verschillende typen netwerken met elkaar te vergelijken. Dit wordt geïllustreerd door de lijngrafieken in hoofdstuk 4. Omdat elke fase meerdere keren is doorlopen door elke netwerk, is het tweede gebruikt om het gemiddelde aantal generaties te bepalen dat nodig was om het leerdoel van een wereld te behalen. Dit is aangegeven in de staafgrafieken in hoofdstuk 4, waarbij ook de standaard fout (zie tabel 3.2) wordt weergegeven door middel van error bars.

De hypothese, zoals aangegeven in hoofdstuk 1.3, is dat het INN sneller het leerdoel bereikt dan het SNN, ofwel het aantal generaties dat nodig is om het leerdoel te bereiken is bij het INN lager dan bij het SNN. Deze hypothese is getest door het toepassen van een eenzijdige onafhankelijke t-toets met een overschrijdingskans, of p-waarde, van 1%. Ofwel, als de verkregen p-waarde lager is dan 0.01 dan is het INN significant sneller dan het SNN. Ter illustratie is hieronder de berekenwijze van de p-waarde voor wereld 1 weergegeven.

In tabel 3.1 staan de verkregen resultaten van wereld 1 voor de twee typen netwerken. Hieruit worden de waarden in tabel 3.2 berekend. De verkregen waarden worden vervolgens gebruikt om de waarden in tabel 3.3 te berekenen, waarmee dan de p-waarde kan worden berekend met behulp van een p-waarden tabel. In tabel 3.4 is het relevante deel van een dergelijke p-waarden tabel weergegeven.

| | x_1 | x_2 | x_3 |
|------------|-------|-------|-------|
| INN | 37 | 43 | 57 |
| SNN | 151 | 124 | 99 |

Tabel 3.1 Aantal generaties per steekproefelement

| | n | vg | μ | σ | sf |
|------------|----------|-----------|---------|----------|-----------|
| INN | 3 | 2 | 45.667 | 10.263 | 5.925 |
| SNN | 3 | 2 | 124.667 | 26.006 | 15.015 |

Tabel 3.2 Aantal steekproefelementen, aantal vrijheidsgraden, steekproefgemiddelde, standaarddeviatie en standaardfout

| | |
|-------------|---------|
| vgt | 4 |
| gvar | 390.833 |
| t | 4.894 |

Tabel 3.3 Totale aantal vrijheidsgraden, gepoolde variantie en t-waarde

| vgt 4 | t waarde bij gegeven p waarde | | | | |
|--------------|--------------------------------------|-------|-------|-------|--------|
| t | 3.00 | 3.30 | 3.75 | 4.60 | 8.61 |
| p | 0.020 | 0.015 | 0.010 | 0.005 | 0.0005 |

Tabel 3.4 Deel van een p-waarden tabel

$x_1 \dots x_n$: steekproefelementen.

n: aantal steekproefelementen

vg: aantal vrijheidsgraden, $\mathbf{vg} = \mathbf{n} - 1$

μ : steekproefgemiddelde, $\mu = (x_1 + \dots + x_n) / n$

σ : standaardafwijking, $\sigma = \sqrt{(1 / (n-1)) * \sum(x_i - \mu)^2}$

sf: standaardfout, $\mathbf{sf} = \sigma / \sqrt{n}$

vgt: totale aantal vrijheidsgraden, $\mathbf{vgt} = \mathbf{vg}_1 + \mathbf{vg}_2$

gvar: gepoolde variantie, $\mathbf{gvar} = (\mathbf{vg}_1 * \sigma_1^2 + \mathbf{vg}_2 * \sigma_2^2) / \mathbf{vgt}$

t: t-waarde, $\mathbf{t} = (\mu_1 - \mu_2) / (\sqrt{\mathbf{gvar} * (1/n_1 + 1/n_2)})$

p: p-waarde, berekenen met behulp van een p-waarden tabel (deels weergegeven) aan de hand van vgt en t:

$p(4.894)$ valt tussen $p(4.60) = 0.005$ en $p(8.61) = 0.0005$.

$p(4.894) = (8.61 - 4.894) / (8.61 - 4.60) * (0.005 - 0.0005) = 0.004$

De p-waarde is 0.004, wat lager is dan de overschrijdingskans van 0.01. Dit betekent dat het INN significant sneller is dan het SNN in wereld 1.

2.7 Testen en aanpassen

2.7.1 Inleiding

Het leerproces van neurale netwerken is in feite een zoektocht naar de ideale oplossing voor een bepaald probleem, waarbij de oplossing bestaat uit het resulterende netwerk. De zoektocht naar een oplossing gebeurt binnen een bepaalde zoekruimte. De grootte van de zoekruimte wordt voornamelijk bepaald door het aantal verschillende staten waarin de omgeving zich kan bevinden en het aantal verschillende staten waarin het neurale netwerk zich kan bevinden. De grootte van deze zoekruimte heeft grote invloed op de benodigde leertijd van neurale netwerken: hoe groter de zoekruimte is, des te meer verschillende staten een neuraal netwerk moet testen.

In dit onderzoek is de zoekruimte aanvankelijk relatief groot genomen waarna deze, door middel van testen en aanpassen, werd verkleind tot een voor het doel van dit onderzoek geschikte zoekruimte. De resulterende zoekruimte is beschreven in hoofdstuk 2.2 t/m 2.5. De originele grote zoekruimte wordt in dit hoofdstuk summier beschreven.

Zoals beschreven in hoofdstuk 2.6 werd elk half uur bepaalde informatie opgeslagen in bestanden. Om de totale onderzoekstijd binnen redelijke grenzen te houden, was er een limiet gesteld aan de tijd dat een leerproces mocht duren. Het leerproces werd gelimiteerd tot 10 uur, waarna de verkregen informatie werd geanalyseerd om de onderzoeksofzet te testen en (eventueel) aan te passen.

De informatie over het aantal ondernomen acties, de verhoudingen tussen deze acties, het succes of het falen van de acties, de gemiddelde leeftijden, etc. werden geanalyseerd op opmerkelijke of onverwachte waarden, waarmee bugs konden worden opgespoord of leeralgoritmen bijgesteld.

Het testen en aanpassen van de verschillende onderdelen van dit onderzoek richtte zich voornamelijk op twee zaken:

- Het aanpassen van de zoekruimte.
- Het identificeren en rectificeren van bugs/programmeerfouten.

In het volgende worden de verschillende onderdelen van de uitgebreide zoekruimte zoals die in het begin van het onderzoek bestond summier beschreven, met een verwijzing naar de definitieve onderzoeksofzet, zoals in voorgaande paragrafen is weergegeven.

2.7.2 De startsituatie

Dynamische omgeving

De dynamische omgeving bestond in het begin niet uit één maar uit twee lagen, een vloer met een laag daarboven. Creatures bevonden zich alleen in de bovenlaag. De vloer bestond voornamelijk uit wall blokjes, maar hierin konden zich ook andere objecten bevinden.

De dynamische omgeving bevatte naast de in hoofdstuk 2.2 beschreven objecten (wall en food blokje) ook:

- Water blokjes
- Lava blokjes
- Bewegende wall blokjes

Water blokjes vormen samen met food blokjes een ecosysteem dat zichzelf in stand houdt.

De hoeveelheid water per water blokje neemt elke iteratie een beetje toe (vergelijk met regen in de natuur). Wanneer een blokje een bepaalde hoeveelheid water bevat, 'loopt het over' en komt er een nieuw water blokje in de buurt te staan. Food blokjes nemen constant in hoeveelheid voedsel af (vergelijk met uitdrogen), behalve als ze in de buurt van een water blokje staan. Dan verbruiken ze water van het water blokje en nemen zelf in hoeveelheid voedsel toe. Als een food blokje een bepaalde hoeveelheid voedsel bevat, dan 'plant het zich voort' en komt er een nieuw food blokje in de buurt te staan.

Lava blokjes stellen natuurlijke gevaren voor. Ze beschadigen andere objecten wanneer deze in de buurt staan: water blokjes verdampen, food blokjes verbranden en creatures raken gewond (verliezen HP).

Naast vaste wall blokjes zijn er ook bewegende wall blokjes die willekeurig of in een bepaald patroon bewegen en zo dynamische obstakels kunnen vormen.

Water en lava blokjes kunnen zich zowel in als op de vloer van de wereld bevinden, terwijl creatures, food blokjes en bewegende wall blokjes alleen op de vloer voorkomen.

Creatures

In het begin van het onderzoek beschikten creatures, *naast of in plaats van* de in hoofdstuk 2.3 beschreven eigenschappen, over de volgende eigenschappen:

a. Lichaam

- Voorkant/Achterkant
 - Een creature heeft *geen* voor- of achterkant. Acties en zintuigen zijn niet relatief aan de oriëntatie van een creature zoals in definitief ontwerp, maar anders opgebouwd (zie (b) en (c)).

- Of de creature op dit veld staat, ernaartoe beweegt of ervan weg beweegt
 - Zoals eerder is uitgelegd in hoofdstuk 2.2 bestaat de omgeving uit een schaakbord met velden waarop objecten/creatures kunnen staan. Deze interne eigenschap geeft aan of de creature stil staat op een veld, zich weg beweegt van dit veld of naar dit veld toe beweegt. In combinatie met de oriëntatie eigenschap bepaalt dit dus naar welk veld de creature beweegt.
- HP
 - HP staat voor 'Hit Points', een veelgebruikte term in videospellen. Het geeft aan hoeveel schade iemand kan oplopen voordat hij sterft.
 - HP heeft, net als energie, een huidige en een maximale hoeveelheid.
 - HP neemt af wanneer een creature wordt beschadigd. Dit kan als het wordt aangevallen door een andere creature ('attack' actie), of wordt beschadigd door een lava blokje.
 - HP neemt (zo nodig) automatisch elke iteratie een beetje toe waarbij energie wordt verbruikt.
 - Als HP nul is, sterft de creature. Op het veld waar de creature stond, komt dan een food blokje te staan met een hoeveelheid voedsel evenredig aan de hoeveelheid energie van de creature.
- Houding
 - Wanneer een creature een actie onderneemt verandert zijn houding, ook als de actie faalt. Zo kunnen creatures van elkaar, en van zichzelf, zien wat de laatste actie is die zij hebben ondernomen.
 - De verschillende mogelijke houdingen komen overeen met de verschillende acties, waarbij de 4 bewegings acties door dezelfde houding 'move' worden weergegeven.
- Leefkosten
 - De leefkosten zijn tussen 1 en 2 energie eenheden per iteratie.
 - De leefkosten zijn afhankelijk van de grootte van het brein, ofwel het neurale netwerk, van de creature. Deze grootte wordt bepaald door het aantal zintuigen en hidden neuronen die actief zijn (beschreven in (b) en (d)).
- Talk
 - Dit is het resultaat van de actie 'talk'. (zie (c)).
 - Talk heeft vier verschillende standen, die andere creatures kunnen waarnemen.
 - De standen hebben geen inherente betekenis. Door evolutie zouden creatures deze standen betekenis kunnen geven. (Bijvoorbeeld dat stand 1 betekent dat de creature een andere creature zoekt om mee te 'maten').

- Kleur
 - Dit wordt bepaald door de genen van een creature. De genen worden gecombineerd in een enkele waarde.
 - Deze gecombineerde waarde zou kunnen worden gebruikt om te zien hoe zeer de genen van een creature verschillen van een andere creature.
- Maximum energieniveau
 - Dit bepaalt de snelheid van de acties van een creature.
 - Hoe meer energie een creature kan bezitten (hoe groter een creature is), hoe langzamer het is (hoe meer iteraties een actie duurt).

b. Zintuigen

In de definitieve opzet kijken zintuigen vanaf een creature in een rechte lijn in de verte. In de oorspronkelijke opzet kunnen ze elk slechts een enkel veld waarnemen. Elke creature beschikt over 10 zulke zintuigen, die elk op een bepaalde plek gericht zijn. Deze plekken zijn niet relatief aan de oriëntatie van een creature. Een zintuig observeert bijvoorbeeld altijd het veld twee noordelijk en één westelijk van een creature. Aangezien de omgeving uit twee lagen bestaat (er is een vloer laag en een boven laag), kunnen zintuigen ook op een vloer veld gericht zijn (zoals één veld noordelijk, één oostelijk en één *omlaag*). De plekken waar zintuigen op zijn gericht veranderen onder invloed van crossover en mutatie (zie verder in dit hoofdstuk).

