Welkom. Dit tentamen duurt 3 uur en telt 20 vragen: 4 open vragen en 16 meerkeuzevragen. Wel: rekenmachine. Niet: literatuur, aantekeningen, programmeerbare rekenmachine, mobiel.

Veel succes!

Open vragen

Elk antwoord op een open vraag dient te zijn voorzien van een uitwerking, dat wil zeggen, van een berekening of een motivatie. Van Opgaven 2 en 4 dienen na afloop van de berekening de getallen te worden ingevuld op het antwoordenblad.

1. We beschouwen een ecologie die bestaat uit k populaties $\{P_i\}_{i=1}^k$. Elk individu uit populatie P_i hanteert strategie S_i. Omschrijf, gebruikmakend van deze notatie, de volgende begrippen.

   (a) Evolutionair stabiele toestand. **Antw**. Een toestand waarbij minimale verstoringen van $|P_i|$, $1 \leq i \leq k$ hersteld worden door het evolutionaire proces. □

   (b) Evolutionair stabiele strategie (ESS). **Antw**. Een toestand die niet kan worden verstoord door het toevoegen van een extra populatie P_{k+1} met een strategie S_{k+1}. Mocht dat gebeuren met $|P_{k+1}| = \epsilon$, dan zal $|P_{k+1}| \leq \epsilon$ in volgende generaties. □

2. Gegeven is het lineaire perceptron zoals afgebeeld in Fig. 2. We zetten de leersnelheid ervan op $\alpha = 0.5$.

   Verder zijn de volgende leervoorbeelden gegeven, met X_1 en X_2 als inputs en T als de gewenste output waarde.

$X1$	$X2$	T
0	0	0.5
−1	0	−0.5
1	1	0
0	1	−1

   (a) Bepaal hoe de gewichten eruit zien na één online gradient descent update, over alle voorbeelden één keer van boven naar beneden. **Antw**. In de volgende tabel representeert c_i de *onmiddellijke correctie*. Er geldt:

$$c_i = (d - y)x_i$$

x1	x2	d	y	c1	c2	c3	w1	w2	w3	
							0.40	-0.80	0.60	INIT
0.00	0.00	0.50	0.60	0.00	0.00	-0.10	0.40	-0.80	0.55	ADAPT
-1.00	0.00	-0.50	0.15	0.65	0.00	-0.65	0.73	-0.80	0.23	ADAPT
1.00	1.00	0.00	0.15	-0.15	-0.15	-0.15	0.65	-0.88	0.15	ADAPT
0.00	1.00	-1.00	-0.73	0.00	-0.27	-0.27	0.65	-1.01	0.01	ADAPT

   De update vindt elke keer plaats. □

   (b) Bepaal hoe de gewichten eruit zien na één batch gradient descent update met de delta-leerregel. **Antw**. In de volgende tabel representeert c_i de *cumulatieve correctie*. Er geldt:

$$c_{i,\text{nieuw}} = c_{i,\text{oud}} + (d - y)\cdot x_i$$

x1	x2	d	y	c1	c2	c3	w1	w2	w3	
							0.40	-0.80	0.60	INIT
0.00	0.00	0.50	0.60	0.00	0.00	-0.10				
-1.00	0.00	-0.50	0.20	0.70	0.00	-0.80				
1.00	1.00	0.00	0.20	0.50	-0.20	-1.00				
0.00	1.00	-1.00	-0.20	0.50	-1.00	-1.80				
							0.65	-1.30	-0.30	ADAPT

   De gewichten worden tussentijds niet ge-update; alleen op het eind gebeurd dat één keer. Dan geldt

$$w_i^{\text{nieuw}} = w_i^{\text{oud}} + \alpha\cdot c_i,$$

   voor $i = 1, 2$. □

3. We bekijken een reinforcement leerprobleem. Daar hoort de toestanden-graaf in Fig. 3 bij. De policy $\Pi(p)$ is gegeven als volgt

$$\Pi(p) : \quad \begin{cases} \Pr(S_i, \text{met de klok mee}, S_j) = p & \text{als } S_j \text{ kloksgewijs na } S_i \text{ komt en } S_i \neq E \\ \Pr(S_i, \text{tegen de klok in}, S_j) \ = 1 - p & \text{als } S_i \text{ kloksgewijs na } S_j \text{ komt en } S_i \neq E \end{cases}$$

   Kortom, bij twee actie-alternatieven is de kans op een stap met de klok mee gelijk aan p en de kans op een stap tegen de klok in gelijk aan $1 - p$. Voor iedere stap is de immediate reward gelijk aan -1. Bepaal de waarden van alle toestanden voor de volgende policies.

