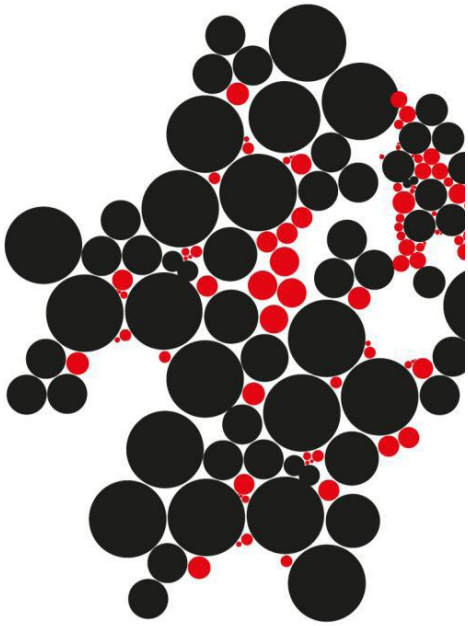


BACHELORTHESE

CREËREN VAN CLUSTERS BINNEN BOEKENDATABASES MET BEHULP VAN EEN NEURAAL NETWERK

Janina Torbecke
GEDRAGSWETENSCHAPPEN
HUMAN FACTORS & MEDIA PSYCHOLOGY (HFM)

AFSTUDEERCOMMISSIE
Prof. dr. F. van der Velde (1e begeleider)
Dr. M. L. Noordzij (2e begeleider)

11-01-2013

UNIVERSITEIT TWENTE.

Samenvatting

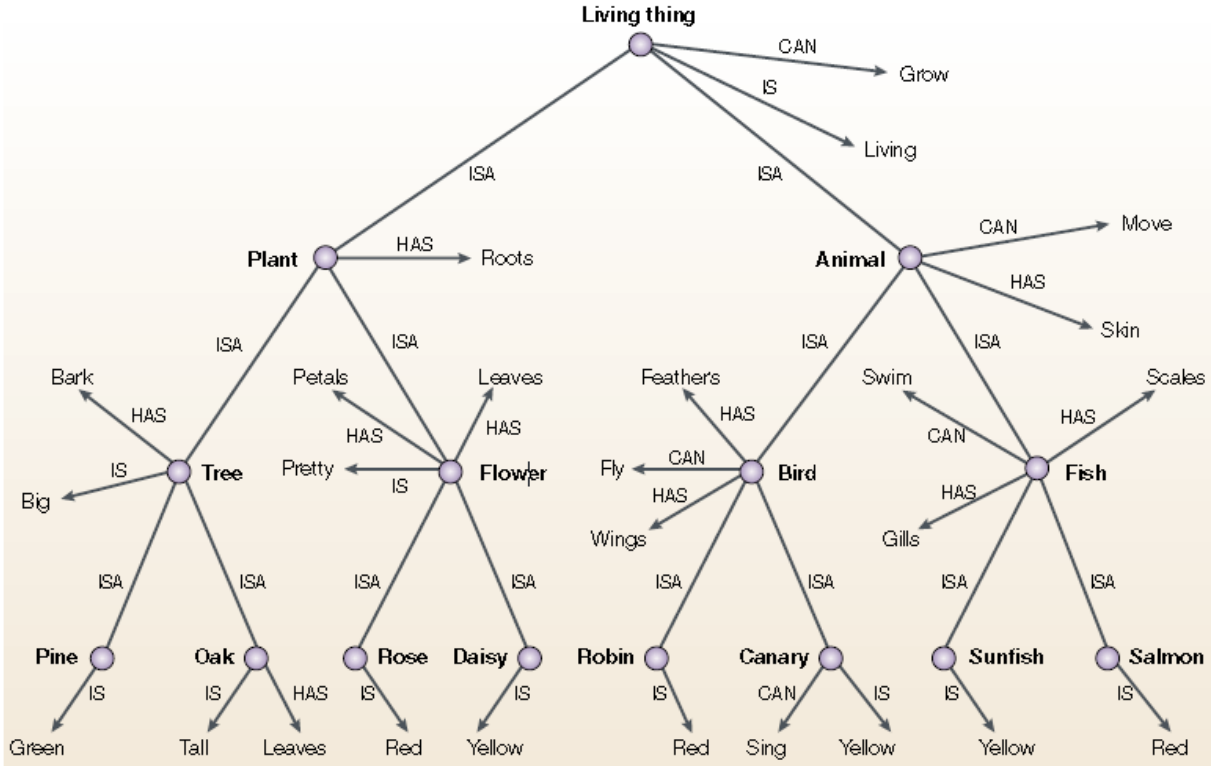
In dit onderzoek werd gekeken naar een mogelijkheid om het verkopers in boeken-, spellen- en dvd-winkels gemakkelijker te maken om hun klanten te adviseren over nieuwe artikelen. Omdat hierbij de persoonlijke voorkeur van klanten een grote rol speelt, kan een advies niet gebaseerd worden op de kwaliteit van de artikelen zoals dat bij veel andere producten het geval is. Op basis van een kunstmatig neuraal netwerk dat ontwikkeld werd door Rumelhart (1986) werden drie netwerken geconstrueerd die getraind werden om eigenschappen van boeken te leren. Door in de trainingsfase alle eigenschappen te relateren aan een bepaalde context in de vorm van inputneuronen in de netwerken, konden de netwerken bepaalde eigenschappen clusteren. Uit het onderzoek bleek dat de netwerken ook in staat zijn om eigenschappen met verschillende contexten met elkaar in verband te brengen. Bijvoorbeeld konden bepaalde auteurs gerelateerd worden aan bepaalde onderwerpen waarover zij vaak boeken hebben geschreven. Uit de analyse van de netwerken bleek dat zij al redelijk goed in staat zijn om zulke verbanden te leggen. Echter zal verder onderzoek gedaan moeten worden en zullen de netwerken verbeterd moeten worden voordat zij in de praktijk gebruikt kunnen worden. Door het bijhouden van eigenschappen van boeken in een dergelijk neuraal netwerk zou het voor verkopers in de toekomst mogelijk kunnen zijn om snel te weten te komen welke boeken (of andere producten) in zijn winkel lijken op een door de klant gewaardeerd boek.

Inhoudsopgave

Inleiding

Winkels hebben er veel baat bij als zij hun klanten optimaal kunnen adviseren en hen daardoor tevreden kunnen stellen. Dit speelt vooral in boeken-, spellen- of dvd-winkels een grote rol. Het advies van de verkoper kan in deze winkels minder gebaseerd worden op de kwaliteit van het artikel en is sterker afhankelijk van de persoonlijke voorkeur van de klanten. Om de toch een advies te kunnen geven kunnen de artikelen gegroepeerd worden op basis van verschillende kenmerken. Bijvoorbeeld kan kunnen boeken die geschreven zijn door dezelfde auteur een groep vormen. Het gewenste resultaat van deze groepering is dat de artikelen binnen een groep met betrekking tot minstens een kenmerk op elkaar lijken. Met andere woorden: De artikelen worden toegekend aan verschillende categorieën. Als bekend is dat een klant van boeken van een bepaalde auteur houdt dan kan door het maken van categorieën worden gekeken welke boeken die auteur verder heeft geschreven. Veel interessanter is het echter om te kijken naar meer dan een categorie. Schrijft de auteur bijvoorbeeld vaak boeken over hetzelfde onderwerp? Vind de klant misschien vooral dat onderwerp interessant en leest hij daarom vaan boeken van dezelfde auteur? Om hier achter te komen moeten de artikelen in een winkel op meer kenmerken dan alleen de auteur of regisseur worden gegroepeerd.

In het verleden zijn er verschillende theorieën over menselijke semantische cognitie ontwikkeld. Collins en Quillian (1969) stelden dat concepten hiërarchisch worden geordend en dat gebruik wordt gemaakt van proposities. Zij ontwikkelden het *hierarchical propositional model*. De concepten zijn in hun model op een zodanige manier geordend dat de categorieën waaronder zij vallen van boven naar beneden steeds specifieker worden. In figuur 1 is een voorbeeld van hun model te zien met als onderwerp levende 'dingen' (*'living things'*). Iedere lijn representeert een propositie en alle punten representeren categorieën zoals plant, dier, boom of vogel (respectievelijk *plant*, *animal*, *tree* and *bird*). De pijlen wijzen naar bepaalde concepten zoals groot (*big*), bewegen (*move*) of groeien (*grow*). Volgens Collins en Quillian zijn proposities in het model



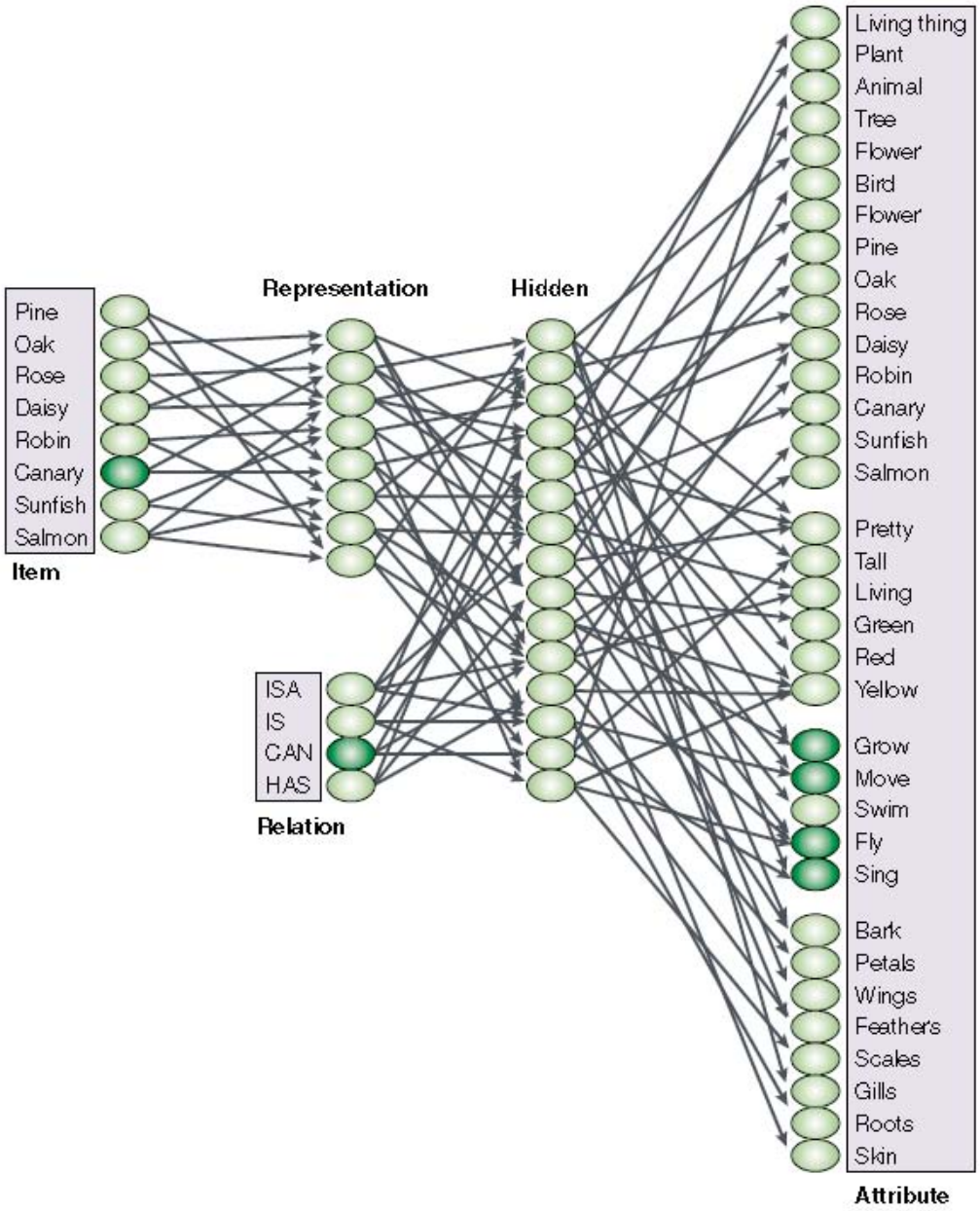
Figuur 1 **Het *hierarchical propositional model* van Collins en Quillian (1969)**