Voor elk zintuig worden de volgende informatie-elementen verzameld:

- Welk object op het veld staat (leeg, wall, food, creature, water, lava).
- Hoeveelheid (energie, voedsel, water, lava).
- Hoeveelheid HP
- Houding (do nothing, move, eat, mate, attack, give energy, talk, change color)
- Oriëntatie (noord, oost, zuid, west).
- Talk (vier waarden)
- Kleur
- Hoe lang geleden het object een actie heeft ondernomen.
- Of het object op het geobserveerde veld staat, er naar toe beweegt of er van weg beweegt

De input waarden van het netwerk waren in het begin van het onderzoek binair (ja/nee). Daarom werden verschillende informatie-elementen (zoals hoeveelheid) opgesplitst in drie

nauwkeurighedsniveaus (hoog/middel/laag). Er zijn dan in totaal $24 + 6 \times 3 = 42$ informatie-elementen *per zintuig*, die elk een input neuron vereisen. Dit resulteert in een totaal van $10 \times 42 = 420$ input neuronen. In het uiteindelijke onderzoek worden slechts 5 informatie elementen gebruikt per zintuig, resulterend in slechts $5 \times 5 = 25$ input neuronen.

c. Acties

- Do Nothing
 - Door deze actie te kiezen doet een creature 1 iteratie lang niks.
 - Dit kost geen energie. De leefkosten voor deze iteratie moeten natuurlijk wel worden betaald.
- Attack
 - Deze actie wordt gebruikt om een andere creature aan te vallen.
 - De HP van de andere creature neemt af.
 - Wanneer de HP van de andere creature nul is, sterft het en zijn resterende energie hoeveelheid vormt dan een food blokje.
 - Deze actie zou kunnen resulteren in prooi en predator creature soorten.
- Give Energy
 - Deze actie wordt gebruikt om energie te schenken aan een andere creature.
 - De energie hoeveelheid van de andere creature neemt toe.
- Talk
 - De creature kiest een van de vier mogelijke waarden van talk.
 - Dit kost een bepaalde kleine hoeveelheid energie.
- Change Color
 - De kleur van een creature, bepaald door zijn genen, kan hiermee een beetje worden veranderd.
 - Dit kost een bepaalde hoeveelheid energie.
 - Hiermee kan een creature zich anders voordoen dan het eigenlijk is. Als er verschillende soorten creatures bestaan met distinctieve kleur waarden, kunnen creatures zich hiermee voordoen als een andere soort.

Acties worden niet uitgevoerd relatief tot de oriëntatie van een creature, zoals in het definitieve ontwerp, maar door een bepaalde actie *én* een bepaalde richting te kiezen. Er bestaan dus ook geen vier verschillende bewegingsacties, maar slechts een enkele 'move' actie.

De kracht van alle acties wordt beïnvloed door bepaalde waarden in de genen (zie (d)). Deze waarden geven aan hoeveel extra energie de creature in een actie stopt, waardoor de actie op een bepaalde manier wordt versterkt. Bij de 'move' actie resulteert deze extra energie in

een snellere beweging. Bij de 'eat' actie wordt er extra energie verbruikt om ook extra energie te verkrijgen. Bij de 'mate' actie bepaalt de extra energie hoeveel energie er aan de nakomeling wordt geschonken. Bij de 'attack' actie resulteert de extra energie in extra schade. Bij de 'give energy' actie bepaalt de extra energie hoeveel energie geschonken wordt.

Het totaal aantal mogelijke acties is acht (move/eat/mate/attack/give energy/talk/change color/do nothing), ofwel acht output neuronen. Om een actie uit te voeren moet ook een richting worden gekozen (noord/oost/zuid/west), ofwel nog vier output neuronen. Dit resulteert in een totaal van $8 + 4 = 12$ output neuronen. In de uiteindelijke onderzoeksopzet worden slechts zes acties gebruikt, resulterend in slechts zes output neuronen.

d. Genen

- Neuraal netwerk
- Zintuig posities.
- Welke zintuigen aan of uit staan.
- Welke hidden neuronen aan of uit staan.
- Maximum energie niveau.
- Maximum leeftijd.
- De extra energie voor de actie 'move'.
- De extra energie voor de actie 'eat'.
- De extra energie voor de actie 'mate'.
- De extra energie voor de actie 'attack'.
- De extra energie voor de actie 'give energy'.

Al de genen kunnen, onder invloed van crossover en mutatie, veranderen. De zintuig posities geven aan welke velden rond de creature worden waargenomen (zie (b)). Het uit hebben staan van zintuigen en hidden neuronen maakt het neurale netwerk kleiner, waardoor de leefkosten van een creature lager zijn. Het maximum energie niveau bepaalt de snelheid van acties. De extra energieën bepalen de kracht van acties. Door middel van de maximum leeftijd kan een soort creatures evolueren tot bijvoorbeeld het hebben van een korte levensspan, waardoor ze zich sneller via natuurlijke selectie aanpassen aan een verandering in de omgeving.

Neurale Netwerken

De gebruikte neurale netwerken waren gebaseerd op Feedforward Neurale Netwerken. Zoals hierboven aangegeven, beschikten creatures over 42 informatie elementen per zintuig,

met 10 zintuigen, tot een maximum van $10 \times 42 = 420$ input neuronen en over 8 acties met 4 richtingen resulterend in 12 output neuronen. Het aantal hidden neuronen was 100.

Zoals hierboven bij (d) genen echter is aangegeven staan niet alle zintuigen en hidden neuronen altijd aan. Bij het genereren van een neuraal netwerk was de kans dat een zintuig of hidden neuron actief was 50%. Ofwel, gemiddeld bestond een netwerk uit 210 input neuronen (5 zintuigen \times 42 informatie elementen), 50 hidden neuronen en 12 output neuronen. Een netwerk bestaat dan uit $(210 \times 50) + (50 \times 12) = 11.100$ verbindingen met gewichten en $50 + 12 = 62$ bias waarden, voor een totaal van $11.100 + 62 = 11.162$ waarden. Een maximaal (alles aan) netwerk, zoals dat in het begin van het onderzoek was ontworpen, bestaat uit $(420 \times 100) + (100 \times 12) + 100 + 12 = 43.312$ waarden. Het uiteindelijk in dit onderzoek gebruikte netwerk bestaat uit 806 waarden (slechts 1.86% van het maximale netwerk en 7.22% van het gemiddelde netwerk in het begin van het onderzoek).

De grootte van het netwerk bepaalde in het begin van het onderzoek, zoals eerder vermeld in dit hoofdstuk, deels de leefkosten van een creature: $1 + (\text{waarden} / \text{maximaal aantal waarden})$ energie eenheden per iteratie.

Het aan en uit zetten van zintuigen en hidden neuronen gebeurt door genetische operatoren bij het leerproces.

Leermethoden

In het begin van het onderzoek bestond de dynamische omgeving uit één grote wereld, met een aantal objecten erin. Hier werden creatures in geplaatst, en via een complexe steady state algoritme geëvolueerd. Goede creatures werden behouden en gebruikt om nieuwe creatures te introduceren. Dit steady state algoritme zou stoppen met het introduceren van nieuwe creatures, wanneer er een stabiele, zichzelf in stand houdende, creature populatie zou zijn ontstaan. Tijdens de eerste tests bleek echter dat creatures vaak in lokale optima vast kwamen te zitten, d.w.z. dat ze niet meer verder evolueerden. Daarom is besloten om de dynamische omgeving op te delen in kleinere werelden, met simpeler steady state algoritmen.

Alle waarden in de genen ondergingen in de eerste versies crossover (50% kans) en mutatie (10% kans). Zo kunnen creatures onder invloed van natuurlijke selectie bijvoorbeeld snel worden (lage maximum hoeveelheid energie), sterk (hoge extra energie voor 'attack'), zeer slim of zeer dom (groot of klein netwerk), etc. Maar dit is in de definitieve opzet van het onderzoek weggelaten om de zoekruimte te verkleinen.

2.7.3 Aanpassingen

Het zeer grote netwerk in de beginfase van het onderzoek resulteerde, zoals verwacht, in trage simulatie snelheden en slechte leervoortgang. Daarom zijn de hierboven genoemde objecten en eigenschappen van de omgeving aangepast of verwijderd om de zoekruimte te verkleinen. De aanpassingen zijn geleidelijk ingevoerd.

Hieronder volgt een samenvatting van de belangrijkste aanpassingen:

- Het verwijderen van de objecten lava, water en moving wall blokjes. Hierdoor werd de dynamische omgeving iets simpeler en nam het aantal input neuronen bovendien af (geen informatie-elementen van die objecten).
- Het verwijderen van de vloer laag uit de omgeving. Zonder water en lava objecten, bestaat de vloer laag geheel uit wall blokjes. Er was geen reden meer om de vloer laag te behouden (het maakt voor creatures namelijk niet uit of ze bovenop objecten of lucht lopen, zolang dit geen verschil voor ze oplevert en ze het ook niet kunnen waarnemen).
- Het verwijderen van alle waarden in de genen behalve van het neurale netwerk. Hierdoor wordt tijdens het evolutieproces alleen nog naar goede waarden van het neurale netwerk gezocht.
- 'Relatief' maken van de creatures. Doordat ze een voor- en achterkant hebben, en acties altijd aan de voorkant optreden, hoeven creatures niks te leren over absoluut noord/oost/zuid/west, maar gewoon naar voren/achteren/links/rechts.
- Het verwijderen en veranderen van zintuigen. Door de zintuigen relatief aan de creature te maken en er de helft van te nemen neemt het aantal benodigde input neuronen sterk af. Bovendien geeft de nieuwe mogelijkheid om in de verte te kijken een duidelijker overzicht van de omgeving. Hiervóór waren ze gelimiteerd tot een aantal velden rond zich heen.
- Het verwijderen en veranderen van acties. Doordat de kracht van acties uit de genen is gehaald en nu een constante waarde heeft, en ze relatief aan een creature worden genomen, zijn acties simpeler geworden. Dit samen met het verwijderen van een aantal acties resulteerde in minder output neuronen.
- Het opdelen van de dynamische omgeving in verschillende delen. Hiermee kan een complexe omgeving in stapjes worden opgebouwd en kunnen lokale optima worden overkomen.

Als gevolg van de aanpassingen werd de dynamiek in de test aanzienlijk verkleind zoals met de volgende berekening worden geïllustreerd. Het maximaal (alles aan) netwerk, zoals dat in

het begin van het onderzoek was ontworpen, bestond uit $(420 \times 100) + (100 \times 12) + 100 + 12 = 43.312$ waarden. Het uiteindelijk in dit onderzoek gebruikte netwerk bestaat uit 806 waarden (slechts 1.86% van de maximale omvang en 7.22% van de gemiddelde omvang). De zoekruimte moest dus aanzienlijk worden verkleind om een werkbaar onderzoeksopzet te maken die aansloot op de mogelijkheden van dit scriptieonderzoek.

2.7.4 'Leuke' Bugs

Omdat het hele onderzoek moest worden geprogrammeerd, is er veel tijd besteed aan het identificeren en rectificeren van bugs in het programma. Zoals hiervoor al is aangegeven, werden tijdens het runnen van het programma grote hoeveelheden data opgeslagen, die na elke run of na een crash werden geanalyseerd op fouten en mogelijke verbeteringen. Tijdens het onderzoek zijn er vele fouten en bugs gevonden en verbeterd. Hierna worden drie interessante bugs beschreven.