(a) $\Pi(1.0)$. **Antw**. De waarde van elke knoop is een de waarde van elke opvolger plus de immediate reward naar elke opvolger. Dit komt neer op het oplossen van de volgende vijf vergelijkingen met vijf onbekenden:

$$A = 1\cdot(-1+B);\ B = 1\cdot(-1+C);\ C = 1\cdot(-1+D);\ D = 1\cdot(-1+E);\ E = 0.$$

Hier komt uit: $(A, B, C, D, E) = (-4, -3, -2, -1, 0)$. $\square$

(b) $\Pi(0.5)$. **Antw**. De waarde van elke knoop is een half keer (de waarde van elke buur plus de immediate reward naar elke buur). Dit komt neer op het oplossen van de volgende vijf vergelijkingen met vijf onbekenden:

$$A = 0.5(-1+E) + 0.5(-1+B);\ B = 0.5(-1+A) + 0.5(-1+C);\ C = 0.5(-1+B) + 0.5(-1+D);$$
$$D = 0.5(-1+C) + 0.5(-1+E);\ E = 0.$$

Oftewel:

$$\begin{pmatrix} 1 & -0.5 & 0 & 0 & -0.5 \\ -0/5 & 1 & -0.5 & 0 & 0 \\ 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & 0 & -0.5 & 1 & -0.5 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 0 \end{pmatrix}$$

Om dit stelsel op te lossen kun je het best Gauss-eliminatie gebruiken. De meeste entries zijn nul, dus dat zal snel gaan. Oplossing: $(A, B, C, D, E) = (-4, -6, -6, -4, 0)$. $\square$

4. We bekijken een reinforcement leerprobleem met discountfactor $\gamma = 1.0$. Er worden twee runs uitgevoerd, resulterend in Epoch 1 = A, B, C, T_1 en Epoch 2 = D, C, T_2. De immediate rewards staan afgebeeld in Fig. 4.

(a) Bepaal de waarden van alle knopen na Monte-Carlo sampling. **Antw**. Rewards-to-go wordt van achter naar voor uitgerekend. Deze waarden beïnvloeden elkaar niet. De (eind-) waarde van een knoop is het gemiddelde van alle rewards-to-go.

	A	B	C	D	T_1	T_2
0.						
Epoch 1.	18	16	10			
Epoch 2.			8	12		
Eind.	18	16	9	12		

$\square$

(b) Bepaal de waarden van alle knopen na temporal difference learning met leerfactor $\alpha = 0.5$. **Antw**. Doe voor elke stap van i naar j in een epoch: Als j terminaal:

$$V(i) := V(i) + \alpha\,(R(i,j) - V(i))$$

Als j niet terminaal:

$$V(i) := V(i) + \alpha\,(R(i,j) + \gamma V(j) - V(i))$$

Voor een update gebruiken we dus de waarde van een volgende toestand. Idee: geef elke keer $V(i)$ een duwtje in de gewenste richting. Updates beïnvloeden elkaar, of het nu in dezelfde run is of tussen runs in. De (eind-) waarde van een knoop is de waarde van de laatste update.

	A	B	C	D	T_1	T_2
0.						
Epoch 1.	1	3	5			
Epoch 2.			6.5	4.5		
Eind.	1	3	6.5	4.5		

$\square$

In beide berekeningen mag je er van uitgaan dat alle toestanden beginwaarde nul hebben.

Meerkeuze vragen

Bij elke meerkeuzevraag is steeds precies één antwoord het juiste. Wel kunnen andere antwoorden "bijna juist" of "deels juist" zijn. Mochten er meerdere goede antwoorden zijn, dan geldt het beste antwoord. Omdat er verschillende versies van de opgaven bestaan, correspondeert de volgorde van de opgaven niet altijd met de volgorde van de stof zoals die behandeld is in de colleges.

1. Welke van de volgende symmetrische spelen bezitten identieke Nash-evenwichten en Pareto-optima?

- Prisoner's dilemma met $R = 3$, $T = 5$, $S = 0$ en $P = 1$.
- Chicken met $R = 0$, $T = 1$, $S = -1$ en $P = -5$.
- Stag hunt met $R = 4$, $T = 3$, $S = 1$ en $P = 3$.
- Battle of the sexes met $R = 0$, $T = 3$, $S = 2$ en $P = 1$.

(a) Prisoner en Stag.

(b) Prisoner en Battle.

(c) Chicken en Stag.

$\surd$ Chicken en Battle.

Antw. $\square$

2. Gespeeld wordt Chicken met $R = 0$, $T = 1$, $S = -1$ en $P = -20$. Geef alle Nash-equilibria als beide spelers spelen volgens een gemixte strategie.

   (a) $\{(0, 1), (1, 0), (0.90, 0.90)\}$.

   (b) $\{(0, 0), (1, 1), (0.90, 0.90)\}$.

   √ $\{(0, 1), (1, 0), (0.95, 0.95)\}$.

   (d) $\{(0, 0), (1, 1), (0.95, 0.95)\}$.

   Antw. Het interne equilibrium kan worden gevonden de stationaire punten te bepalen van

$$\begin{aligned}\text{Payoff}_A = p(q\text{Payoff}_A(C, C) + \\ (1-q)\text{Payoff}_A(C, D)) + \\ (1-p)(q\text{Payoff}_A(C, C) + \\ (1-q)\text{Payoff}_A(C, D)).\end{aligned}$$

   Dus partieel differentiëren naar p, partieel differentiëren naar q, en dan beide vergelijkingen op nul stellen. Dat levert op $(p, q) = (0.95, 0.95)$. $\Box$

3. Gespeeld wordt het standaard Prisoner's dilemma. Geef alle Nash-equilibria als beide spelers spelen volgens een gemixte strategie.