altijd geldig voor alle concepten die lager staan in de hiërarchie. Zo kunnen alle levende dingen groeien (*grow*) en alle vogels kunnen vliegen (*fly*). Om te bepalen of een propositie geldt voor een bepaald concept, kan daarom eerst worden gekeken of die propositie opgeslagen is bij dat concept. Bijvoorbeeld kan gekeken worden naar de propositie 'vogel kan bewegen' (*bird can move*). De propositie is in de figuur niet opgeslagen bij het concept 'vogel' (*bird*). De volgende stap is daarom om hoger in de hiërarchie te zoeken naar de propositie 'kan bewegen'. In het voorbeeld wordt de propositie bij het concept 'dier' (*animal*) gevonden. Omdat het concept 'vogel' lager in de hiërarchie staat dan 'dier', geldt dat alle vogels kunnen bewegen. Over het algemeen geldt voor dit model dat specifieke eigenschappen dicht bij een concept opgeslagen zijn (bijv. 'vogel kan vliegen') en algemenere eigenschappen verder van het concept vandaan liggen (bijv. 'vogel is levend' – *'bird is living'*).

Dit feit leidt tot twijfels aan het model. Als het model klopt dan zouden mensen specifieke eigenschappen sneller aan een concept moeten kunnen toekennen dan algemene eigenschappen, aangezien de afstand tussen het concept en algemene eigenschappen groter is.

Een dergelijk verschil kon echter niet worden aangetoond in onderzoek (McCloskey, Glucksberg, 1979; Murphy, Brownell, 1985).

Een model dat semantische cognitie beter benadert is het door Rumelhart (1986) ontwikkelde *parallel distributed processing model* (*PDP*). Het model van Rumelhart is niet hiërarchisch opgebouwd zoals het model van Collins en Quillian, maar het bevat neuronen en connecties tussen deze neuronen. De neuronen worden afhankelijk van een activatie functie (*activation function*) geactiveerd en de connecties hebben een bepaald gewicht dat de activatie van neuronen



Figuur 2 **Het *parallel distributed processing model (PDP)* van Rumelhart (1986)**

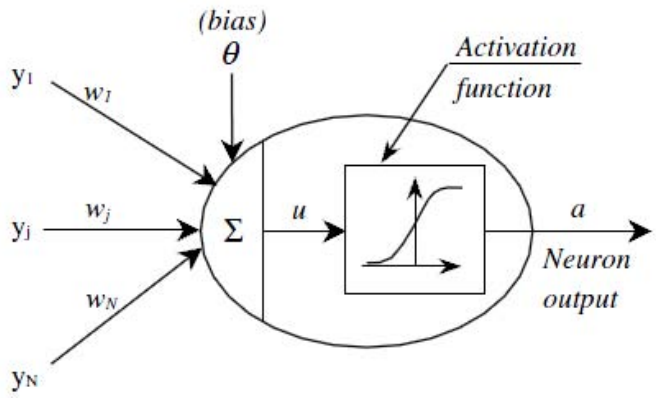
beïnvloedt. Door het veranderen van het gewicht van connecties tussen neuronen kunnen verbanden worden gelegd tussen concepten en eigenschappen. Het model bestaat uit input-, hidden en outputneuronen (figuur 2). De inputneuronen representeren verschillende concepten (bijv. 'vogel' of 'plant') en relaties (bijv. 'is' of 'heeft'). Deze inputneuronen staan in de figuur aan de linkerkant (groepen '*relation*' en '*item*'). De outputneuronen (groep '*attribute*' in de figuur) staan aan de rechterkant. Alle andere neuronen zijn hidden neuronen. De connecties tussen de neuronen lopen van links naar rechts. De neuronen binnen dezelfde groep zijn niet met elkaar verbonden.

Zoals boven al genoemd kan het gewicht van connecties tussen neuronen veranderen. Dit gebeurt door het netwerk te trainen met een aantal voorbeelden van juiste combinaties van input- en outputneuronen. Bijvoorbeeld kan getraind worden dat de activatie van de inputneuronen 'kanarie' en 'kan' moet leiden tot een activatie van de outputneuronen 'groeien', 'bewegen', 'vliegen' en 'zingen'. In de figuur wordt dit voorbeeld aangetoond door de donkergroene cirkels ('*canary*', '*can*', '*grow*', '*move*', '*fly*', *'sing*'). Tijdens het trainingsproces ontstaat in de lagen met hidden neuronen een patroon (combinatie van gewichten en activaties van connecties en neuronen) dat het mogelijk maakt om tot de gewenste output te leiden. De gewichten worden veranderd tot het moment dat het netwerk alle voorbeelden van input/output-combinaties heeft geleerd. Het leerproces wordt gerealiseerd door herhaaldelijk de activatie van de hidden neuronen en de gewichten van de connecties te berekenen en aan te passen. Ieder neuron krijgt een bepaalde input van andere neuronen of – in het geval van de inputneuronen – een externe input. Deze input is afhankelijk van de activatie van de andere neuronen en het gewicht van de connecties tussen deze neuronen en het neuron waarvoor de activatie berekend moet worden.

De output van een neuron N wordt met behulp van de volgende formule berekend:

$$a = f\left(\sum_{i=0}^{n} w_i y_i\right)$$

Hierbij is a de output ofwel *activation* van het neuron N. $f()$ is een bepaalde activatie functie die gevarieerd kan worden. Een voorbeeld is $f(x) = x$. Dit zou betekenen dat de output $f(x)$ van N gelijk is aan zijn input x. x, de input, is de som van de activaties y_j van alle inputneuronen van N vermenigvuldigd met het gewicht w_i van hun connecties met het neuron N. Figuur 3 laat dit proces grafisch zien. y_i t/m y_N zijn de inputneuronen die verbonden zijn met neuron N. w_i t/m w_N zijn de gewichten van de connecties tussen de inputneuronen en neuron N. Vervolgens wordt de som van alle inputneuronen, vermenigvuldigd met het gewicht van hun connectie met N berekend: $\sum_{i=0}^{n} w_i y_i$. Ten slotte wordt door de activatie functie de output bepaald. In de figuur is bovendien een *bias* afgebeeld. Daarop zal hier niet in worden gegaan.



Figuur 3 **Activatie van een neuron (Sakla, & Ashour, 2005)**

Nadat deze berekening voor alle neuronen minstens een keer werd uitgevoerd hebben ook de outputneuronen bepaalde activatiewaarden. Op basis van het verschil tussen de activatie van een outputneuron en zijn 'doelactivatie', de activatie die door de voorbeelden in de trainingsfase vastgelegd is, wordt het gewicht van de connecties en/of de activatie van hidden neuronen veranderd die verbonden zijn met dat outputneuron. Deze methode is bedoeld om het verschil tussen de feitelijke activatie en de doelactievatie, de error, te verkleinen. Na een succesvolle training is de error gelijk aan nul. Er is dan geen verschil meer tussen de activatie van outputneuronen en het gewenste resultaat.

Het berekenen van activaties van links naar rechts – van input- over hidden naar outputneuronen wordt '*forward propagation of activation*' en het aanpassen van gewichten en activaties op basis van de error wordt '*backpropagation of error*' genoemd. Deze twee stappen worden herhaald totdat de error gelijk is aan nul en daarom geen connectiegewichten of activaties meer veranderd hoeven te worden.

Rumelhart liet zien dat een getraind netwerk geleerde samenhangen ook kan toepassen op nieuwe concepten. Hij voegde aan een getraind netwerk een extra inputneuron toe dat een 'mus' ('*sparrow*') representeerde en dat verbonden was met de hidden neuronen die in figuur 2 in tot de groep '*representation'* behoren. Vervolgens trainde hij het netwerk met het voorbeeld 'mus–ISEEN– vogel/dier/levend ding' ('*sparrow*–ISA–bird/animal/living thing'). Tijdens de trainingsfase mochten in dit geval echter alleen de nieuwe connecties worden veranderd die het nieuwe neuron met de *representation*-neuronen verbonden. Toen het netwerk opnieuw getraind was ging Rumelhart hem testen door voor het netwerk onbekende combinaties van inputneuronen te activeren. Hij testte de combinaties 'mus–KAN' ('*sparrow–CAN*'), 'mus–HEEFT' ('*sparrow–HAS*') en 'mus–IS' ('*sparrow–IS*'). Er bleek dat in elk geval outputneuronen geactiveerd werden die overeenkomen met eigenschappen van andere vogels.

In dit onderzoek wordt gekeken of het netwerk van Rumelhart toegepast kan worden op boeken. Er wordt getest of een dergelijk neuraal netwerk in staat is een aantal eigenschappen en kenmerken van boeken te leren. Vervolgens wordt onderzocht welke mogelijkheden er door het toevoegen van een extra neuron en het herhaalde trainen van het netwerk ontstaan. Het extra neuron representeert in dit onderzoek een fictief boek. De hypothese is dat als het netwerk leert dat het boek geschreven is door bijvoorbeeld auteur A, dat er dan automatisch verbanden gelegd worden met bijvoorbeeld een bepaald onderwerp of categorie. In het voorbeeld van Rumelhart met de mus ontstond er automatisch een verband tussen het mus-neuron en onder andere vliegen, omdat het netwerk geleerd heeft dat een mus een vogel is en omdat vogels kunnen vliegen. Met betrekking tot boeken is dit verband minder triviaal. Een auteur hoeft niet altijd over

hetzelfde onderwerp te schrijven en prijzen worden niet altijd gewonnen voor boeken van dezelfde uitgever. Doel van dit onderzoek is om te kijken of er desondanks verbanden tussen het nieuwe neuron en eigenschappen ontstaan die niet getraind worden.