- Neurale netwerken kregen *elke iteratie* input en gaven dan ook *elke iteratie* output.
 - Er werd aan de netwerken 'gevraagd' na te denken over welke actie ze straks zouden willen uitvoeren, terwijl ze al met een actie bezig waren. In veel gevallen was dit onnodig denkwerk, want er was niets te kiezen. Zo worden de bewegingsacties door het netwerk gekozen in de eerste van de 8 iteraties die zo'n actie duurt. Het 'denken' van het netwerk tijdens de andere 7 iteraties is dus geheel overbodig.
 - Het brein uitschakelen terwijl een actie nog uitgevoerd wordt leverde een aanzienlijke verbetering van de simulatie snelheid op.
- 'Ghost' blokjes
 - Na een tijd een simulatie te hebben gedraaid, bleek de omgeving vol te staan met food blokjes die niet meer verdwenen.
 - Na analyse bleek dit te gebeuren wanneer een food blokje werd geduwd terwijl het al bewoog en dus al geduwd was. Dit mocht eigenlijk niet; het blokje was al in transitie tussen twee velden. Door de extra duw werd de eerste transitie verstoord en er bleef op het veld een nabeeld van een food blokje achter dat niet veranderde, waarop geen acties konden worden uitgevoerd en die ook niet konden worden bewogen/weggeduwd.
- Verkeerde relatieve zintuig richtingen
 - Om het onderzoek te vereenvoudigen is op een bepaald moment de werking van de zintuigen aanmerkelijk aangepast. Die bestond hieruit dat overgestapt werd van zintuigen die een bepaalde plek observeren naar zintuigen die relatief aan een creature een bepaalde richting opkeken.

- Dit bleek niet tot een verbetering van het leergedrag te leiden en daarom werd er na verscheidene simulaties met slecht leergedrag in detail naar het gedrag van de creatures gekeken. Daarbij bleek dat een redelijk goede creature onverwachte zeer uiteenlopende beslissingen kon nemen. Als het bijvoorbeeld een food blokje zag, liep het creature er soms goed op af maar soms liep het in een compleet andere richting.
- Om de oorzaak van dit gedrag te onderzoeken is in detail naar de input van het netwerk gekeken, en toen bleek dat de relatieve zintuigrichtingen niet correct waren. Bijvoorbeeld als een creature noordelijk gericht was, dan keek het linkerzintuig correct naar het westen maar als de creature oostelijk gericht was, dan keek het linkerzintuig naar het oosten in plaats van naar het noorden.

3 UITVOERING ONDERZOEK

Op basis van het testen en aanpassen van de verschillende delen van het onderzoek is de volgende definitieve opzet van het onderzoek gekozen.

- Het onderzoek bestaat uit verschillende fasen; elke fase bestaat uit een bepaalde wereld met een bepaald leerdoel.
- De elkaar opvolgende fasen hebben een oplopende moeilijkheidsgraad; er is complexer gedrag nodig om het leerdoel te behalen.
- Per fase wordt bijgehouden hoeveel generaties van creatures er zijn gesimuleerd voordat het leerdoel is bereikt; ofwel hoe snel het leerdoel wordt bereikt.
- Voor zowel het Sequential Neuraal Netwerk als het Incremental Neuraal Netwerk wordt alleen voor de eerste fase een willekeurige populatie gegenereerd. Voor elke volgende fase wordt de eindpopulatie van de vorige fase gebruikt.
- Als control wordt bovendien ook elke fase uitgevoerd met een willekeurig gegenereerde populatie van het Sequential Neuraal Netwerk. Dit netwerk heeft dezelfde architectuur als het Sequential Neuraal Netwerk (en het Incremental Neuraal Netwerk van wereld 4), maar heeft niet in een eerdere wereld geleerd en begint dus in elke wereld met een schone lei.
- Alle fasen worden door elk type netwerk drie keer doorlopen.

De kenmerken van de verschillende fasen, de werelden en hun leerdoelen, zijn kort samengevat in tabel 4 en worden verder in dit hoofdstuk in detail beschreven.

| | Kenmerken | Doel | Subdoel(en) |
|-----------------|--|--|-----------------------|
| Wereld 1 | 3 food blokjes willekeurig geplaatst 1 creature | Voedsel vinden | Doelgericht bewegen |
| Wereld 2 | 3 food blokjes willekeurig geplaatst 1 creature | Voedsel eten | Als in wereld 1 |
| Wereld 3 | 3 food blokjes willekeurig geplaatst 1 creature | Zo lang mogelijk leven | Als in wereld 1 en 2 |
| Wereld 4 | 5 food blokjes willekeurig geplaatst 1 creature 3 andere creatures aanwezig | Nakomelingen maken | Als in wereld 1 t/m 3 |
| Wereld 5 | Grote wereld Veel food blokjes willekeurig geplaatst Initiële populatie van 50 creatures | Populatie zelfstandig in stand houden, ofwel niet uitsterven | Als in wereld 1 t/m 4 |

Tabel 4 Samenvatting van kenmerken en doelen van werelden in dit onderzoek

3.1 Wereld 1

Opbouw

Deze wereld is 16 bij 16 velden groot. De rand van de wereld (1 veld breed) bestaat uit wall blokjes. Het daadwerkelijke veld waarin creatures kunnen bewegen is dan 14 bij 14 velden groot. Er zijn altijd, op willekeurige velden, 3 food blokjes aanwezig in de wereld. De hoeveelheid voedsel in de food blokjes is het maximum. De hoeveelheid neemt in deze wereld niet automatisch af.

Leerdoel

Een creature leert in deze wereld food blokjes vinden. Wanneer de creature naast een food blokje staat (en met zijn voorkant hiernaar georiënteerd is (zie hoofdstuk 2.7), worden alle food blokjes op nieuwe velden geplaatst ten minste 5 velden van de positie van de creature. Als een creature drie keer een food blokje heeft gevonden, is het doel behaald. Maar een creature kan zo lang het leeft food blokjes blijven vinden en krijgt hier ook punten voor.

Creatures

De genen van de populatie worden elke generatie een voor een in de wereld geplaatst als een creature (zie hoofdstuk 2.5). Ze worden willekeurig in het midden gebied (4 bij 4) van de wereld geplaatst. Creatures beginnen met de maximale hoeveelheid energie. Wanneer een nieuwe creature in de wereld wordt geplaatst, worden eerst alle feromoon sporen (van de vorige creature) weggehaald en worden de food blokjes willekeurig op andere velden geplaatst.

Fitness

De maximale afstand in deze wereld is $\sqrt{(14^2 + 14^2)} \approx 20$. De score die creatures krijgen is (maximale afstand in de wereld – afstand tot food blokje) ofwel $(20 - \sqrt{((x\text{-positie creature} - x\text{-positie food})^2 + (y\text{-positie creature} - y\text{-positie food})^2)})$. De maximale score (als een creature naast het food blokje staat) is dan $20 - 1 = 19$, *per gevonden doel*. De fitness is dan de score voor de dichtst behaalde afstand tot de huidige food blokjes, plus de maximale score maal het aantal keer dat de creature een food blokje heeft gevonden.

Neurale netwerken

Het Sequential Neuraal Netwerk is, zoals in elke wereld, volledig opgebouwd. Zo kan het bijvoorbeeld de actie 'mate' kiezen, terwijl de creature alleen is in deze wereld. Het bevat $((5 \times 5) \times 25) + (25 \times 6) = 775$ verbindingen met bijbehorende gewichten en $25 + 6 = 31$ bias

waarden. Ofwel, $775 + 31 = 806$ waarden die moeten worden getraind (zie ook hoofdstuk 2.4).

Het Incremental Neuraal Netwerk is in deze wereld op de volgende manier opgebouwd:

Input neuronen (per zintuig):

- De afstand tot het object als het object een wall blokje is.
- De afstand tot het object als het object een food blokje is.
- Hoe lang geleden op dit veld een creature aanwezig is geweest. (feromoon)

Hidden neuronen: 5

Output neuronen: De 4 bewegings acties: move forward, move backward, turn left en turn right.

De inputs voor 'het waarnemen van en afstand tot een creature' en 'hoeveelheid' (energie of voedsel) bestaan nog niet, aangezien er geen andere creatures aanwezig zijn om te waarnemen en 'hoeveelheid' nog overbodig is (voedsel hoeveelheid neemt niet af en energie hoeveelheid kan niet worden aangevuld). De 'eat' en 'mate' outputs bestaan nog niet, aangezien beide acties in deze wereld toch niet succesvol kunnen worden uitgevoerd. (mate: geen andere creatures om mee te maten; eat: food blokjes verplaatsen zodra creature ernaast staat)

Er zijn in totaal $3 \times 5 = 15$ input neuronen. Het aantal verbindingen, en bijbehorende gewichten, in het netwerk is $(15 \times 5) + (5 \times 4) = 95$. Het aantal bias waarden in het netwerk is $5 + 4 = 9$. In totaal zijn er dus $95 + 9 = 104$ waarden die moeten worden getraind. Dit is ongeveer een achtste van de 806 waarden van het SNN.

3.2 Wereld 2

Opbouw

Deze wereld is 16 bij 16 velden groot. De rand van de wereld (1 veld breed) bestaat uit wall blokjes. Het daadwerkelijke veld waarin creatures kunnen bewegen is dan 14 bij 14 velden groot. Er zijn altijd, op willekeurige velden, 3 food blokjes aanwezig in de wereld. De hoeveelheid voedsel in de food blokjes is het maximum. De hoeveelheid neemt in deze wereld niet automatisch af.

Leerdoel

Een creature leert in deze wereld voedsel te eten. Wanneer een creature drie keer succesvol de 'eat' actie heeft uitgevoerd op een food blokje, worden alle food blokjes op nieuwe velden geplaatst ten minste 5 velden van de positie van de creature. Als een creature dit drie keer heeft gedaan, is het doel behaald. Een creature krijgt meer energie door te eten, waardoor het langer in leven kan blijven. En een creature kan zo lang het leeft voedsel blijven eten en krijgt hier ook punten voor.

Creatures

De genen van de populatie worden elke generatie één voor één in de wereld geplaatst als een creature. Ze worden willekeurig in het midden gebied (4 bij 4) van de wereld geplaatst. Creatures beginnen met de maximale hoeveelheid energie. Wanneer een nieuwe creature in de wereld wordt geplaatst, worden eerst alle feromoon sporen (van de vorige creature) weggehaald en worden de food blokjes willekeurig op andere velden geplaatst.

Fitness

De maximale afstand in deze wereld is $\sqrt{(14^2 + 14^2)} \approx 20$. De score die creatures krijgen is (maximale afstand in de wereld – afstand tot food blokje) ofwel $(20 - \sqrt{((x\text{-positie creature} - x\text{-positie food})^2 + (y\text{-positie creature} - y\text{-positie food})^2)})$. De maximale score is dan $20 - 1 = 19$, *per drie maal eten van voedsel*. De fitness is dan de score voor de dichtst behaalde afstand tot de huidige food blokjes, plus de maximale score maal het aantal keer dat de creature drie maal voedsel heeft gegeten.

Neurale netwerken

Het Sequential Neuraal Netwerk is, zoals in elke wereld, volledig opgebouwd. Het bevat 775 verbindingen met bijbehorende gewichten en 31 bias waarden, ofwel 806 waarden die moeten worden getraind.

Het Incremental Neuraal Netwerk is in deze wereld op de volgende manier opgebouwd:
Het bestaat uit het netwerk van wereld 1, waaraan voor deze wereld de volgende neuronen zijn toegevoegd:

Toegevoegde input neuronen: Geen.

Toegevoegde hidden neuronen: 5

Toegevoegde output neuronen: 1

- De 'eat' actie

Er worden in elke wereld hidden neuronen toegevoegd om het netwerk de ruimte te geven nieuwe dingen te leren. Aangezien het doel in deze wereld berust op voedsel eten, is de 'eat' output neuron toegevoegd. Er zijn nog steeds 15 input neuronen. Het aantal hidden neuronen is nu 10. Het aantal output neuronen is toegenomen, en wel met één voor een totaal van 5. Het aantal verbindingen, en bijbehorende gewichten, in het netwerk is dan $(15 \times 10) + (10 \times 5) = 200$. Het aantal bias waarden in het netwerk is nu $10 + 5 = 15$. In totaal zijn er dus $200 + 15 = 215$ waarden die moeten worden getraind, waarvan er 104 al eerder zijn getraind in wereld 1. Het aantal waarden is 2.1 keer zoveel als in wereld 1. Dit is ongeveer een kwart van de 806 waarden van het SNN.

3.3 Wereld 3

Opbouw

Deze wereld is 16 bij 16 velden groot. De rand van de wereld (1 veld breed) bestaat uit wall blokjes. Het daadwerkelijke veld waarin creatures kunnen bewegen is dan 14 bij 14 velden groot. Er zijn altijd, op willekeurige velden, 3 food blokjes aanwezig in de wereld. De beginhoeveelheid voedsel van de food blokjes is het maximum. De hoeveelheid neemt in deze wereld *wel* automatisch af. Wanneer een food blokje verdwijnt (omdat zijn voedsel hoeveelheid nul is), wordt er direct een nieuw food blokje in de wereld geplaatst.