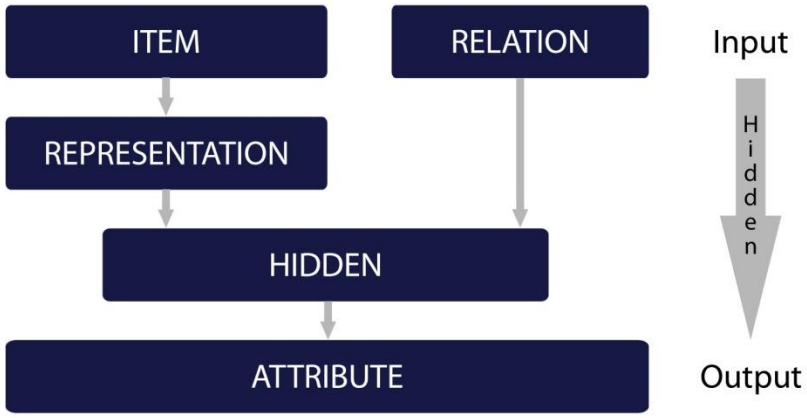
Als dergelijke verbanden door het netwerk kunnen worden gelegd, zou dat nieuwe mogelijkheden kunnen bieden voor verkopers van boeken, spellen of dvd's. Als bekend is dat een klant vaak boeken van auteur A leest zou een kunstmatig neuraal netwerk gekoppeld kunnen worden aan de database van de verkoper en met behulp van dat netwerk gekeken kunnen worden of die auteur in verband kan worden gebracht met andere eigenschappen van boeken zoals het onderwerp of gewonnen prijzen. Op basis van de uitkomsten zouden vervolgens boeken aanbevolen kunnen worden die aan dezelfde criteria voldoen als boeken van auteur A.

Methoden

Voor dit onderzoek werd gebruik gemaakt van het programma MemBrain waarmee neurale netwerken gesimuleerd kunnen worden. Met behulp van MemBrain werden gedurende het onderzoek drie verschillende neurale netwerken gemaakt die gebaseerd zijn op het neurale netwerk van Rumelhart (Rumelhart 1990; Rumelhart & Todd 1993). Net zo als in het model van Rumelhart bestaat elk netwerk uit meerdere lagen:

- *item*-laag met input neuronen
- *relation*-laag met input neuronen
- *representation*-laag met hidden neuronen
- extra laag met hidden neuronen
- *attribute*-laag met output neuronen

In tegenstelling tot het model van Rumelhart lopen de connecties in de drie gemodelleerde netwerken van boven naar onderen in plaats van links naar rechts. De inputneuronen bevinden zich boven in het netwerk en de outputneuronen beneden. Een schematische weergave van de netwerken is te zien in figuur 4. De reden voor de andere opbouw



Figuur 4 **Schematische weergave van de gemodelleerde netwerken**

van de netwerken vergeleken met het netwerk van Rumelhart is de representatie van neuronen in MemBrain. De inputlinks zijn in het programma altijd aan de bovenkant van een neuron en de

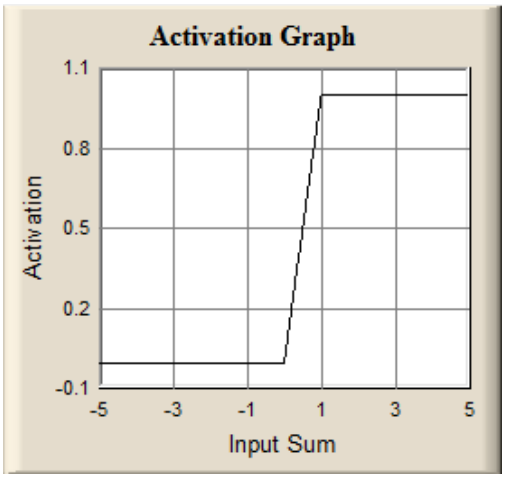
outputlinks aan de onderkant. Om de netwerken zo overzichtelijk mogelijk te houden werd daarom voor dit design gekozen.

Aangezien de netwerken in dit onderzoek toegepast werden op boeken, representeren de input neuronen in de *item*-laag bepaalde boektitels. De inputneuronen in de *relation*-laag geven de context van de outputneuronen in de *attribute*-laag aan met betrekking tot de input neuronen in de *item*-laag. Een voorbeeld hiervan is de context 'geschreven door' die de relatie tussen een auteur en een boek beschrijft. In het vervolg zullen de neuronen in de *item*-, *relation*-, *representation*- en *attribute*-laag respectievelijk item-, context-, representatie- en attribuutneuronen worden genoemd. Later zal per neuraal netwerk worden uitgelegd welke input en output neuronen er precies zijn.

Als activatie functie voor alle neuronen in het netwerk werd '*identical 0 to 1*' gebruikt. Deze functie ziet er als volgt uit:

$$f(x) = \begin{cases} 0, & x < A_{min} \\ x, & A_{min} \leq x \leq A_{max} \\ 1, & x > A_{max} \end{cases}$$

Omdat de activatie A van een neuron bij de activatie functie '*identical 0 to 1*' per definitie minimaal 0 is en maximaal 1, is $A_{min} = 0$ en $A_{max} = 1$. In figuur 5 is deze functie grafisch te zien.



Figuur 5 **Curve van de *identical (0 to 1)* activatie functie**

De drie in MemBrain gemodelleerde netwerken verschillen in hun complexiteit. Het eerste netwerk bevat alleen basisinformatie over boeken om te kunnen testen of het netwerk werkt zoals gewenst. De twee andere netwerken zijn complexer dan het eerste. Dit betekent dat zij vergeleken met het eerste netwerk meer neuronen en connecties bevatten. Er werd onderzocht of het netwerk ook een grote

hoeveelheid boeken en hun eigenschappen kan leren en deze kan koppelen aan de contextneuronen.

Om de verschillende netwerken te testen werd gebruik gemaakt van fictieve boeken. Dit betekent dat de netwerken niet met eigenschappen van bestaande boeken werden getraind, maar met eigenschappen van fictieve boeken. Het voordeel hiervan is dat het onderzoek daardoor op een heel structurele en gecontroleerde manier uitgevoerd kon worden. De gegevens van alle boeken konden zo veranderd worden als het op dat moment nodig was. Er is bewust voor gekozen om de boeken, auteurs, enzovoort geen namen te geven om te voorkomen dat bij de analyse op een subjectieve manier verbanden gelegd worden tussen de boektitels en de verschillende outputneuronen. In tabel 1 staan alle vijf boeken met bijbehorende eigenschappen die voor netwerk A werden gebruikt. Hier is te zien dat dit netwerk alleen de boektitels, de auteurs en de onderwerpen van de boeken bevat. De eerste twee boeken (titel1 en titel2) werden geschreven door auteur1 en gaan over onderwerp1. De overige drie boeken werden geschreven door auteur2. Titel3, het derde boek, gaat over onderwerp1 en titel4 en titel5 zijn geschreven over onderwerp2.

Boektitel	**Auteur**	**Onderwerp**
Titel1	Auteur1	Onderwerp1
Titel2	Auteur1	Onderwerp1
Titel3	Auteur2	Onderwerp1
Titel4	Auteur2	Onderwerp2
Titel5	Auteur2	Onderwerp2

Tabel 1 **Boeken netwerk A**

Het tweede netwerk, netwerk B, werd naast boektitel, auteur en onderwerp ook getraind met betrekking tot de uitgever van een boek en de door een boek gewonnen prijzen. Tabel 2 laat tien verschillende boeken zien, waarbij de eerste vier geschreven zijn door auteur1, de daaropvolgende drie boeken door auteur2 en de laatste drie door auteur3. Drie van de boeken van auteur1 zijn uitgegeven door uitgever1. Het vierde boek en alle boeken van de andere auteurs zijn uitgegeven door uitgever2. Alle boeken van auteur1 gaan over onderwerp1 en auteur2 heeft alleen boeken over onderwerp2 geschreven. Auteur3 heeft boeken over beide onderwerpen

geschreven: Zijn eerste boek gaat over onderwerp1 en de andere twee boeken gaan over onderwerp2. Verder hebben de boeken 1, 2, 3, 7 en 10 prijzen gewonnen zoals te zien in de tabel.

Boektitel	Auteur	Uitgever	Onderwerp	Gewonnen prijzen
Titel1	Auteur1	Uitgever1	Onderwerp1	Prijs1
Titel2	Auteur1	Uitgever1	Onderwerp1	Prijs1, Prijs2
Titel3	Auteur1	Uitgever1	Onderwerp1	Prijs1
Titel4	Auteur1	Uitgever2	Onderwerp1	
Titel5	Auteur2	Uitgever2	Onderwerp2	
Titel6	Auteur2	Uitgever2	Onderwerp2	
Titel7	Auteur2	Uitgever2	Onderwerp2	Prijs2
Titel8	Auteur3	Uitgever2	Onderwerp1	
Titel9	Auteur3	Uitgever2	Onderwerp2	
Titel10	Auteur3	Uitgever2	Onderwerp2	Prijs3

Tabel 2 **Boeken netwerk B**

In netwerk C werden ten slotte nogmaals drie eigenschappen toegevoegd: het geslacht van de auteurs, een categorie van de boeken en een doelgroep. In tabel 3 is te zien dat de helft van de boeken geschreven is door een man en de andere helft door een vrouw. Verder valt de helft van de boeken onder de categorie vrije tijd en de andere helft onder de categorie kennis. De gebruikte doelgroepen voor de boeken zijn kinderen, tieners en volwassenen. Naast de genoemde veranderingen werd het aantal auteurs in dit netwerk verhoogd naar zes.

Boektitel	Auteur	Geslacht auteur	Uitgever	Categorie	Onderwerp	Doelgroep	Gewonnen prijzen
Titel1	Auteur1	man	Uitgever1	VrijeTijd	Onderwerp1	Kinderen	Prijs1
Titel2	Auteur1	man	Uitgever1	VrijeTijd	Onderwerp1	Kinderen	Prijs1
Titel3	Auteur2	vrouw	Uitgever2	VrijeTijd	Onderwerp2	Tieners	
Titel4	Auteur2	vrouw	Uitgever2	VrijeTijd	Onderwerp2	Tieners	
Titel5	Auteur3	man	Uitgever1	VrijeTijd	Onderwerp1	Kinderen	
Titel6	Auteur3	man	Uitgever1	Kennis	Onderwerp1	Kinderen	
Titel7	Auteur4	vrouw	Uitgever1	Kennis	Onderwerp2	Tieners	Prijs2
Titel8	Auteur4	vrouw	Uitgever2	Kennis	Onderwerp3	Volwassenen	
Titel9	Auteur5	man	Uitgever2	Kennis	Onderwerp2	Volwassenen	
Titel10	Auteur6	vrouw	Uitgever1	Kennis	Onderwerp3	Volwassenen	Prijs2

Tabel 3 **Boeken netwerk C**

Tijdens het gehele onderzoek werden in totaal zeven verschillende contextneuronen gebruikt. In tabel 4 is te zien met welke outputneuronen elk contextneuron geassocieerd wordt. Een X in de kolommen netwerk A, netwerk B en netwerk C geeft aan dat het netwerk een bepaald contextneuron bevat.

Context-neuron	Netwerk A	Netwerk B	Netwerk C	Associatie
geschrevenDoor	X	X	X	Auteur
			X	Geslacht auteur
uitgegevenDoor		X	X	Uitgever
inCategorie	X	X	X	Categorie
gaatOver	X	X	X	Onderwerp
voorLeeftijdsgroep			X	Doelgroep
heeftGewonnen		X	X	Gewonnen prijzen
is	X	X	X	Boektitel

Tabel 1 **Context-neuronen in alle netwerken**

Voor alle drie netwerken werden vervolgens zogenaamde '*lessons*' (les) aangemaakt. Door een *lesson* wordt bepaald hoe een netwerk getraind wordt en wat het netwerk in het ideale geval gaat leren. In drie csv-bestanden[1] werden de *lessons* voor de drie netwerken gemaakt door verschillende input- en bijbehorende outputpatronen aan te geven. Elk patroon bevat alle input- en outputneuronen van het netwerk, waarbij telkens andere neuronen geactiveerd ofwel gedeactiveerd worden. Een gedeactiveerd neuron wordt gerepresenteerd door een 0 en een geactiveerd neuron door een 1. Voor boek1 in netwerk A werden bijvoorbeeld drie input/outputpatronen (I/O-patronen) aangemaakt zoals te zien in tabel 5.

Input								**Output**								
Titel1	Titel2	Titel3	Titel4	Titel5	is	Geschreven Door	gaatOver	Titel1	Titel2	Titel3	Titel4	Titel5	Auteur1	Auteur2	Onderwerp1	Onderwerp2
1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Tabel 2 **Deel van de lesson van netwerk A**

Aan de linkerkant van de tabel staan alle inputneuronen van het netwerk en aan de rechterkant alle outputneuronen. Door iedere rij representeert precies een I/O-patroon dat getraind werd. Een activatie van de inputneuronen *titel1* en *is* (in de eerste rij gekenmerkt met een 1) zal na een succesvol afgeronde trainingsfase leiden tot een activatie van het

[1] Het csv-bestand van netwerk A is als tabel te vinden in de bijlage. De overige csv-bestanden kunnen verkregen worden bij de auteur.

outputneuron *titel1* en een deactivatie van alle andere outputneuronen. Op een dergelijke manier werden de eigenschappen van alle boeken in de drie netwerken ingevuld in de csv-bestanden.

Het maximale aantal I/O-patronen in de csv-bestanden is in dit onderzoek per netwerk $n_{titels} * m_{contextneuronen}$, waarbij n_{titels} het aantal boeken in het netwerk is en $m_{contextneuronen}$ het aantal gebruikte contextneuronen.

Omdat niet voor ieder boek gebruik wordt gemaakt van alle contextneuronen, is het precieze aantal I/O-patronen gelijk aan:

$$\sum_{i=1}^{n_{titels}} m_i$$

In dit geval is n_{titels} weer het aantal boeken in het netwerk en m_i is het aantal feitelijk gebruikte contextneuronen per boek. Bij een boek dat geen prijs heeft gewonnen is bijvoorbeeld geen I/O-patroon met *heeftGewonnen* nodig en zijn er minder dan $n_{titels} * m_{contextneuronen}$ I/O-patronen. Dit betekent dat altijd geldt $\sum_{i=1}^{n_{titels}} m_i < n_{titels} * m_{contextneuronen}$.

Om de netwerken te trainen werden de csv-bestanden met I/O-patronen geïmporteerd in MemBrain en uitgevoerd met een zogenaamde *teacher*, ofwel docent. Bij het uitvoeren van de *lesson* wordt het netwerk geleerd welke inputpatronen bij bepaalde outputpatronen horen.

Per neuraal netwerk werd gekeken of dat netwerk in staat was om alle aangegeven patronen te leren, dat wil zeggen of een bepaalde activatie van inputneuronen na het uitvoeren van de *lesson* altijd leidt tot een bepaalde activatie van outputneuronen zoals aangegeven in het csv-bestand. Dit is het geval als de *Mean Squared Output Error* gelijk is aan nul. Deze error kan MemBrain tijdens de trainingsfase automatisch berekenen.

Mochten de netwerken de I/O-patronen succesvol hebben geleerd, werd de *Activation Threshold*, de activatie, van alle neuronen en het gewicht van alle links tussen de neuronen

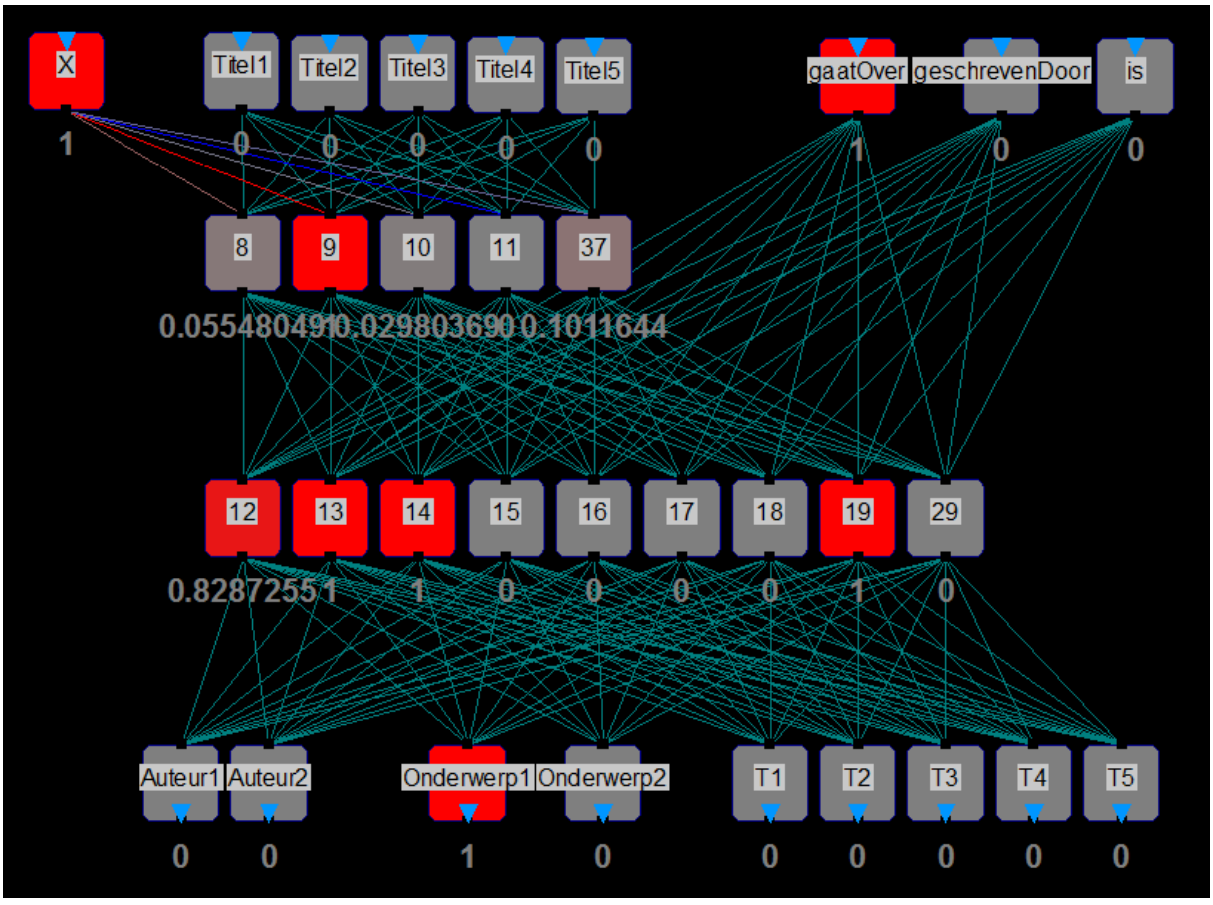
'vastgezet'. Dat houdt in dat deze waarden niet meer konden worden veranderd door de *teacher* als een nieuwe *lesson* werd uitgevoerd. Na het vastzetten van deze gegevens, werd een extra inputneuron *X* toegevoegd aan het netwerk en verbonden met alle neuronen in de representatielaag. Dit neuron kan worden beschouwd als een voor het netwerk onbekende boektitel. Vervolgens werd een nieuwe *lesson* aangemaakt met het nieuwe neuron *X*. Deze *lesson* bevat een of twee I/O-patronen waardoor geprobeerd wordt om neuron *X* te koppelen aan een bepaalde context en zodoende aan een bepaalde output. Bijvoorbeeld kunnen de neuronen *X* en *geschrevenDoor* worden aangegeven als geactiveerde inputneuronen en *Auteur1* als gewenst geactiveerd outputneuron, terwijl alle andere input- en outputneuronen gedeactiveerd blijven. Het doel hiervan is om – zoals in de inleiding beschreven – te onderzoeken of door het extra neuron verbanden ontstaan tussen het neuron *X* en andere, niet getrainde eigenschappen.

Wederom werd nagegaan of het netwerk in staat was om het nieuwe patroon te leren door alleen het gewicht van de nieuwe links tussen neuron *X* en de neuronen in de representatielaag te veranderen. Doordat alle andere links vastgezet werden, waren dit de enige gewichten die aangepast konden worden. Vervolgens werd gekeken naar de output van een voor het netwerk onbekende combinatie van inputneuronen. Als netwerk A bijvoorbeeld geleerd heeft dat *X geschreven is* door *Auteur1*, dan kan gekeken worden welke outputneuronen geactiveerd worden als de neuronen *X* en *gaatOver* als input worden geactiveerd.

Resultaten

Netwerk A

Netwerk A bleek in staat te zijn alle I/O-combinaties te leren. Dit betekent dat het netwerk na de trainingsfase alle geleerde inputpatronen zonder fouten kan 'vertalen' naar het getrainde outputpatroon. Figuur 6 laat zien hoe het netwerk er na het toevoegen van een extra inputneuron uitziet. De outputneuronen T1 t/m T5 representeren hierbij de titels 1 t/m 5. Verder is onder de neuronen hun activatie aangegeven. Naarmate de activatiewaarde hoger wordt, wordt een neuron roder. Een gedeactiveerd neuron is grijs. Is een connectie in het netwerk groen gekleurd, is zijn gewicht vastgezet voor de *teacher*. Andere kleuren van de connecties zoals rood of blauw geven het gewicht van de connecties aan.



Figuur 6 **Netwerk A na het toevoegen van neuron X**

Na het toevoegen van neuron *X* werd een aantal *lessons* uitgevoerd met verschillende I/O-patronen. In tabel 6 staan de uitkomsten van alle gebruikte combinaties van inputneuronen.

Hierbij is als input telkens neuron *X* geactiveerd met het contextneuron dat in de contextkolom aangegeven staat. In de kolom 'geleerde output' is aangegeven welke output het netwerk heeft geleerd bij het inputpatroon. In de eerste lesson (eerste rij in de tabel) werd het netwerk getraind met *X* en *gaatOver* als input-neuronen en *onderwerp1* als output-neuron. De waarden in de outputkolommen geven de activering van alle outputneuronen aan nadat neuron *X* werd geactiveerd samen met het contextneuron dat vermeld staat in de kolom 'nieuwe context'. Als bij 'nieuwe context' *geen* staat, betekent dit dat getest werd wat de uitkomsten zijn als alleen het *X* neuron reactiveerd wordt zonder een neuron voor de context. 'Nieuwe context' betekent dat het netwerk niet heeft geleerd wat de gewenste output is als *X* en het contextneuron in de kolom 'nieuwe context' worden geactiveerd. Staat in een rij geen neuron in de kolom 'context' en/of 'geleerde output' aangegeven, dan geldt altijd het neuron dat verder boven in de kolom staat. In de tweede rij werd neuron *X* bijvoorbeeld met het contextneuron *is* aangeboden aan het netwerk. Getraind werd het netwerk in dat geval net als in de eerste rij in de tabel met *X* en *gaatOver* als inputneuronen en *onderwerp1* als outputneuron.