Leerdoel

Een creature leert in deze wereld zo lang mogelijk leven. Als een creature 80% van zijn maximale levensspan heeft bereikt (80% van 5000 iteraties), dan is het doel behaald. Een creature krijgt meer energie door food blokjes te vinden (geleerd in wereld 1) en hiervan te eten (geleerd in wereld 2), waardoor het langer in leven blijft.

Creatures

De genen van de populatie worden elke generatie een voor een in de wereld geplaatst als een creature. Ze worden willekeurig in het midden gebied (4 bij 4) van de wereld geplaatst. Creatures beginnen met de maximale hoeveelheid energie. Wanneer een nieuwe creature in de wereld wordt geplaatst, worden eerst alle feromoon sporen (van de vorige creature) weggehaald en worden de food blokjes willekeurig op andere velden geplaatst.

Fitness

De score die creatures krijgen is simpelweg hun leeftijd op het moment van sterven.

Neurale netwerken

Het Sequential Neuraal Netwerk is, zoals in elke wereld, volledig opgebouwd. Het bevat 775 verbindingen met bijbehorende gewichten en 31 bias waarden, ofwel 806 waarden die moeten worden getraind.

Het Incremental Neuraal Netwerk is in deze wereld op de volgende manier opgebouwd:

Het bestaat uit het netwerk van wereld 2, waaraan voor deze wereld de volgende neuronen zijn toegevoegd:

Toegevoegde input neuronen: Geen.

Toegevoegde hidden neuronen: 5

Toegevoegde output neuronen: Geen.

Er worden in elke wereld hidden neuronen toegevoegd om het netwerk de ruimte te geven nieuwe dingen te leren. Er zijn nog steeds 15 input neuronen. Het aantal hidden neuronen is nu 15. Het aantal output neuronen is nog steeds 5. Het aantal verbindingen, en bijbehorende gewichten, in het netwerk is dan $(15 \times 15) + (15 \times 5) = 300$. Het aantal bias waarden in het netwerk is nu $15 + 5 = 20$. In totaal zijn er dus $300 + 20 = 320$ waarden die moeten worden getraind, waarvan er 215 al eerder zijn getraind in wereld 2. Het aantal waarden is 1.5 keer zoveel als in wereld 2. Dit is ongeveer $4/10$ van de (806) waarden van het SNN.

3.4 Wereld 4

Opbouw

Deze wereld is 16 bij 16 velden groot. De rand van de wereld (1 veld breed) bestaat uit wall blokjes. Het daadwerkelijke veld waarin creatures kunnen bewegen is dan 14 bij 14 velden groot. Er zijn altijd, op willekeurige velden, 5 food blokjes aanwezig in de wereld. De beginhoeveelheid voedsel van de food blokjes is het maximum. De hoeveelheid neemt in deze wereld *wel* automatisch af. Wanneer een food blokje verdwijnt (omdat zijn voedsel hoeveelheid nul is), wordt er direct een nieuw food blokje in de wereld geplaatst. Hoewel er in deze wereld 'mate' acties worden uitgevoerd, resulteren deze *niet* in nakomelingen ook al zijn ze succesvol uitgevoerd. Hierdoor wordt de kleine wereld niet overvol met nakomelingen die (mogelijk) zelf ook voedsel eten en weer nakomelingen maken. Bovendien zijn deze nakomelingen het resultaat van een 'mate' actie tussen een creature en zijn kloon; crossover heeft dan geen effect.

Leerdoel

Een creature leert in deze wereld met andere creatures nakomelingen te maken, terwijl het tegelijkertijd zijn energie voorraad op peil houdt. Het doel is bereikt als een creature 2 succesvolle 'mate' acties heeft uitgevoerd en zijn leeftijd minimaal de helft van zijn maximale leeftijd is (50% van 5000 iteraties). Hiervoor is het nodig om energie te verkrijgen door food blokjes te vinden (geleerd in wereld 1) en hiervan te eten (geleerd in wereld 2), waardoor het lang in leven blijft (wereld 3). Aangezien de 'mate' actie zoveel energie en tijd kost, moeten creatures leren omgaan met de hoeveelheid energie die ze tot hun beschikking hebben en wanneer ze ervoor moet kiezen om voedsel te zoeken of nakomelingen te maken.

Creatures

De genen van de populatie worden elke generatie een voor een in de wereld geplaatst als een creature. Ze worden willekeurig in het midden gebied (4 bij 4) van de wereld geplaatst. Creatures beginnen *niet* met de maximale hoeveelheid energie, maar 75% hiervan (750 eenheden ipv 1000). Dit is gedaan om het voedsel zoekgedrag extra te stimuleren. Om succesvol 'mate' acties uit te voeren, moet de creature beschikking hebben tot andere creatures. Hiervoor wordt simpelweg de huidige creature drie maal 'gekloond'. Deze klonen kunnen niet sterven; zodra hun energie niveau onder de 200 energie eenheden valt, wordt het bijgevuld (met 250 tot 500 eenheden). Deze klonen gedragen zich hetzelfde als de huidige creature, aangezien hun hersenen uit een kopie van het creature's netwerk bestaan. Deze klonen worden echter geen fitness toegewezen. Er is voor deze kloon manier gekozen,

zodat creatures leerden omgaan met echte slimme andere creatures; er is een 'dummy creature' manier uitgeprobeerd die willekeurig acties uitvoerden, maar dit leverde slechte resultaten op.

Wanneer een nieuwe creature in de wereld wordt geplaatst, worden eerst alle feromoon sporen (van de vorige creatures) weggehaald en worden de food blokjes willekeurig op andere velden geplaatst.

Fitness

De score die creatures krijgen is als volgt: Ze krijgen hun leeftijd gedeeld door 50 aantal punten, ofwel maximaal 100 punten ($5000 / 50 = 100$). Als ze geen enkele keer succesvol een 'mate' actie hebben gedaan, krijgen ze hier echter maximaal 50 punten voor. Voor de eerste succesvolle 'mate' actie krijgen ze 6 punten, en voor elke opvolgende succesvolle 'mate' actie steeds minder, namelijk 3, 2, 1, 1, 1, 0, 0, 0, 0, 0, etc.

Neurale netwerken

Het Sequential Neuraal Netwerk is, zoals in elke wereld, volledig opgebouwd. Het bevat 775 verbindingen met bijbehorende gewichten en 31 bias waarden, ofwel 806 waarden die moeten worden getraind.

Het Incremental Neuraal Netwerk is in deze wereld op de volgende manier opgebouwd:

Het bestaat uit het netwerk van wereld 3, waaraan voor deze wereld de volgende neuronen zijn toegevoegd:

Toegevoegde input neuronen: 2 (per zintuig)

- De afstand tot het object als het object een creature is.
- Hoeveel het object bevat. (energie bij creatures of voedsel bij food blokjes)

Toegevoegde hidden neuronen: 10

Toegevoegde output neuronen: 1

- De 'mate' actie.

Er zijn in deze wereld dubbel zoveel hidden neuronen toegevoegd omdat het netwerk behoorlijk complex gedrag moet leren. Er zijn nu 25 input neuronen. Het aantal hidden neuronen is nu ook 25. Het aantal output neuronen is nu 6. Het aantal bias waarden in het netwerk is nu $25 + 6 = 31$. In totaal zijn er dus 806 waarden die moeten worden getraind, waarvan er 320 al eerder zijn getraind in wereld 3. Het aantal waarden is 2.5 keer zoveel als in wereld 3. Het netwerk is nu even groot als het SNN.

3.5 Wereld 5

Opbouw

Deze wereld is 30 bij 30 velden groot. De rand van de wereld (1 veld breed) bestaat uit wall blokjes. Het daadwerkelijke veld waarin creatures kunnen bewegen is dan 28 bij 28 velden groot. Dit zijn vier keer zoveel velden als in de vorige werelden. 'Mate' acties in de wereld resulteren *we/* in nakomelingen.

Er worden in het begin, op willekeurige velden, 25 food blokjes in de wereld geplaatst. De beginhoeveelheid voedsel van de foodblokjes is het maximum. De hoeveelheid neemt in deze wereld *we/* automatisch af. Er zijn maximaal 25 food blokjes in de wereld aanwezig. Als het aantal food blokjes lager is dan 25, dan wordt het aantal food blokjes aangevuld met maximaal 1 food blokje per 10 iteraties. De totale hoeveelheid voedsel in de wereld is hierdoor beperkt.

Leerdoel

In deze wereld moet de populatie van creatures zichzelf in stand houden. Als na 3 uur er nog steeds een populatie van creatures in de wereld aanwezig is, is het doel behaald.

Creatures

De eindpopulatie van wereld 4 vormt de basis voor deze wereld. De beste 50 creatures worden *tegelijktijd* in de wereld geplaatst op willekeurige velden. Ze beginnen net als in wereld 4 met 750 energie eenheden. Hierna worden er door de onderzoeker geen creatures meer in de wereld geplaatst.

Fitness

In deze wereld krijgen creatures geen score. Het enige wat uitmaakt is het overleven van de populatie als geheel, niet van individuele creatures.

Neurale netwerken

Het Sequential Neuraal Netwerk en het Incremental Neuraal Netwerk zijn beide volledig opgebouwd. Ze bevatten 775 verbindingen met bijbehorende gewichten en 31 bias waarden, ofwel 806 waarden die in de vorige werelden zijn getraind. De mutatie kansen van de nodes in de netwerken zijn nu 0.01 voor beide typen netwerken.

4 RESULTATEN

4.1 Introductie

In hoofdstuk 3 is het uitgevoerde onderzoek in detail beschreven, alsook de keuzes die daaraan ten grondslag liggen. Zoals daar is aangegeven, doorloopt elk type neurale netwerk alle fasen van dit onderzoek meerdere keren. Ofwel, elk type neurale netwerk heeft per wereld meerdere resultaten. De resultaten hiervan worden in de volgende paragrafen, één wereld per paragraaf, gepresenteerd.

Hierbij wordt eerst één van de behaalde resultaten, per type neurale netwerk, in een grafiek weergegeven, zodat de verschillen in leerverloop tussen de typen netwerken kunnen worden geïllustreerd. In deze grafieken is het aantal creatures van de populatie dat het doel van de wereld heeft behaald uitgezet tegen het aantal generaties.

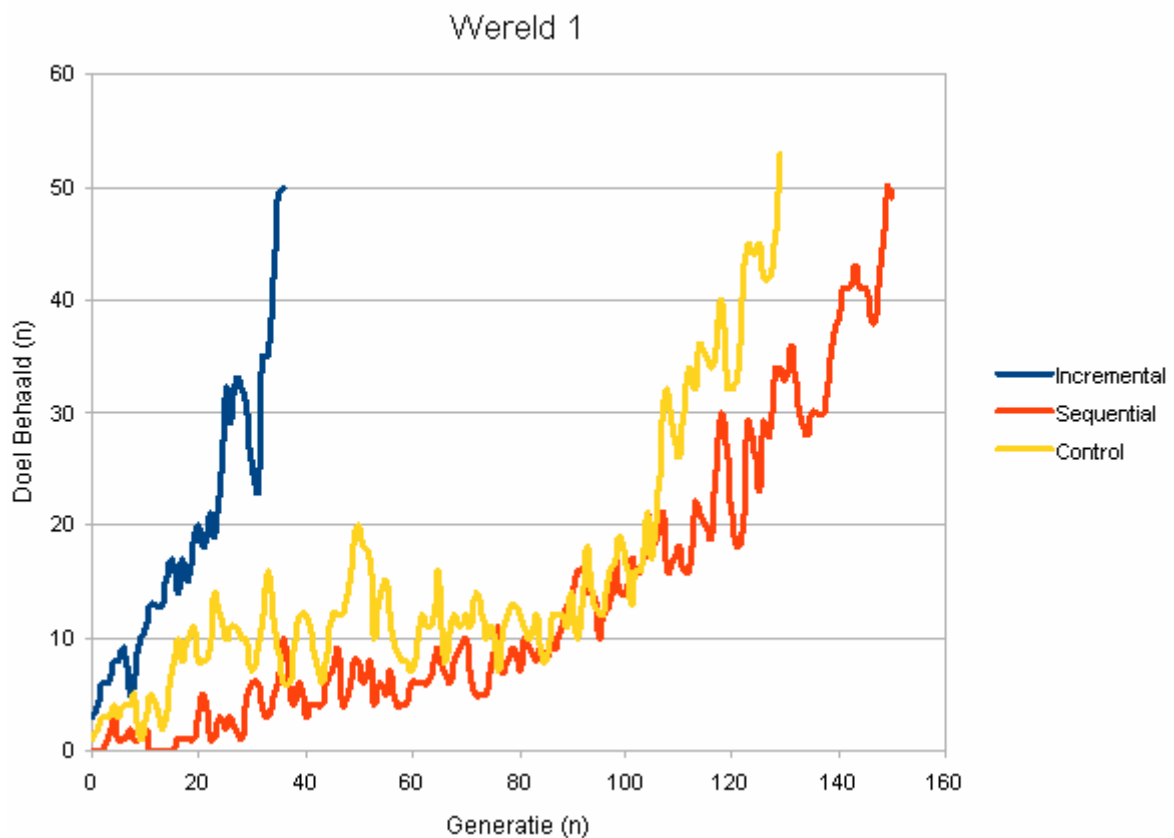
Hierna worden de typen neurale netwerken vergeleken door te kijken naar het aantal generaties dat benodigd was om het leerdoel van een wereld te behalen. Hiervoor worden de meerdere resultaten van die wereld per type netwerk gemiddeld. Aangezien de control na wereld 1 nooit meer het leerdoel van een wereld behaalt, wordt het in deze vergelijkingen weggelaten.

Zoals beschreven in hoofdstuk 2.5 heeft een populatie het leerdoel van een wereld bereikt als minstens 45 creatures van de populatie het doel hebben behaald, twee generaties achter elkaar.

4.2 Onderzoeksresultaten

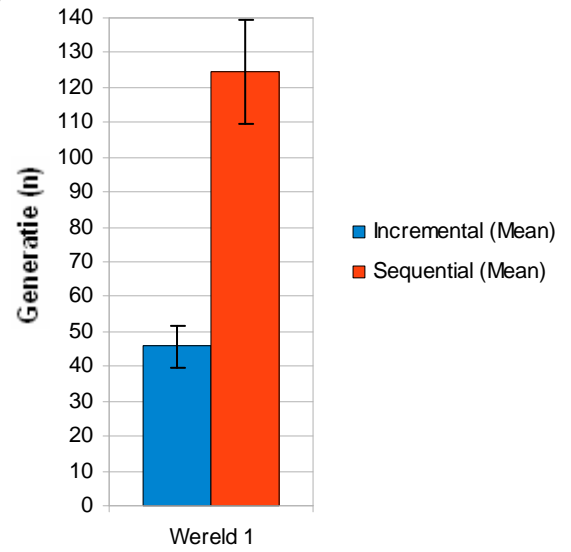
Resultaten wereld 1

In figuur 8 is duidelijk te zien dat het Incremental Neuraal Netwerk veel sneller het leerdoel bereikt dan de andere netwerken. Zoals verwacht is er nauwelijks verschil tussen het Sequential Neuraal Netwerk en de control, aangezien beide netwerken op dezelfde manier zijn opgebouwd en ze allebei willekeurig zijn gegenereerd. Er valt bovendien op te merken dat het SNN en de control veel meer schommelingen ondergaan dan het INN, die (op één grote terugval na) bijna constant toeneemt.



Figuur 8

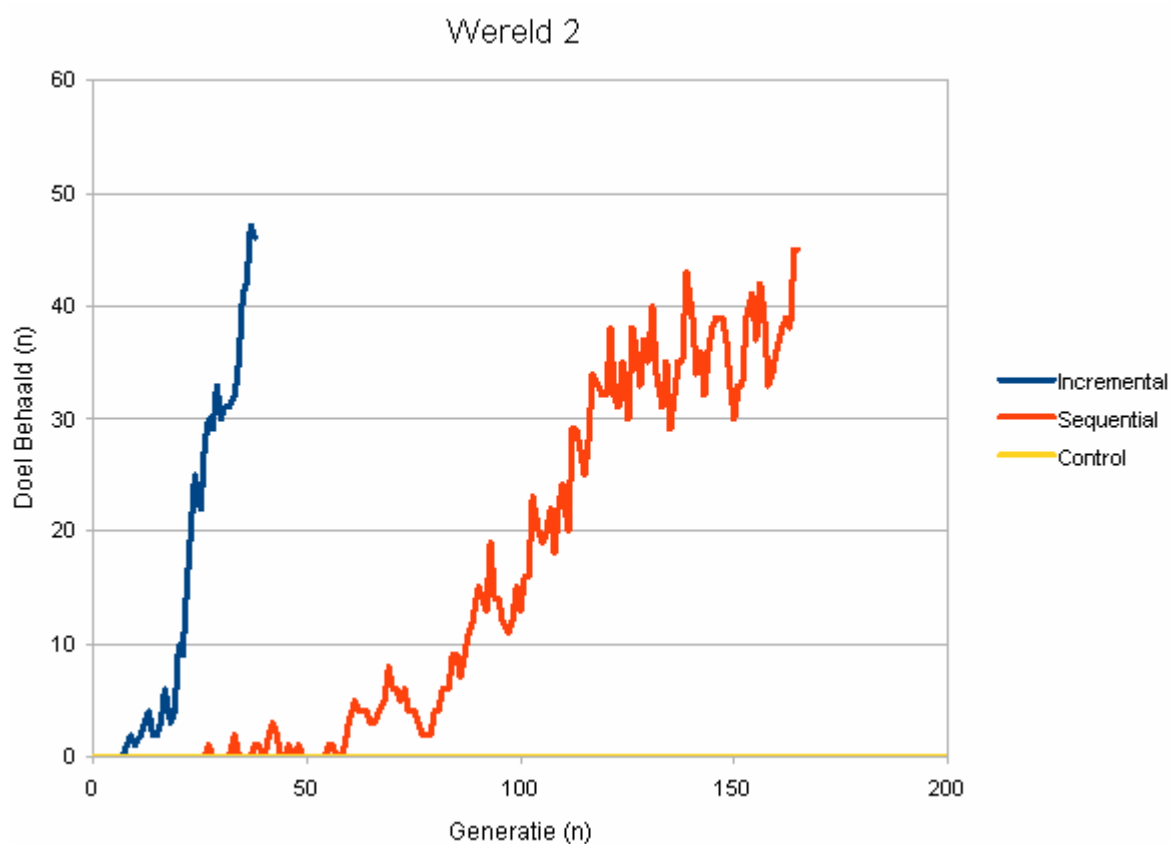
Zoals in figuur 9 duidelijk zichtbaar is, doet het INN er significant ($p = 0.004$) minder lang over om het leerdoel te bereiken dan het SNN.



Figuur 9

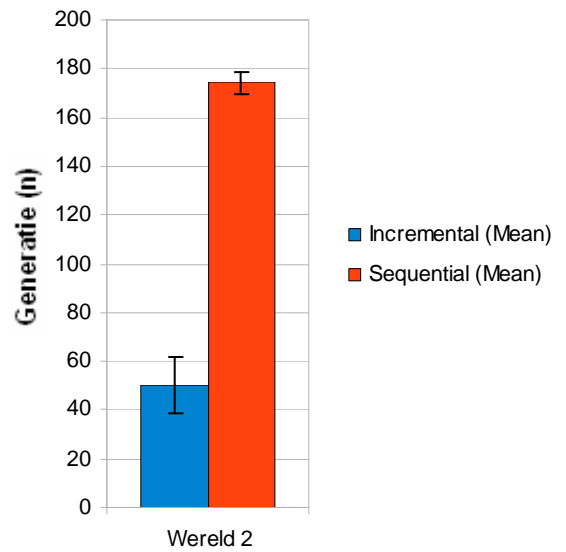
Resultaten wereld 2

In figuur 10 is duidelijk te zien dat het Incremental Neuraal Netwerk veel sneller het leerdoel bereikt dan de andere netwerken. Het blijkt voor de control in deze wereld onmogelijk om het doel te bereiken. Dit was verwacht, omdat het leerdoel in deze (en volgende werelden) complexer is en uitgaat van eerder geleerd gedrag. Het INN ondergaat zeer weinig schommelingen en stijgt zeer sterk, terwijl het SNN een langzame start heeft, onderhevig is aan grotere terugvallen en vooral nabij het einde moeite heeft om voortgang te maken.



Figuur 10

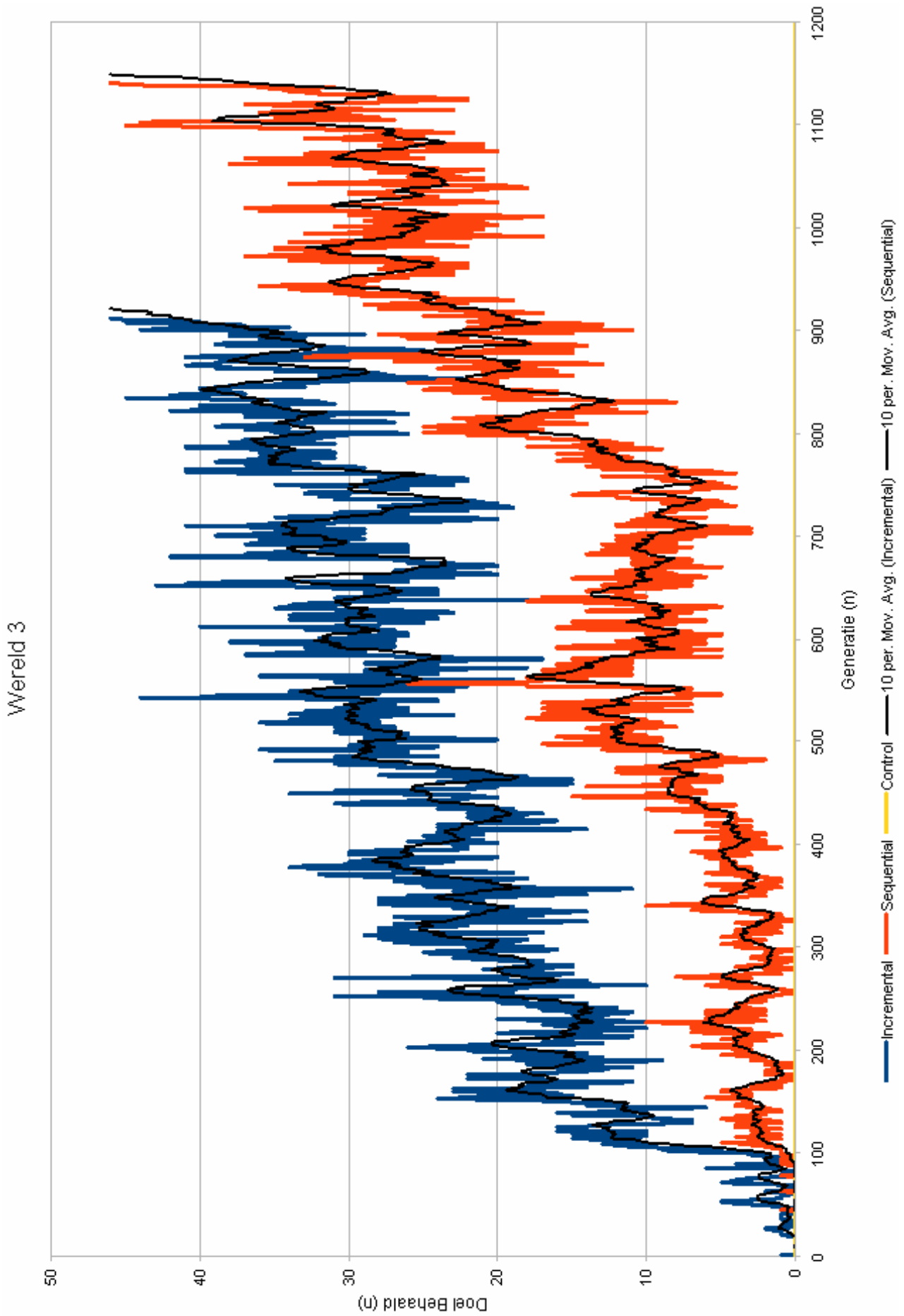
Zoals in figuur 11 duidelijk zichtbaar is, doet ook in deze wereld het INN er significant ($p = 0.0003$) minder lang over om het leerdoel te bereiken dan het SNN. Het INN doet er ongeveer even lang over als in wereld 1, terwijl het SNN er ongeveer 40 generaties langer over doet.