Uit de tabel blijkt dat het netwerk niet getrainde verbanden legt tussen *X* en de outputneuronen als *X* in een nieuwe context wordt aangeboden aan het netwerk. Nadat het netwerk had geleerd dat *X* een van de vijf boeken is (*X* en *is* als input- en een van de neuronen *T1* t/m *T5* als outputneuronen) werd *X* samen geactiveerd met *gaatOver* ofwel *geschrevenDoor*. Er blijkt dat bijna altijd de outputneuronen geactiveerd werden die ook geactiveerd werden als een van de neuronen *titel1* t/m *titel5* samen met *gaatOver* of *geschrevenDoor* aangeboden werden aan het netwerk. Als het netwerk heeft geleerd dat *X* titel1 (*T1*) *is* dat wordt *auteur1* geactiveerd als *X* en *geschrevenDoor* als geactiveerde inputneuronen worden aangeboden aan het netwerk. In tabel 1 is te zien dat het netwerk geleerd werd dat *titel1 geschreven is door auteur1*. Bij *T3* als geleerde output valt op dat beide onderwerpneuronen geactiveerd werden (.36), terwijl *titel3* alleen over *onderwerp1* gaat. De activatie van *onderwerp2* is te verklaren door het feit dat het boek geschreven is door *auteur2*. Deze auteur heeft naast boek drie alleen boeken over *onderwerp2* geschreven.

Context	Geleerde output	Nieuwe context	Output								
			Auteur1	Auteur2	Onderwerp1	Onderwerp2	T1	T2	T3	T4	T5
gaatOver	Onderwerp1	geschrevenDoor	.47								
		is			1			.47	1		
		geen	.25	.04	.48		.93		.64		
	Onderwerp2	geschrevenDoor		1		1					
		is								1	
Geschreven Door	Auteur1	gaatOver			1						
		is						1	.17		
	Auteur2	gaatOver			.30	1					
		is								.47	
is	T1	gaatOver			1						
		geschrevenDoor	1								
	T2	gaatOver			1						
		geschrevenDoor	1								
	T3	gaatOver			.36	.36					
		geschrevenDoor		1							
	T4	gaatOver				1					
		geschrevenDoor		1							
	T5	gaatOver				1					
		geschrevenDoor		1							

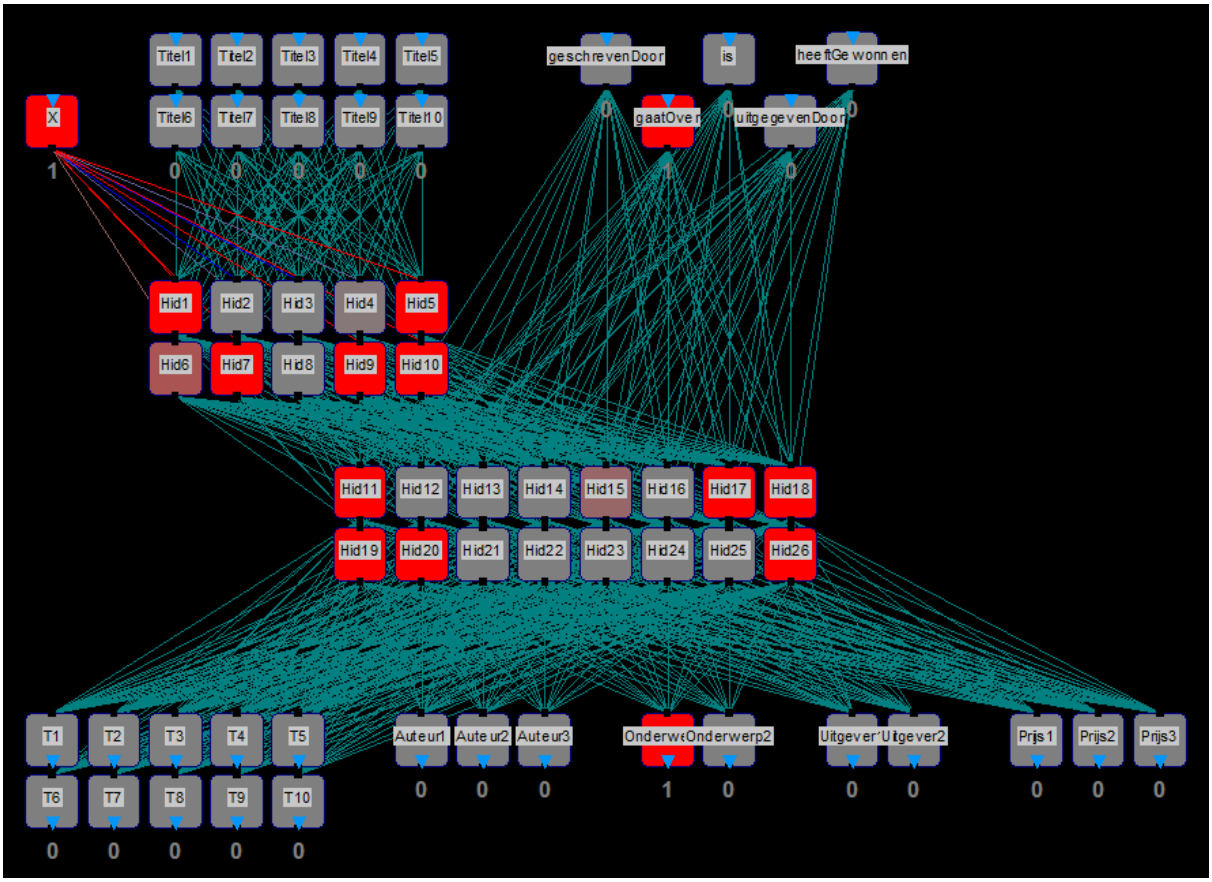
Tabel 3 **Combinaties van contextneuronen en neuron X netwerk A**

Ook bij een auteur- of onderwerpneuron als geleerde output kon het netwerk verbanden leggen als een nieuwe context werd aangeboden. Met betrekking tot de uitkomsten met *is* als nieuwe context is echter nauwelijks een patroon te herkennen.

Zoals boven al genoemd werd het netwerk geleerd dat *X* en *gaatOver* leiden tot een activatie van *onderwerp1*. Bij het activeren van alleen *X* werden *auteur1, auteur2, onderwerp2, T1* en *T3* geactiveerd, waarbij de activatie van *auteur2* heel laag was (.04). Vooral de uitkomsten met betrekking tot de auteurs zijn interessant. *Auteur1* heeft alleen over *onderwerp1* geschreven en wordt van de twee auteur neuronen het sterkst geactiveerd. *Auteur2* heeft twee boeken geschreven die over *onderwerp1* en *onderwerp2* gaan en nog een boek dat alleen over *onderwerp2* gaat. Vergeleken met *auteur1* gaan de boeken van *auteur2* vaker over *onderwerp2*. Echter er is nog wel een kleine invloed van *onderwerp1* bij deze auteur. Dit zou de lage activatie van *auteur2* in het netwerk kunnen verklaren.

Netwerk B

Netwerk B kon alle I/O-patronen succesvol te leren. Na het toevoegen van een extra neuron *X* (figuur 7) werd een analyse uitgevoerd zoals bij netwerk A. Echter werden hierbij niet alle mogelijke combinaties getest, aangezien het om een grote hoeveelheid mogelijke patronen gaat. Er werden willekeurig een aantal patronen gekozen en getest. In tabel 7In tabel 3 is te zien dat de helft van de boeken geschreven is door een man en de andere helft door een vrouw.



Figuur 7 **Netwerk B na toevoegen van neuron X**

Verder valt de helft van de boeken onder de categorie vrije tijd en de andere helft onder de categorie kennis. De gebruikte doelgroepen voor de boeken zijn kinderen, tieners en volwassenen. Naast de genoemde veranderingen werd het aantal auteurs in dit netwerk verhoogd naar zes. staan alle resultaten van de analyse.

Twee van de gebruikte *lessons* na het toevoegen van neuron *X* aan netwerk B houden in dat neuron *X* gekoppeld wordt aan *auteur1*, ofwel *auteur2*. De uitkomsten na het activeren van neuron *X* en *gaatOver* zijn respectievelijk *onderwerp1* en *onderwerp2*. Als dit wordt vergeleken met de

boekenlijst (tabel 2), dan is te zien dat *auteur1* inderdaad over *onderwerp1* heeft geschreven en *auteur2* over *onderwerp2*.

Ook met betrekking tot andere contextneuronen zijn de uitkomsten vaak te verklaren door tabel 2. Echter komen de activaties van outputneuronen niet altijd overeen met de verwachting. Na het trainen van *X gaatOver onderwerp1* levert een activatie van *X* en *geschrevenDoor* een activatie van de outputneuronen *auteur1* (.57) en *auteur3* (1) op. *Auteur3* heeft een boek over *onderwerp1* geschreven, terwijl *auteur1* vier boeken over dat onderwerp heeft geschreven. Daarom zou men een hogere activatie van *auteur1* en een lagere activatie van *auteur3* kunnen verwachten.