Figuur 11

Resultaten wereld 3

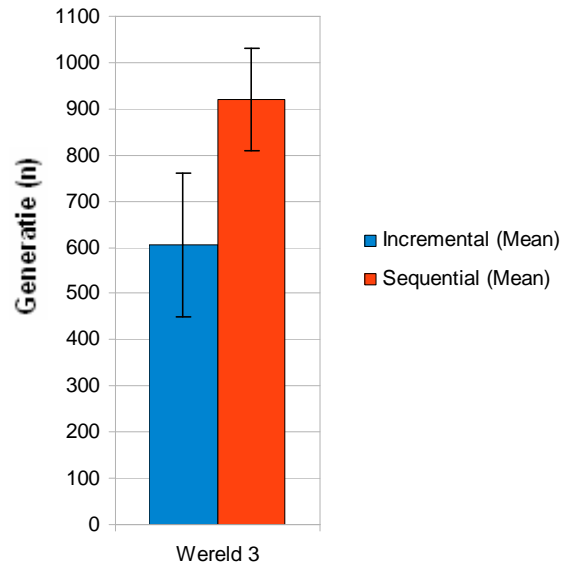
In figuur 12 is te zien dat het Incremental Neuraal Netwerk sneller het leerdoel bereikt dan de andere netwerken. Net als in wereld 2 blijkt het voor de control in deze wereld onmogelijk om het doel te bereiken. Dit was verwacht, omdat het leerdoel in deze (en volgende werelden) complexer is en uitgaat van eerder geleerd gedrag. Het is duidelijk zichtbaar dat het INN vanaf het begin meer succes heeft dan het SNN en rond generatie 100 neemt het INN een grote voorsprong op het SNN en geeft dit niet meer op. Het valt ook op te merken dat het SNN rond generatie 550 een langzame terugval ondergaat en hier pas rond generatie 750 van herstelt, terwijl het INN sinds generatie 100 constant stijgt.



Figuur 12

Zoals in figuur 13 zichtbaar is, doet het INN in deze wereld er niet significant ($p = 0.0885$) minder lang over om het leerdoel te behalen dan het SNN.

Het verschil tussen de leertijd voor het INN en SNN is niet meer zo groot als in wereld 1 en 2, waar het SNN respectievelijk 2.7 en 3.5 keer zoveel tijd nodig had als het INN, terwijl het nu maar 1.5 keer zo lang erover deed.



Figuur 13

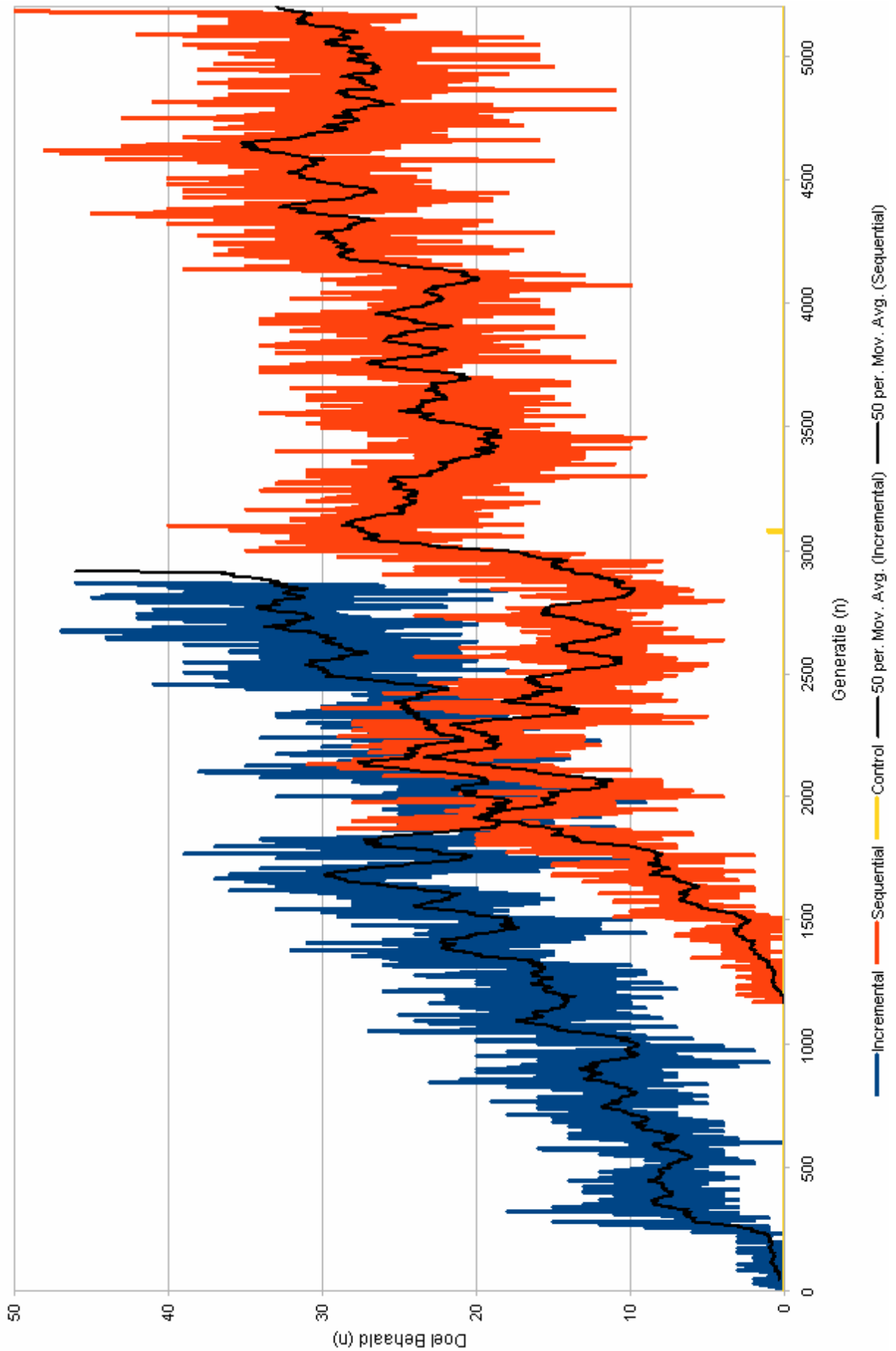
Resultaten wereld 4

In figuur 14 is duidelijk te zien dat het Incremental Neuraal Netwerk sneller het leerdoel bereikt dan de andere netwerken.

Net als in de vorige werelden blijkt het voor de control in deze wereld onmogelijk om het doel te bereiken. Dit was verwacht, omdat het leerdoel in deze (en volgende werelden) complexer is en uitgaat van eerder geleerd gedrag. Er is echter wel een zeer kleine uitstulping rond generatie 3100. De control beschikte 5 generaties lang over een enkele creature die het doel van de wereld had behaald. Maar deze ging blijkbaar verloren voordat de populatie van zijn genen kon profiteren, of zijn succes was toeval. Het illustreert echter wel dat zelfs een willekeurig neuraal netwerk het doel van de wereld mogelijk zou kunnen behalen, na extreem veel generaties, simpelweg door willekeur.

Het is duidelijk zichtbaar dat het INN vanaf het begin veel meer succes heeft dan het SNN: vanaf praktisch de eerste generatie bezit het INN over creatures die het leerdoel behalen, terwijl het SNN dit pas bereikt rond generatie 1200. Vanaf generatie 1200 stijgt het SNN echter sterk en, gecombineerd met een terugval van het INN, zitten beide netwerken rond generatie 1700 op hetzelfde niveau. Het INN herstelt echter van de terugval en ondergaat een constante stijging tot het eind, terwijl het SNN nu een terugval ondergaat. Nadat het INN het eind heeft bereikt, herstelt het SNN van zijn terugval en ondergaat een langzame maar zekere stijging totdat het ook uiteindelijk het eind bereikt.

Wereld 4

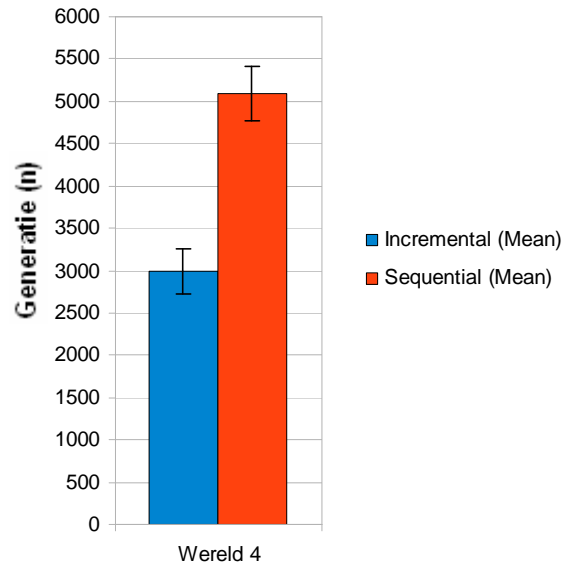


Figuur 14

Zoals in figuur 15 duidelijk zichtbaar is, doet ook in deze wereld het INN er significant ($p = 0.0035$) minder lang over om het leerdoel te bereiken dan het SNN.

Het SNN heeft nu 1.7 keer zoveel tijd nodig als het INN; dit is een lichte toename in vergelijking met de 1.5 in de vorige wereld.

Uit deze resultaten blijkt dat de betere leersnelheid van het INN niet kan worden verklaard door de grootte van het neurale netwerk: in wereld 4 zijn beide netwerken namelijk even groot. Als de leersnelheid alleen een functie was van de grootte van het netwerk, dan zouden het INN en SNN in deze wereld geen significant verschil in leersnelheid moeten tonen. Dit doen ze echter wel. Blijkbaar resulteert het incrementale leerproces van het INN in snellere leersnelheden dan het sequentiële leerproces van het SNN.