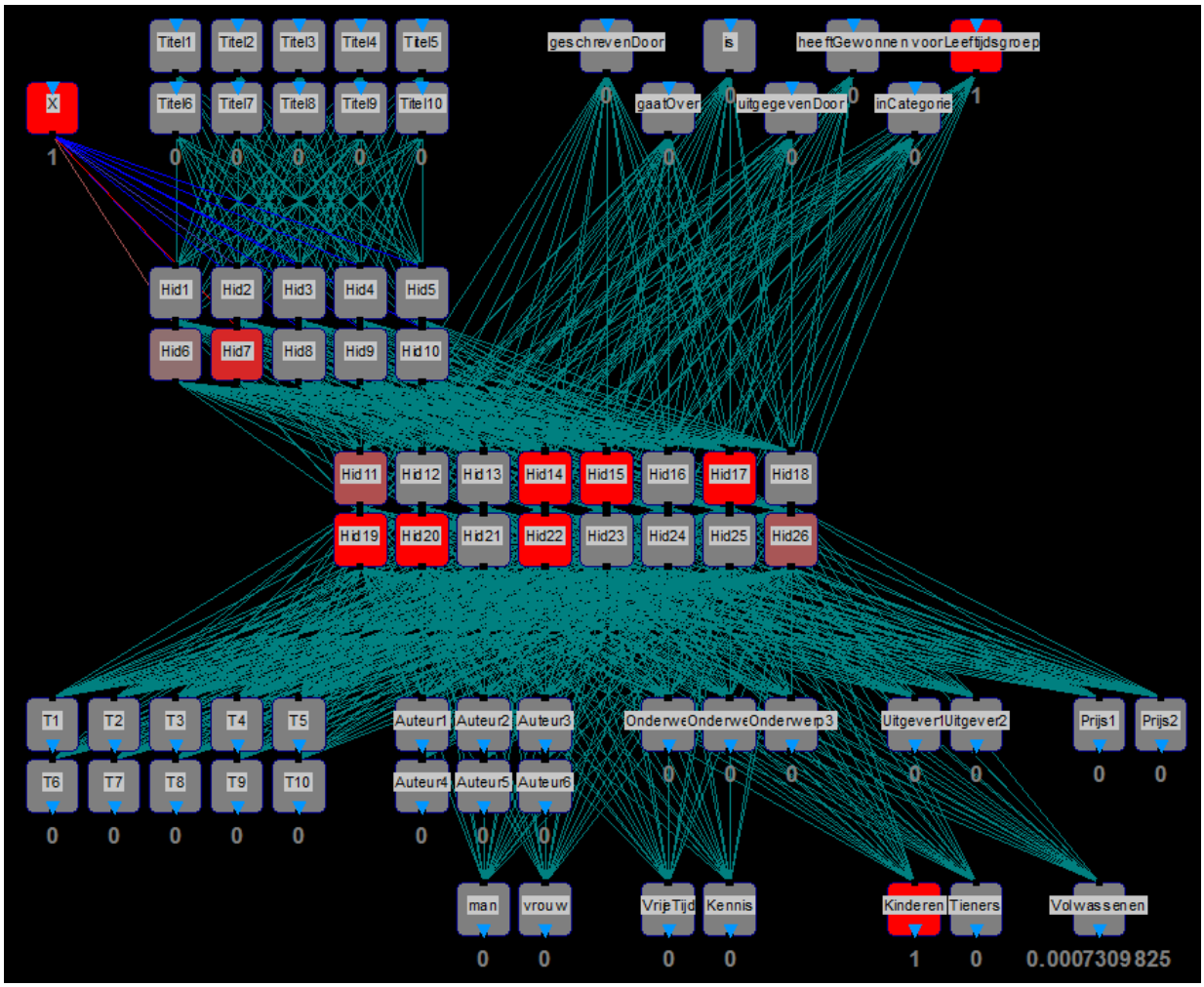
Verder werd het netwerk getraind met *X* en *gaatOver* als inputneuronen en *onderwerp2* als outputneuron. Het activeren van alleen *X* levert een activatie van de outputneuronen *auteur3, onderwerp2* en *prijs3* op, waarbij de activatie van alle neuronen 1 is. Van de vijf boeken over *onderwerp2* zijn er twee geschreven door *auteur3* en drie door *auteur2*. In het ideale geval zou *auteur2* daarom ook geactiveerd moeten worden. Dat *onderwerp2* geactiveerd wordt is een bevestiging voor de aanname dat tijdens het trainen een sterk verband ontstond tussen *X* en dat onderwerp. Verder werd een keer *prijs3* gewonnen door een boek over *onderwerp2*. Echter werd ook een keer *prijs2* gewonnen, hoewel dat neuron niet geactiveerd wordt in de outputlaag. Een reden voor de sterke activatie van *prijs3* zou kunnen zijn dat deze prijs slechts een keer gewonnen werd en dat was door een boek over *onderwerp2*. Het zou kunnen zijn dat daardoor een sterke connectie in het netwerk ontstond tussen *onderwerp2* en *prijs3*.

Context	Geleerde output	Nieuwe context	Output																			
			Auteur1	Auteur2	Auteur3	Onderwerp1	Onderwerp2	Uitgever1	Uitgever2	Prijs1	Prijs2	Prijs3	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
geschrevenDoor	Auteur1	gaatOver				1																
		is																				
		uitgegevenDoor						1														
		heeftGewonnen								1												
	Auteur2	gaatOver					.70															
		is															.83					
		uitgegevenDoor							1													
		heeftGewonnen															1					.22
gaatOver	Onderwerp1	geschrevenDoor	.57		1												.38					
		is																				
		uitgegevenDoor						.15	.24													
		heeftGewonnen								1												
	Onderwerp2	geschrevenDoor	.02	.32	1																	
		is															.64		.15		.26	
		uitgegevenDoor							1													
		heeftGewonnen								1	1	.67										
		geen			1		1					1										
geschrevenDoor	Auteur2	gaatOver					1															
EN		is																.61	.76			
heeftGewonnen	Prijs2	uitgegevenDoor							1									1				

Tabel 4 **Combinaties van contextneuronen en neuron X in netwerk B**

Netwerk C

Netwerk C werd op dezelfde manier geanalyseerd als netwerk A en B. Ook dit netwerk leerde alle I/O-patronen, zodat na de training een activatie van een geleerde inputpatroon altijd leidt tot een activatie van de van tevoren bepaalde outputpatroon. Figuur 8 laat het netwerk zien nadat neuron *X* werd toegevoegd en in tabel 8 staan de uitkomsten van de analyse met betrekking tot neuron *X*. Ook hier werden willekeurig patronen voor de analyse gekozen, omdat het niet mogelijk is om handmatig alle mogelijke combinaties te testen.



Figuur 8 **Netwerk C na toevoegen van neuron X**

Netwerk C werd onder andere getraind met neuron *X* en *voorLeeftijdsgroep* als inputneuronen en *kinderen* als outputneuronen. Bij het activeren van *X* met andere contextneuronen blijkt dat telkens outputneuronen geactiveerd worden die volgens tabel 3 in verband staan met kinderen als doelgroep. Zo zijn de uitkomsten van *X* met de neuronen

heeftGewonnen, *inCategorie* en *uitgegevenDoor* respectievelijk *prijs1, vrijeTijd* en *uitgever1*. Hierbij hebben de geactiveerde outputneuronen een minimale activatie van .99. Tevens is de activatie van de outputneuronen *auteur1* en *man* gelijk aan 1 als de inputneuronen *X* en *geschrevenDoor* worden geactiveerd. Uit tabel 3 blijkt dat alle kinderboeken geschreven zijn door mannen. Verder zijn twee kinderboeken (50 procent) geschreven door *auteur1*, wat verklaart dat *auteur1* geactiveerd wordt. De andere twee kinderboeken zijn geschreven door *auteur3*. Het outputneuron voor *auteur3* wordt in het netwerk echter niet geactiveerd. Hiernaast valt op dat het outputneuron *onderwerp1* na het activeren van *X* en *gaatOver* slechts een lage activatie vertoont (.09), terwijl alle kinderboeken over *onderwerp1* gaan.

In sommige gevallen worden in een bepaalde context ook outputneuronen geactiveerd die niet tot de aangeboden context horen. Het netwerk werd bijvoorbeeld getraind met *X* en *gaatOver* als inputneuronen en *onderwerp3* als outputneuron. Als vervolgens *X* en *inCategorie* worden geactiveerd dan worden de outputneuronen *prijs1* (.67) en *kennis* (1) geactiveerd. *Prijs1* wordt tijdens de trainingsfase gekoppeld aan *heeftGewonnen* en wordt daarom in dit geval als bijverschijnsel beschouwd.

Als neuron *X* en *is* worden geactiveerd na het succesvolle trainen van een andere inputpatroon, dan is er vaak geen patroon te herkennen met betrekking tot de uitkomsten in de outputlaag. Niet altijd worden er alleen neuronen geactiveerd die een titel representeren, maar vaak worden ook neuronen geactiveerd die samenhangen met een andere context. Bijvoorbeeld werd netwerk C getraind met *X* en *voorLeeftijdsgroep* als inputneuronen en *kinderen* als outputneuron. De uitkomst na het activeren van *X* en *is* is *onderwerp1* (.59), terwijl verwacht wordt dat een of meerdere van de neuronen *T1* t/m *T10* worden geactiveerd.

Een tweede test in verband met de *is*-relatie is het activeren van *X* en *is* na het trainen van *X* en *geschrevenDoor* als inputneuronen en *man* als outputneuron. Dit leidt tot een activatie van de outputneuronen *auteur1* (1), *onderwerp1* (.87), *uitgever1* (1) en *vrijeTijd* (1). In dit geval blijkt dat er

connecties zijn tussen het *is*-neuron en de outputneuronen die geactiveerd worden als *X* samen met een ander contextneuron wordt aangeboden aan het netwerk. *Auteur1* is hierbij de enige uitzondering, omdat dat outputneuron alleen in de geactiveerd wordt als ook *is* geactiveerd wordt. De outputneuronen *onderwerp1, uitgever1* en *vrijeTijd* worden, zoals in de tabel te zien, ook geactiveerd als *X* in combinatie met *gaatOver, uitgegevenDoor* ofwel *inCategorie* geactiveerd wordt.

Tenslotte werd het netwerk geleerd dat *X* en *is* als geactiveerde inputneuronen leiden tot *T3* als outputneuron. Wordt nu alleen *X* geactiveerd als inputneuron, dan zijn de geactiveerde outputneuronen *auteur2* (1)*, uitgever2* (.61) en *tieners* (1). Omdat het boek *T3* geschreven werd door *auteur2*, uitgegeven werd door *uitgever2* en voor *tieners* geschreven is, komen de uitkomsten goed overeen met de verwachtingen. Echter werden geen outputneuronen geactiveerd die tot een andere context dan *geschrevenDoor, uitgegevenDoor* of *voorLeeftijdsgroep* horen.

Context	Geleerde output	Nieuwe context	Output																			
			Auteur1	Auteur2	Auteur3	Auteur4	Auteur5	Auteur6	Man	Vrouw	Onderwerp1	Onderwerp2	Onderwerp3	Uitgever1	Uitgever2	Prijs1	Prijs2	VrijeTijd	Kennis	Kinderen	Tieners	Volwassenen
voorLeef-tijdsgroep	Kinderen	heeftGewonnen														.99						
		inCategorie																1				
		uitgegevenDoor												1								
		is									.59											
		gaatOver									.09											
		geschrevenDoor	1						1													
inCategorie	Kennis	voorLeeftijdsgroep																				1
		heeftGewonnen														1						
		uitgegevenDoor													1							
		is									.21		1									1
		gaatOver											1									
		geschrevenDoor				1				1												
geschreven Door	Man	gaatOver									1											
		is	1								.87			1				1				
		uitgegevenDoor												1								
		heeftGewonnen															1					
		inCategorie																1	.99			
		voorLeeftijdsgroep																		1		
gaatOver	Onderwerp3	geschrevenDoor			1				1													
		is	.69		.99				1													
		uitgegevenDoor												1								
		heeftGewonnen														1						
		inCategorie														.67			1			
		voorLeeftijdsgroep																		1		
is	T3	geen		1											.61						1	

Tabel 5 **Combinaties van contextneuronen en neuron X netwerk C. De outputneuronen T1 t/m T10 werden in geen van de testgevallen geactiveerd. Daarom zijn zij voor de overzichtelijkheid weggelaten in de tabel**

Conclusie

Als naar de resultaten van alle netwerken wordt gekeken dan valt op dat de netwerken allemaal goed in staat waren om de aangeleverde I/O-patronen te leren. Alle netwerken konden getraind worden zodat zij de patronen die in de *lessons* werden vastgelegd, na de trainingsfase zonder fouten konden vertalen naar het gewenste outputpatroon.

Werd vervolgens een extra neuron *X* toegevoegd aan het netwerk, dat een voor het netwerk onbekend boek representeerde, en werd bovendien een eigenschap van dat onbekende boek getraind, zo bleek dat het netwerk automatisch verbanden legde tussen het nieuwe neuron en outputneuronen die in een andere context staan en die met betrekking tot het nieuwe neuron niet getraind werden.