Figuur 15

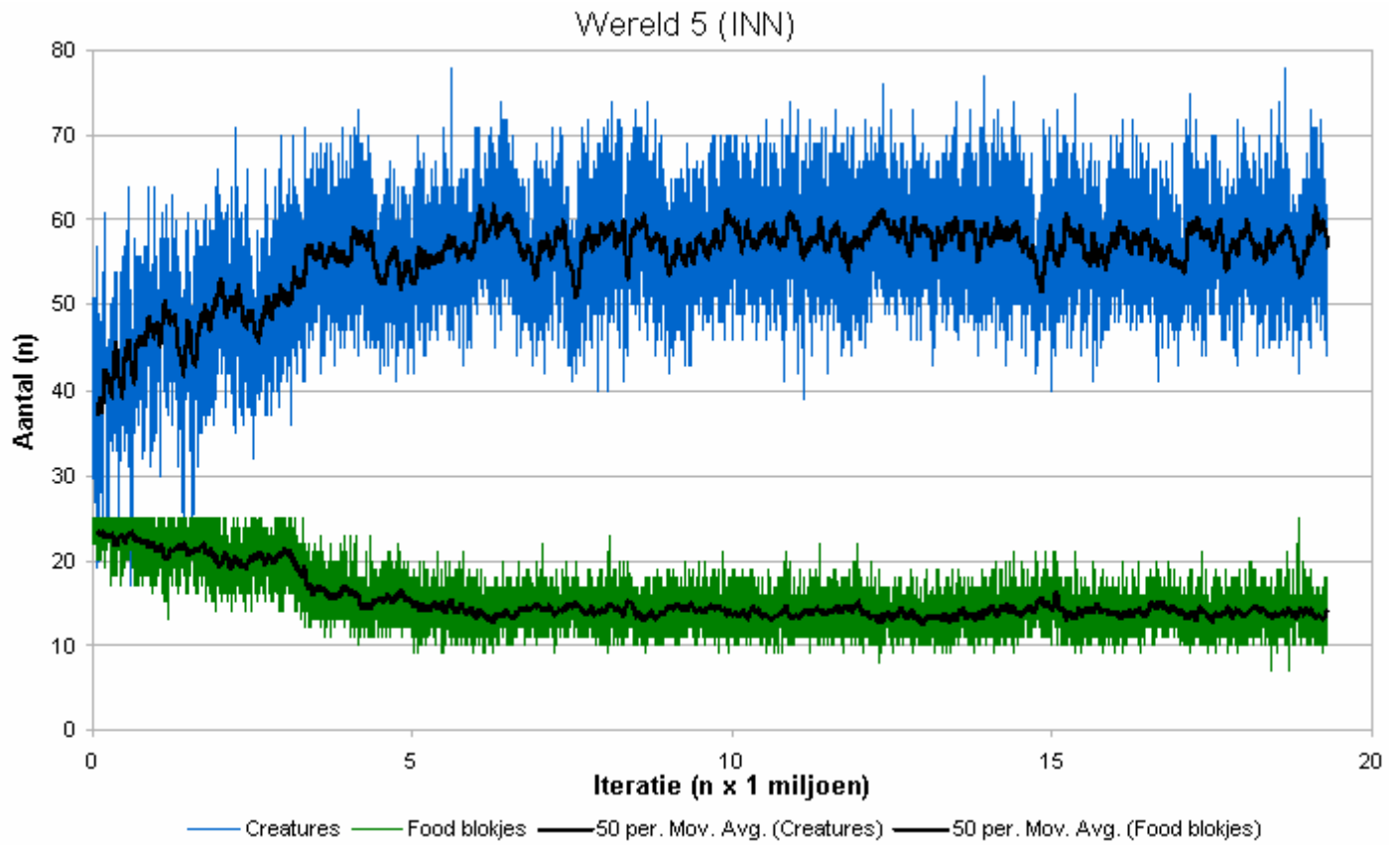
Resultaten wereld 5

In figuur 16 en figuur 17 zijn voor elk van de typen netwerken één van de behaalde resultaten in wereld 5 weergegeven. De andere behaalde resultaten zijn gelijkend. De control is niet weergegeven omdat het niet tot een zichzelf in stand houdende populatie leidde. Dit was verwacht, aangezien de control ook in wereld 5 met een populatie begon die uit willekeurig gegenereerde creatures bestond, waardoor ze niet hadden geleerd zich voort te planten en bijna direct uitstierven.

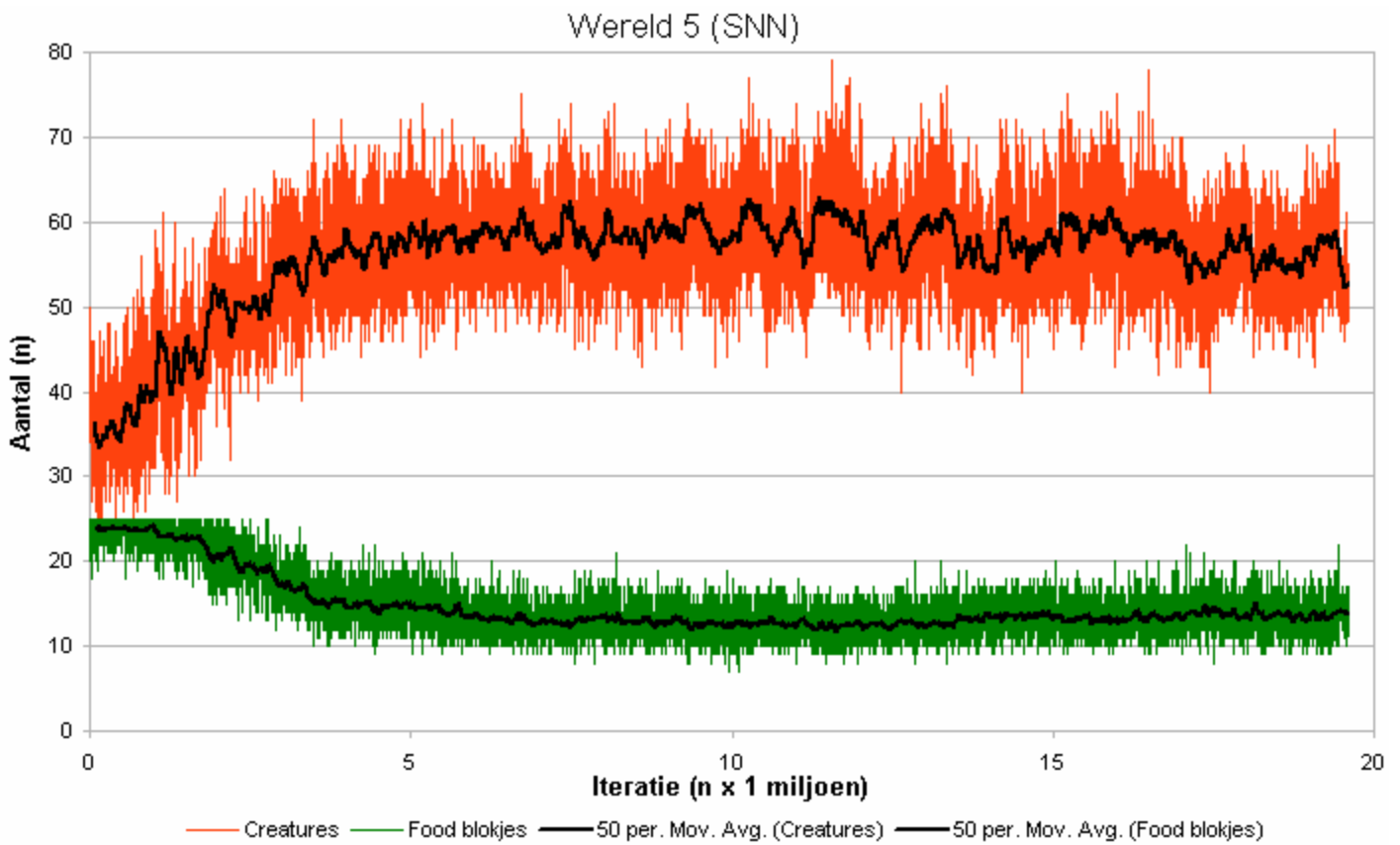
Zoals duidelijk te zien is, neemt het aantal creatures in de populatie geleidelijk toe, en het aantal food blokjes af, totdat een balans wordt bereikt waarbij het aantal creatures tussen de 45 en 70 varieert en het aantal food blokjes tussen de 10 en 20 varieert.

Omdat het aantal food blokjes in de wereld beperkt is, zijn creatures in competitie voor voedsel. De genen van creatures die minder succesvol zijn in eten en 'maten' verdwijnen uit de populatie door natuurlijke selectie, en de genen die leiden tot succesvol eten en 'maten' domineren hierdoor de populatie. Dit leidt tot een zichzelf in stand houdende populatie.

Zowel het INN als het SNN behalen in deze wereld het doel.



Figuur 16 Verloop INN in wereld 5



Figuur 17 Verloop SNN in wereld 5

5 CONCLUSIE EN DISCUSSIE

Uit de onderzoeksresultaten blijkt dat het Incremental Neuraal Netwerk (INN) in drie van de vier werelden (wereld 1, 2 en 4) significant minder generaties nodig heeft om het gestelde leerdoel te behalen dan het Sequential Neuraal Netwerk (SNN). Deze resultaten bevestigen de onderzoekshypothese dat het INN sneller evolueert in een dynamische omgeving dan het SNN. Uit het onderzoek blijkt verder dat de grotere leersnelheid van het INN niet wordt veroorzaakt door het verschil in grootte van het netwerk, want ook in wereld 4, waar beide netwerken even groot zijn, is het INN sneller. Daarnaast laten de resultaten van wereld 5 zien dat er, behalve het verschil in leersnelheid, geen verschil is tussen de functionaliteit van de netwerken; beide typen netwerken leiden namelijk tot een stabiele zichzelf in stand houdende populatie. Hieruit kan de conclusie worden getrokken dat het ontwikkelde concept, het stapsgewijze opbouwen van een netwerk met zowel input, hidden en output neuronen, een positief effect heeft op de leersnelheid.

In wereld 3 was het INN niet significant sneller dan het SNN, maar in wereld 1, 2 en 4 wel. Daar kunnen verschillende redenen voor zijn. Een mogelijkheid is het verschil in toegevoegde nodes aan het INN. In wereld 1, 2 en 4 worden zowel hidden nodes als input of output nodes toegevoegd aan het neurale netwerk. In wereld 3 worden echter alleen hidden nodes toegevoegd en geen input of output nodes. Het is mogelijk dat het toevoegen van input en output nodes aan een netwerk een sterker positief effect heeft op de leersnelheid van het netwerk dan de toevoeging van hidden nodes. Toegevoegde input nodes geven het netwerk namelijk toegang tot nieuwe informatie, en de toegepaste fitness functies belonen gebruik van deze informatie. Toegevoegde output nodes geven het netwerk toegang tot nieuwe acties, en de toegepaste fitness functies belonen gebruik van deze acties. Hierdoor wordt de leersnelheid gestimuleerd. Daarentegen geeft de toevoeging van hidden nodes het netwerk geen toegang tot nieuwe informatie of acties. Wel geeft het ruimte aan het netwerk om informatie op een nieuwe manier te verwerken en op basis van deze informatie op een nieuwe manier acties te kiezen. Mogelijk wordt de leersnelheid hierdoor minder gestimuleerd dan door de toevoeging van input of output nodes.

Het niet significante resultaat kan echter ook toeval zijn. Zoals eerder is uitgelegd in hoofdstuk 3 doorloopt in dit onderzoek elk type netwerk elke wereld drie keer. Eén van de resultaten van het INN voor wereld 3 wijkt zeer sterk af van de andere twee: het is twee maal zo hoog. Dit kan een uitschieter zijn dat, gecombineerd met het lage aantal datapunten, het resultaat niet significant maakt. Als dit inderdaad een uitschieter is dan zou verder onderzoek, met meer datapunten, wel resulteren in een significant resultaat.

Men kan zich afvragen of dezelfde onderzoeksresultaten zullen worden verkregen als de zoekruimte groter wordt gemaakt (creatie van een wereld 6, 7, etc.). Bij elke vergroting van de zoekruimte in dit onderzoek (wereld 1 naar wereld 2, etc.) bleek echter dat het INN beter presteerde dan het SNN. Daarom mag aangenomen worden dat het INN ook in een meer dynamische omgeving sneller blijft leren dan een SNN. Ook in een niet-dynamische omgeving zal een INN met grote waarschijnlijkheid goed presteren, maar alleen in situaties waarin de taak waarop het INN wordt toegepast kan worden opgedeeld in deeltaken, want zonder deeltaken is een INN namelijk gelijk aan een Feedforward Neuraal Netwerk. Op basis hiervan kan worden geconcludeerd dat de resultaten van het onderzoek goed generaliseerbaar zijn.

Ook zou het soort deeltaken effect kunnen hebben op de leersnelheid van het INN. In dit onderzoek zijn de deeltaken namelijk sequentieel: succes in de ene deeltaak is nodig om de volgende deeltaak succesvol te uit te voeren. De deeltaken zijn dus afhankelijk van elkaar. Het ligt echter voor de hand dat ook in een situatie waarin de deeltaken onafhankelijk van elkaar zijn, het INN sneller zal zijn dan het SNN, omdat het voor alle deeltaken - behalve de laatste - uit een kleiner netwerk bestaat en daarom minder waarden bevat die moeten worden getraind. Dit is echter niet duidelijk in het geval de deeltaken tegenovergesteld zijn. Als de eerste deeltaak bijvoorbeeld naar links lopen is en de tweede naar rechts lopen, dan zou het INN, door de lage mutatiekans van het deel van zijn netwerk dat de eerste deeltaak heeft geleerd, moeite kunnen hebben om zijn gedrag aan te passen. Dit vormt dus een punt voor verder onderzoek.