Hoewel niet alle resultaten ofwel output-patronen met het extra neuron *X* overeenkomen met de verwachtingen, schijnen de netweken al redelijk in staat te zijn om verbanden te leggen tussen eerder verworven – getrainde – 'kennis' en nieuwe input.

Als het *X* neuron in de *lesson* gekoppeld wordt aan een bepaalde auteur dan kan het netwerk het neuron vervolgens automatisch in verband te brengen met een bepaalde onderwerpen waarover die auteur heeft geschreven. Dit werd duidelijk door *X* samen met *gaatOver* te activeren. Deze onderwerpen waren meestal sterk gerelateerd aan de auteur van het boek. Hetzelfde geldt andersom: Als een neuron gekoppeld wordt aan een onderwerp dan wordt hetzelfde neuron in verband gebracht met een auteur die over dat onderwerp heeft geschreven.

Hiernaast bleek dat de netwerken hebben geleerd welke outputneuronen tot een bepaald contextneuron hoorden. Dit betekent dat bijvoorbeeld de activatie van een neuron *gaatOver* voornamelijk leidt tot een activatie van onderwerpneuronen. Echter waren er ook bijverschijnselen: Naast de outputneuronen die tot het contextneuron hoorden werden soms neuronen geactiveerd die tot een andere context horen.

Het *is*-contextneuron blijkt in alle netwerken een uitzondering te zijn. Een activatie van dit neuron heeft niet consequent geleid tot een activatie van een titelneuron als output, maar vaak werden ook of zelfs alleen maar neuronen geactiveerd die met andere contextneuronen getraind werden, zoals onderwerp- of auteurneuronen.

Het blijkt dat er geen sterke connecties ontstonden tussen het contextneuron *is* en de titel-outputneuronen. In plaats daarvan leidt de activatie van het *is*-neuron met *X* tot een activatie van outputneuronen die – in een andere context – in verband staan met *X* ofwel met het getrainde I/O-patroon.

De activatie van alleen *X* leidde tot een activatie van outputneuronen die met betrekking tot eerder verworven kennis in relatie staan met het samen met *X* getrainde outputneuron. Werd het netwerk geleerd dat boek *X* over *onderwerp1* gaat dan worden bij het activeren van alleen *X* als inputneuron vooral outputneuronen geactiveerd die met betrekking tot andere boeken in relatie staan met *onderwerp1* (bijvoorbeeld een neuron dat een auteur representeert die vaak over *onderwerp1* heeft geschreven). Echter komen de uitkomsten niet altijd volledig overeen met de verwachte output als gekeken wordt naar de eigenschappen van boeken in de netwerken.

Discussie

Bij het maken, trainen en analyseren van de netwerken kwamen er meerdere moeilijkheden naar voren. Deze werden vooral veroorzaakt door het grote aantal mogelijke netwerken en analysemogelijkheden. Bij het construeren van de netwerken moesten beslissingen worden gemaakt met betrekking tot het aantal neuronen in de twee lagen met *hidden* neuronen. In de representatielaag werd net als in het model van Rumelhart en Todd (1993) hetzelfde aantal neuronen gebruikt als in de itemlaag. Omdat de in dit onderzoek gemaakte netwerken echter met betrekking tot het aantal input- en outputneuronen en de verhouding ervan sterk verschillen van het model van Rumelhart en Todd, kon het aantal gebruikte neuronen in de *hidden* laag niet gebaseerd worden op hun model. In een vervolgonderzoek zouden meerdere netwerken gemaakt en getest kunnen worden, door alleen het aantal *hidden* neuronen te variëren. In dit onderzoek werd niet duidelijk welke invloed het verwijderen of toevoegen van *hidden* neuronen zou hebben op de prestatie van de netwerken.

Ook bij het trainen speelde de grootte van het netwerk en in het bijzonder het aantal input- en outputneuronen een rol. Het aantal mogelijke patronen voor het trainen van een netwerk is $n_{titels} * m_{contextneuronen}$, waarbij n_{titels} het aantal boektitels is (inputneuronen in de itemlaag) en $m_{contextneuronen}$ het aantal contextneuronen (inputneuronen in de contextlaag). Bij een netwerk met vijf boektitels en drie categorieën (contextneuronen) zijn er daarom al 15 I/O-patronen die het netwerk moet leren. Naarmate een netwerk groter en complexer wordt, wordt ook het aantal I/O-patronen voor het trainen van het netwerk groter. In dit onderzoek werden alle I/O-patronen handmatig in een csv-bestand ingevuld en in MemBrain geïmporteerd. Aangezien de tijd die nodig is om een trainingsbestand te maken, was het niet mogelijk om meerdere grote netwerken te maken, te trainen en te testen. Ook dit zou verder onderzocht kunnen worden.

In dit onderzoek werden de netwerken getest door willekeurige combinaties van inputneuronen aan te bieden, omdat het door de grote hoeveelheid mogelijke inputpatronen niet mogelijk is om elke combinatie te testen. Om de analyse betrouwbaarder en preciezer te maken, zouden de netwerken met behulp van andere programma's geanalyseerd kunnen worden. Door het gebruik van analyseprogramma's zou het mogelijk zijn om meer inputpatronen te testen en te vergelijken met de verwachte uitkomst.

Omdat de netwerken allemaal precies een keer getraind werden en de analyse op basis van deze trainingssessies werd uitgevoerd, kunnen geen uitspraken worden gemaakt over de algemene betrouwbaarheid van de netwerken. Er werd niet getest of de netwerken dezelfde resultaten opleveren als alle gewichten van connecties en alle activaties van neuronen binnen de netwerken gerandomiseerd (teruggezet) worden en de netwerken vervolgens opnieuw worden getraind. Als de analyse van de netwerken uitgevoerd kan worden met behulp van een programma, dan zou ook deze vraag verder onderzocht kunnen worden.

Uit de analyse van de netwerken bleek dat de *is*-context zich onderscheidt van de andere contexten. Dit bleek al in onderzoek van Rogers en McClelland (2008). Zij toonden aan dat verschillen tussen concepten in de *is*-context een grotere rol spelen dan het in een andere context het geval is. Dit heeft te maken met connecties met de hidden layer. In tegenstelling tot andere contextneuronen worden verbanden tussen inputneuronen in de item-laag en outputneuronen in de output-laag in de is context één op één geleerd. Dit betekent in dit onderzoek dat met betrekking tot de titel-outputneuron slechts een connectie is met de inputneuronen: *titel1-isT1*, *titel2-isT2*, enz. Als dit gerelateerd wordt aan het aantal hidden neuronen, dan wordt duidelijk dat het netwerk niet 'gedwongen' wordt om een gedistribueerde representatie van de titel-outputneuronen in de hidden laag te maken. Onderwerp-ouputneuronen in de *gaatOver* context worden tijdens de trainingsfase aan meerdere boektitels gerelateerd (meerdere boeken kunnen hetzelfde onderwerp hebben). Daardoor moet het netwerk binnen de laag met hidden neuronen een gedistribueerd netwerk ontwikkelen dat leidt tot de gewenste output. Om een

dergelijk netwerk te maken dat voor alle inputneuronen werkt zijn veel connecties binnen het netwerk nodig. Het aantal connecties dat nodig is om de gewenste output te leveren in de *is*-context (bijvoorbeeld *T1*) is daarentegen veen kleiner. Dit betekent tegelijk ook dat de connecties minder verspreid zijn over het netwerk dan connecties die nodig zijn om de correcte outputneuronen in een andere context te leveren. In een vervolgonderzoek zou onderzocht kunnen worden of het variëren van het aantal hidden neuronen dit effect verandert.

Afsluitend kan geconcludeerd worden dat de geconstrueerde netwerken nog niet ideaal werken, maar dat zij wel in staat zijn om verbanden te leggen tussen verschillende categorieën. Door middel van hulpprogramma's zouden de netwerken verder ontwikkeld kunnen worden en zou het toevoegen van nieuwe informatie gemakkelijker gemaakt kunnen worden. Voor boeken-, spellen- of dvd-winkels zou een beter ontwikkeld netwerk van toe gevoegde waarde kunnen zijn, doordat het net werk automatisch kan analyseren, welke kenmerken bijvoorbeeld het favoriete boek van een klant heeft en welke andere boeken op dat boek lijken. Echter zou eerst getest moeten worden hoeveel data het netwerk kan verwerken en of er een limiet aan verbonden is. Dit is van belang voor de winkels, omdat zij al hun artikelen in het netwerk zouden moeten kunnen opslaan. Zoals te lezen zijn er veel mogelijkheden voor verder onderzoek en het ontwikkelen van het netwerk.

Referenties

Collins, A. M. & Quillian, M. R. (1969) Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, *8*, 240–247. doi:10.1016/S0022-5371(69)80069-1

McCloskey, M., & Glucksberg, S.(1979). Decision processes in verifying category membership statements: implications for models of semantic memory. *Cognitive Psychology*, *11*, 1-37. doi: 10.1016/0010-0285(79)90002-1

Murphy, G. L. & Bronell, H. H. (1985). Category differentiation in object recognition: typicality constraints on the basic category advantage. *Journal of Experimental Psychology: Learning, Memory, and Cognition. 11(1)*, 70-84. doi: 10.1037/0278-7393.11.1.70

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.

Rumelhart, D. E. (1990). Brain style computation: Learning and generalization. In S. F. Zornetzer, J. L. Davis, and C. Lau (Eds.), *An Introduction to Neural and Electronic Networks* (pp. 405-420). San Diego, CA: Academic Press.

Rumelhart, D. E., & Todd, P. M. (1993). Learning and connectionist representations. In D. E. Meyer, and S. Kornblum (Eds.), *Attention and Performance XIV: Synergies in Experimental Psychology, Artifical Intelligence, and Cognitive Neuroscience* (pp. 3–35). Cambridge, MA: MIT Press.