Het INN is niet alleen sneller dan het SNN, maar heeft bovendien een aantal eigenschappen waardoor het beter bruikbaar is voor bepaalde taken dan het SNN. Zo kan een complexe taak worden opgesplitst in minder complexe deeltaken, met simpelere fitness functies, die één voor één kunnen worden geleerd. Ook kunnen er altijd nieuwe informatie-elementen (input neuronen) en nieuwe resultaat categorieën (output neuronen) aan het netwerk worden toegevoegd. Hiervoor moet het netwerk wel opnieuw trainen, maar het grootste gedeelte van het netwerk hoeft niet of nauwelijks te veranderen. Bovendien biedt het INN de mogelijkheid om nieuwe trainingsdata (hidden neuronen) toe te voegen zonder eerder geleerde functionaliteit van het netwerk te verliezen. Hierdoor kan een systeem deel voor deel worden opgebouwd.

In dit aspect is het INN natuurgetrouw. Dit kan onder andere worden geconcludeerd uit een vergelijking met onderzoek in [Elman] naar taalacquisitie door kinderen. In dat onderzoek

wordt een recurrent neurale netwerk gelimiteerd in zijn geheugencapaciteit. Wanneer het netwerk dan wordt getraind op zinnen uit een verzonden taal, kan het door dit 'slechte' geheugen slechts basis elementen uit die taal leren (bv. werkwoord, zelfstandig naamwoord, etc). Dit geheugen wordt daarna vergroot en het netwerk opnieuw getraind. Door de verhoogde geheugencapaciteit kan het netwerk nu complexere constructies in de taal leren die voortbouwen op de eerder geleerde elementen. Het netwerk leert de taal dus stap voor stap.

Kinderen leren taal ook stap voor stap. Ze hebben toegang tot taal via hun omgeving, maar hun hersenen zijn nog niet volgroeid. [Elman] laat zien dat deze beperking de taalacquisitie niet in de weg zit, maar juist helpt. Door de onvolgroeide hersenen worden ze via natuurlijke wijze gelimiteerd tot het leren van basis elementen uit onze taal. En door de ontwikkeling van hun hersenen neemt hun capaciteit om complexere taal elementen te leren toe.

Het INN kan hiermee worden vergeleken. Het netwerk begint namelijk klein, ofwel heeft een beperkte geheugencapaciteit, en leert simpele dingen. Het bouwt hierop voort wanneer het netwerk wordt uitgebreid. Deze uitbreiding geeft het netwerk namelijk een grotere geheugencapaciteit. De uitbreiding van geheugencapaciteit in het INN is zelfs een meer natuurgetrouw model van de hersenen dan dat in [Elman]. In de gebruikte recurrent neurale netwerken wordt het geheugen namelijk op artificiële wijze verstoord, terwijl in het INN neuronen en verbindingen worden toegevoegd, zoals ook in de hersenen optreedt.

Deze vergelijking kan nog verder worden doorgetrokken. Hersenen bestaan namelijk niet uit één geheel, maar uit verschillende delen. Sommige delen zijn ook terug te vinden in andere diersoorten en hebben gelijkende functies. Zo zijn de hersenen opgebouwd uit oudere delen (langer geleden geëvolueerd) omringd door nieuwere delen. De oudere delen houden zich bezig met simpele taken, die door nieuwere delen worden gebruikt voor complexere taken. In de evolutie zijn de hersenen dus opgebouwd door nieuwe delen toe te voegen die gebruik maken van de al bestaande delen, net als in het INN.

Zoals eerder beschreven is het INN een aangepast Feedforward Neuraal Netwerk (zie hoofdstuk 2.4). Het is niet duidelijk of een soortgelijke aanpassing van andere typen netwerken tot vergelijkbare positieve resultaten als in dit onderzoek zullen leiden. Bij een Kohonen netwerk bijvoorbeeld zijn de posities van neuronen relatief tot elkaar van belang, waardoor voor het toevoegen van neuronen en verbindingen een algoritme zou moeten worden ontworpen dat hiermee rekening houdt. Bij andere typen netwerken kunnen andere problemen voorkomen die specifieke aanpassingen vereisen. Nader onderzoek is dus nodig om te achterhalen of een aanpassing als in dit onderzoek uitgevoerd is bij het INN ook goed toe te passen is op andere typen netwerken en dan ook tot vergelijkbare positieve eigenschappen zal leiden.

Mogelijkheden voor verder onderzoek worden geboden doordat, naast de hiervoor beschreven onderzoeksresultaten, het onderzoek ook heeft geresulteerd in een nieuwe onderzoeksomgeving, namelijk de in dit rapport beschreven onderzoeksopzet en de broncode daarvan. Dit biedt anderen niet alleen de mogelijkheid om het onderzoek te herhalen, maar ook om het te gebruiken voor verder onderzoek. De broncode is echter wel in de niet zo algemeen toegepaste programmeertaal D geschreven. Omdat D is gebaseerd op de veel gebruikte programmeertaal C++ is de broncode goed te lezen, aan te passen en uit te breiden door iemand die C++ kent. De onderzoeksopzet kan bijvoorbeeld worden uitgebreid door nieuwe typen neurale netwerken toe te voegen.

De ontworpen onderzoeksopzet is echter niet alleen bruikbaar voor het onderzoeken van neurale netwerken, maar biedt ook mogelijkheden voor ander onderzoek, zoals het vergelijken van leermethoden en het bestuderen van het gedrag van creatures. Zo biedt de onderzoeksopzet de mogelijkheid om twee populaties in dezelfde omgeving met elkaar te laten concurreren. Dat zou bijvoorbeeld kunnen door verschillende populaties van creatures te ontwerpen, die niet onderling kunnen 'maten', maar elkaar wel kunnen aanvallen met behulp van een 'attack' actie. Deze populaties van creatures worden samen in een wereld geplaatst, waar ze dan dus in competitie zijn voor voedsel. De ene soort creatures zou beter kunnen zijn in het vinden van voedsel dan de andere soort, waardoor de populatie aantallen gaan verschillen, en een van beide soorten kan uitsterven. Ook kan, door het toepassen van de 'attack' actie, een interessante roofdier-prooi dynamiek ontstaan. Door de verschillende soorten creatures verschillende typen netwerken te laten gebruiken kan ook het verschil in eigenschappen tussen deze netwerken in het onderzoek worden betrokken.

Tot slot kan worden opgemerkt dat er ook economische redenen zijn om het INN toe te passen. Het ontwerpen en trainen van neurale netwerken kost veel tijd en daardoor geld. Wanneer nieuwe data beschikbaar komt, kan deze niet zomaar aan een neurale netwerk worden toegevoegd. Het netwerk moet eerst worden getraind op alle beschikbare data, inclusief de nieuwe data. Als er nieuwe data elementen worden toegevoegd kan hetzelfde netwerk zelfs helemaal niet worden gebruikt; er moet een nieuwe worden ontworpen. Ook het toevoegen van mogelijke uitkomsten van het netwerk geeft dit type problemen. Door middel van het INN kunnen bestaande netwerken echter wel worden gebruikt in deze situaties; er kunnen namelijk nieuwe delen aan de bestaande netwerken worden toegevoegd. Bij het INN hoeft geen geheel nieuw netwerk te worden ontworpen en getraind, maar kan het bestaande netwerk worden uitgebreid en vervolgens worden getraind. Dit scheelt veel tijd en geld.

REFERENTIES

- [Alpaydin]** Alpaydin, E. *GAL: Networks that grow when they learn and shrink when they forget*. In Technical Report TR-91-032, ICSI, Berkeley, CA. (1991)
- [Aran]** Aran, O. & Alpaydin, E. *An Incremental Neural Network Construction Algorithm for Training Multilayer Perceptrons*. (2003)
- [Ash]** Ash, T. Dynamic node creation in backpropagation networks. In *Connection Science*, Vol. 1, No. 4, pp. 365–375. (1989)
- [Belew]** Belew, R. K., McInerney, J. & Schraudolph, N. N. *Evolving Networks: Using the Genetic Algorithm with Connectionist Learning*. (1990)
- [Bullinaria]** Bullinaria, J. A. *Evolving Neural Networks: Is it Really Worth the Effort?* In ESANN'2005 Proceedings of the European Symposium on Artificial Neural Networks, pp. 267-272. (2005)
- [Dorigo]** Dorigo, M., Maniezzo, V. & Coloni, A. *The Ant System: Optimization by a Colony of Cooperating Agents*. In *IEEE Transactions on Systems, Man and Cybernetics - Part B*, Vol. 26, No. 1, pp. 1-13. (1996)
- [Elman]** Elman, J. L. *Learning and Development in Neural Networks: The Importance of Starting Small*. In *Cognition*, Vol. 48, pp. 71-99. (1993)
- [Fahlman]** Fahlman, S. E. & Lebiere, C. *The Cascade-Correlation Learning Architecture*. (1991)
- [Gardner]** Gardner, M. *Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game "Life"*. In *Scientific American*, Vol. 223, pp. 120-123. (1970)
- [Gazzaniga]** Gazzaniga, M. S., Ivry, R. B. & Mangun, G. R. *Cognitive Neuroscience: The Biology of the Mind, Second Edition*. (2002)
- [Geman]** Geman, S., Bienenstock, E. & Doursat, R. *Neural networks and the bias/variance dilemma*. In *Neural Computation*, Vol. 4, pp. 1–58. (1992)

- [Gerstner]** Gerstner, W. & Kistler, W. M. *Hebbian Models*. In Gerstner, W. & Kistler, W. M., *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Ch. 10. (2002)
- [Hodgkin]** Hodgkin, A. & Huxley, A. *A quantitative description of membrane current and its application to conduction and excitation in nerve*. In *The Journal of Physiology*, Vol. 117, pp. 500–544. (1952)
- [Izhikevich]** Izhikevich, E. M. *Simple Model of Spiking Neurons*. In *IEEE Transactions on Neural Networks*, Vol. 14, No. 6, pp. 1569-1572. (2003)
- [Kasiński]** Kasiński, A. & Ponulak, F. *Comparison of Supervised Learning Methods For Spike Time Coding in Spiking Neural Networks*. In *International Journal of Applied Math and Computer Science*, Vol. 16, No. 1, pp. 101-113. (2006)
- [Kohonen]** Kohonen, T. *Self-Organizing Maps. Third, extended edition*. (2001)
- [LeCun]** LeCun, Y., Bottou, L., Orr, G. B. & Müller, K. *Efficient BackProp*. In Orr, G. B. & Müller, K. *Neural Networks: tricks of the trade*. (1998)
- [Montana]** Montana, D. J. & Davis, L. *Training Feedforward Neural Networks Using Genetic Algorithms*. In Kaufmann, M. *Proceedings of 11th International Joint Conference of Artificial Intelligence*, pp. 762-767. (1989)
- [Nabhan]** Nabhan, T. & Zomaya, A. *Toward generating neural network structures for function approximation*. In *Neural Networks*, Vol. 7, No. 1, pp. 89–99. (1994)
- [Nolfi]** Nolfi, S., Elman, J. L. & Parisi, D. *Learning and Evolution in Neural Networks*. (1994)
- [Reynolds]** Reynolds, C. W. *Flocks, Herds and Schools: A distributed behavioral model*. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25-34. (1987)
- [Rojas]** Rojas, R. *Weighted networks - the perceptron*. In Rojas, R. *Neural Networks - A Systematic Introduction*, Ch. 3. (1996)

[Rumelhart] Rumelhart, D. E. & Zipser, D. *Feature Discovery by Competitive Learning*. In *Cognitive Science: A Multidisciplinary Journal*, Vol. 9, No. 1, pp. 75-112. (1985)

[Russell] Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach, Second Edition*. (2003)

[Setiono] Setiono, R. *Feedforward neural network construction using cross validation*. In *Neural Computation*, Vol. 13, pp. 2865–2877. (2001)

[Vreeken] Vreeken, J. *Spiking Neural Networks, an introduction*. (2003)

[Yaeger 1] Yaeger, L. S. *Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context*. In Langton, C. ed. *Proceedings of the Artificial Life III Conference*, pp. 263-298. (1994)

[Yaeger 2] Griffith, V. & Yaeger, L.S. *Ideal Free Distribution in Agents with Evolved Neural Architectures*. In Rocha, L. et al. eds. *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pp. 372-378. (2006)

[Yao 1] Yao, X. & Liu, Y. *Evolving Artificial Neural Networks through Evolutionary Programming*. (1996)

[Yao 2] Yao, X. & Liu, Y. *A New Evolutionary System for Evolving Artificial Neural Networks*. In *IEEE Transactions on Neural Networks*, Vol. 8, No. 3, pp. 694-713. (1997)