Rogers, T. T., & McClelland, J. L. (2008). Précis of Semantic Cognition: A Parallel Distributed Processing Approach. *Behavioral and Brain Sciences, 31*, 689–749. doi:10.1017/S0140525X0800589X

Sakla, S. S. S., & Ashour, A. F. (2005). Prediction of tensile capacity of single adhesive anchors using neural networks. *Computer and Structures*, *83*, 1792-1803. doi: 10.1016/j.compstruc.2005.02.008

Bijlage

Trainingsdata netwerk A

Titel1	Titel2	Titel3	Titel4				
geschrevenDoor	gaatOver	is					
				Auteur1	Auteur2		
				Onderwerp1	Onderwerp2		
				T1	T2	T3	T4
1	0	0	0				
1	0	0					
				1	0		
				0	0		
				0	0	0	0
0	1	0	0				
1	0	0					
				1	0		
				0	0		
				0	0	0	0
0	0	1	0				
1	0	0					
				0	1		
				0	0		
				0	0	0	0
0	0	0	1				
1	0	0					
				0	1		
				0	0		
				0	0	0	0
1	0	0	0				
0	1	0					
				0	0		
				1	0		
				0	0	0	0
0	1	0	0				
0	1	0					
				0	0		
				1	0		
				0	0	0	0
0	0	1	0				
0	1	0					
				0	0		
				1	0		
				0	0	0	0
0	0	0	1				

0	1	0					
				0	0		
				0	1		
				0	0	0	0
1	0	0	0				
0	0	1					
				0	0		
				0	0		
				1	0	0	0
0	1	0	0				
0	0	1					
				0	0		
				0	0		
				0	1	0	0
0	0	1	0				
0	0	1					
				0	0		
				0	0		
				0	0	1	0
0	0	0	1				
0	0	1					
				0	0		
				0	0		
				0	0	0	1

Parameters van neuronen en links in netwerk A

```
/// parameter array for all neurons
const SNeuronParms NEURON_PARMS[] =
{
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1,
	MB_AF_IDENTICAL_0_1
};

/// additional parameters for all hidden neurons
const SHiddenNeuronParms
NEURON_PARMS_HIDDEN[] =
{
	(MB_FLOAT)0.3713342, 0, 6,
	(MB_FLOAT)-0.1750792, 6, 6,
	(MB_FLOAT)0.1040472, 12, 6,
	(MB_FLOAT)0.419668, 18, 6,
	(MB_FLOAT)-0.5835868, 24, 6,
	(MB_FLOAT)0.8841635, 30, 8,
	(MB_FLOAT)0.212348, 38, 8,
	(MB_FLOAT)-0.1973516, 46, 8,
	(MB_FLOAT)-0.4336009, 54, 8,
	(MB_FLOAT)0.2019473, 62, 8,
	(MB_FLOAT)0.5290638, 70, 8,
	(MB_FLOAT)-0.06435704, 78, 8,
	(MB_FLOAT)0.2668961, 86, 8,
	(MB_FLOAT)-0.1556596, 94, 8
};

/// additional parameters for all output neurons
const SOutputNeuronParms
NEURON_PARMS_OUTPUT[] =
{
	(MB_FLOAT)0.02491592, 102, 9,
	(MB_FLOAT)0.04394566, 111, 9,
	(MB_FLOAT)0.377433, 120, 9,
	(MB_FLOAT)-0.9287041, 129, 9,
	(MB_FLOAT)0.1827724, 138, 9,
	(MB_FLOAT)-0.1001895, 147, 9,
	(MB_FLOAT)0.2873461, 156, 9,
	(MB_FLOAT)0.09634973, 165, 9,
	(MB_FLOAT)-0.3981763, 174, 9
};

/// parameters for all links
const SNeuralLinkParms
NEURAL_LINK_PARMS[] =
{
	(MB_FLOAT)0.4023486, 0,
	(MB_FLOAT)-172.4969, 1,
	(MB_FLOAT)0.6513177, 2,
	(MB_FLOAT)59.15486, 3,
	(MB_FLOAT)40.08971, 7,
	(MB_FLOAT)0.5, 8,
	(MB_FLOAT)-153.3583, 0,
	(MB_FLOAT)163.9233, 1,
	(MB_FLOAT)-73.76974, 2,
	(MB_FLOAT)-0.09917229, 3,
	(MB_FLOAT)-2.427555, 7,
	(MB_FLOAT)0.5, 8,
	(MB_FLOAT)-89.98376, 0,
	(MB_FLOAT)0.1475714, 1,
	(MB_FLOAT)25.67703, 2,
	(MB_FLOAT)2.501858, 3,
	(MB_FLOAT)-35.46279, 7,
	(MB_FLOAT)0.5, 8,
	(MB_FLOAT)-137.2503, 0,
	(MB_FLOAT)120.0598, 1,
```

(MB_FLOAT)-59.57887, 2,
(MB_FLOAT)-28.93247, 3,
(MB_FLOAT)0.4588461, 7,
(MB_FLOAT)0.5, 8,
(MB_FLOAT)12.5783, 0,
(MB_FLOAT)83.25596, 1,
(MB_FLOAT)0.2790233, 2,
(MB_FLOAT)-23.84449, 3,
(MB_FLOAT)-17.567, 7,
(MB_FLOAT)0.5, 8,
(MB_FLOAT)-89.03544, 9,
(MB_FLOAT)-11.7439, 10,
(MB_FLOAT)-0.03567298, 11,
(MB_FLOAT)0.1850635, 12,
(MB_FLOAT)-4.115958, 4,
(MB_FLOAT)10.93726, 5,
(MB_FLOAT)-26.3587, 6,
(MB_FLOAT)1.558906, 13,
(MB_FLOAT)0.003092902, 9,
(MB_FLOAT)-0.08599256, 10,
(MB_FLOAT)-0.01216, 11,
(MB_FLOAT)0.2369514, 12,
(MB_FLOAT)-60.47086, 4,
(MB_FLOAT)0.1612237, 5,
(MB_FLOAT)0.5734511, 6,
(MB_FLOAT)-0.3653895, 13,
(MB_FLOAT)4.086321, 9,
(MB_FLOAT)-165.8112, 10,
(MB_FLOAT)-0.4660542, 11,
(MB_FLOAT)-109.4091, 12,
(MB_FLOAT)-0.229787, 4,
(MB_FLOAT)14.24122, 5,
(MB_FLOAT)-0.4294642, 6,
(MB_FLOAT)0.2151236, 13,
(MB_FLOAT)-36.06222, 9,
(MB_FLOAT)19.64949, 10,
(MB_FLOAT)-14.13424, 11,
(MB_FLOAT)-0.6565506, 12,
(MB_FLOAT)-1.348626, 4,
(MB_FLOAT)-1.731969, 5,
(MB_FLOAT)0.7035173, 6,
(MB_FLOAT)30.00626, 13,
(MB_FLOAT)0.3363007, 9,
(MB_FLOAT)0.2476448, 10,
(MB_FLOAT)58.3278, 11,
(MB_FLOAT)-1.854101, 12,
(MB_FLOAT)2.223782, 4,
(MB_FLOAT)1.419956, 5,
(MB_FLOAT)-0.06751554, 6,
(MB_FLOAT)0.05812101, 13,
(MB_FLOAT)74.85789, 9,
(MB_FLOAT)-129.6029, 10,
(MB_FLOAT)0.3866714, 11,
(MB_FLOAT)-94.03124, 12,
(MB_FLOAT)3.153565, 4,
(MB_FLOAT)0.01703946, 5,
(MB_FLOAT)-0.0753591, 6,
(MB_FLOAT)-35.4511, 13,
(MB_FLOAT)0.4114298, 9,
(MB_FLOAT)-0.1437386, 10,
(MB_FLOAT)-1.994932, 11,
(MB_FLOAT)-2.159243, 12,
(MB_FLOAT)-1.031458, 4,
(MB_FLOAT)62.23438, 5,
(MB_FLOAT)-54.79739, 6,
(MB_FLOAT)-0.7239584, 13,
(MB_FLOAT)-107.1846, 9,
(MB_FLOAT)-4.407818, 10,
(MB_FLOAT)-0.5260885, 11,
(MB_FLOAT)-7.352483, 12,
(MB_FLOAT)8.242699, 4,
(MB_FLOAT)-2.233763, 5,
(MB_FLOAT)-52.74686, 6,
(MB_FLOAT)-0.6716719, 13,
(MB_FLOAT)14.67906, 4,
(MB_FLOAT)-33.3497, 5,
(MB_FLOAT)0.7413801, 9,
(MB_FLOAT)0.04719272, 10,
(MB_FLOAT)-0.170525, 11,
(MB_FLOAT)-14.8242, 12,
(MB_FLOAT)-10.85749, 6,
(MB_FLOAT)0.07705909, 13,
(MB_FLOAT)7.38266, 14,
(MB_FLOAT)-5.573292, 15,
(MB_FLOAT)-4.473188, 16,
(MB_FLOAT)0.1049395, 17,
(MB_FLOAT)-0.08508359, 18,
(MB_FLOAT)-2.138716, 19,
(MB_FLOAT)0.4776271, 20,
(MB_FLOAT)0.356202, 21,
(MB_FLOAT)-7.180313, 22,
(MB_FLOAT)-0.4654012, 14,
(MB_FLOAT)-0.9786194, 15,
(MB_FLOAT)0.006869629, 16,
(MB_FLOAT)-19.00201, 17,
(MB_FLOAT)0.007527963, 18,
(MB_FLOAT)0.3351172, 19,
(MB_FLOAT)1.21757, 20,
(MB_FLOAT)-0.3814489, 21,
(MB_FLOAT)-5.598265, 22,

```
(MB_FLOAT)-0.3680538, 14,
(MB_FLOAT)-14.09892, 15,
(MB_FLOAT)-0.3519468, 16,
(MB_FLOAT)0.6374884, 17,
(MB_FLOAT)0.7119118, 18,
(MB_FLOAT)-1.698442, 19,
(MB_FLOAT)-4.953266, 20,
(MB_FLOAT)-0.02019005, 21,
(MB_FLOAT)1.154241, 22,
(MB_FLOAT)-0.7088193, 14,
(MB_FLOAT)-22.83438, 15,
(MB_FLOAT)-5.475389, 16,
(MB_FLOAT)-22.12148, 17,
(MB_FLOAT)-1.321564, 18,
(MB_FLOAT)0.9816445, 19,
(MB_FLOAT)0.488433, 20,
(MB_FLOAT)-0.574815, 21,
(MB_FLOAT)5.493714, 22,
(MB_FLOAT)-0.3216407, 14,
(MB_FLOAT)-18.48431, 15,
(MB_FLOAT)12.12119, 16,
(MB_FLOAT)1.153553, 17,
(MB_FLOAT)-25.81159, 18,
(MB_FLOAT)-1.320654, 19,
(MB_FLOAT)-0.6073443, 20,
(MB_FLOAT)-0.2416218, 21,
(MB_FLOAT)-1.505397, 22,
(MB_FLOAT)-0.2916592, 14,
(MB_FLOAT)2.657991, 15,
(MB_FLOAT)-77.86802, 16,
(MB_FLOAT)0.469447, 17,
(MB_FLOAT)-0.09909708, 18,
(MB_FLOAT)-1.727358, 19,
(MB_FLOAT)-0.8704011, 20,
(MB_FLOAT)-0.1420103, 21,
(MB_FLOAT)-6.236023, 22,
(MB_FLOAT)-0.1457843, 14,
(MB_FLOAT)-10.84319, 15,
(MB_FLOAT)1.151944, 16,
(MB_FLOAT)0.2275923, 17,
(MB_FLOAT)1.073808, 18,
(MB_FLOAT)0.2189994, 19,
(MB_FLOAT)-2.32048, 20,
(MB_FLOAT)-0.3635174, 21,
(MB_FLOAT)-13.52057, 22,
(MB_FLOAT)-0.4865227, 14,
(MB_FLOAT)0.3006042, 15,
(MB_FLOAT)-1.196565, 16,
(MB_FLOAT)-12.49515, 17,
(MB_FLOAT)1.189822, 18,
(MB_FLOAT)0.01069862, 19,
(MB_FLOAT)-1.928932, 20,
(MB_FLOAT)-0.4369743, 21,
(MB_FLOAT)-1.128102, 22,
(MB_FLOAT)-0.34182, 14,
(MB_FLOAT)1.610189, 15,
(MB_FLOAT)-0.225404, 16,
(MB_FLOAT)-0.72083, 17,
(MB_FLOAT)-6.371729, 18,
(MB_FLOAT)0.4303048, 19,
(MB_FLOAT)-4.269864, 20,
(MB_FLOAT)-0.8345941, 21,
(MB_FLOAT)-8.80378, 22
};
```