   (a) $\emptyset$.

   √ $\{(0, 0)\}$.

   (c) $\{(1, 0), (0, 1)\}$.

   (d) $\{(0, 0), (1, 0), (0, 1)\}$.

   Antw. Geen intern equilibrium. $\Box$

4. In het geïtereerde prisoner's dilemma laten we de volgende strategiën tegen elkaar spelen: ALL-C, ALL-D, RAND, en TFT. We krijgen de volgende matrix

	$ALL-C$	$ALL-D$	$RAND$	TFT
$ALL-C$	3	0	1.5	3
$ALL-D$	5	1	3	$\downarrow 1$
$RAND$	4	0.5	2.25	2.25
TFT	3	$\uparrow 1$	2.25	3

   Welke entry in deze matrix is fout? Formuleer je antwoord als (kolom, rij).

   (a) (ALL-D, TFT)

   (b) (TFT, ALL-D)

   (c) (RAND, ALL-D)

   √ Het goede antwoord staat er niet bij.

   Antw. (RAND, ALL-D) is nog twijfelachtig maar kun je zelf snel enkele ronden simuleren. (Strikt genomen volgt Payoff = 0.5 uit de centrale limietstelling.) $\Box$

5. Wat is Pavlov voor een strategie?

   (a) Als TFT, maar pas vergelden na twee opeenvolgende defects van de tegenstander.

   √ Werk nu samen als en slechts als zowel jij als de tegenstander in de vorige ronde dezelfde strategie hanteerden.

   √ Wissel van strategie als de tegenstander in de vorige ronde verzaakte.

   (d) Wissel van strategie als die bij gelijkblijvend gedrag van de tegenstander meer zou opleveren.

   Antw. Win-stay, loose-shift. Item (c) is ook goed:

```
Pavlov:
(c, c) -> c (samen hetzelfde)
(c, d) -> d
(d, c) -> d
(d, d) -> c (samen hetzelfde)

Wissel van strategie als de tegenstander in de
verzaakte:
(c, c) -> c
(c, d) -> d (wissel)
(d, c) -> d
(d, d) -> c (wissel)
```

   Item (d) is Nash. $\Box$

6. Waar lijdt, volgens Flake, de niet-spatiële evolutionaire variant van het IPD aan?

   (a) Het experiment is afhankelijk van te veel parameters, zoals het aantal strategieën K, de grootte van de startpopulatie per strategie P_i, $1 \leq i \leq K$, het aantal ronden N, en het aantal episoden E.

   (b) De initiële proportie-keuze van strategieën beïnvloedt de uiteindelijke populatie-proporties. Dat zou niet mogen.

   √ Er wordt aangenomen dat elk individu uit elke strategie interacteert met elk individu uit elke andere strategie met gelijke kans. Deze aanname is onrealistisch.

   (d) De variantie van de gemiddelde opbrengst per episode is te groot, zodat de uitkomsten van een willekeurig uitgevoerd experiment insignificant zullen zijn.

7. Bekijk Fig. 1.

   (a) ALL-C = $\Box$, ALL-D = $\triangle$, Random = $\diamond$, TFT = $\times$, Pavlov= $+$.

   (b) ALL-C = $\triangle$, ALL-D = $\diamond$, Random = $\Box$, TFT = $\times$, Pavlov= $+$.

   √ ALL-C = $\diamond$, ALL-D = $\triangle$, Random = $\Box$, TFT = $+$, Pavlov= $\times$.

   (d) ALL-C = $\Box$, ALL-D = $\triangle$, Random = $\diamond$, TFT = $+$, Pavlov= $\times$.

   Antw. Zie Flake. $\Box$

8. Bekijk het volgende binaire classificatie-probleem. Mogelijke leer-instanties:

$$\mathcal{D} = \{(m, n) \in N \times N \mid 0 \leq m \leq 40 \text{ en } 0 \leq n \leq 30\}$$

   Mogelijke classificaties: $+$ en $-$. Hypotheseruimte:

$$\mathcal{H} = \{I \times J \mid I \text{ is een } t_{40}\text{-rij en } J \text{ is een } t_{30}\text{-rij}\}$$

   Een t_K*-rij* is een niet-lege verzameling van opeenvolgende gehele getallen, waarvan het eerste getal niet-negatief is, en het laatste getal niet groter is dan K.

   Gegeven zijn de leervoorbeelden

$$\begin{array}{llll} L = \{ & (4,4):+ & (5,7):+ & (6,9):+ \\ & (7,6):+ & (9,7):- & (9,9):+ \\ & (10,5):+ & (10,12):+ & (11,7):+ \\ & (11,10):- & (11,15):+ & (14,12):+ \quad \} \end{array}$$

Er worden twee hypothesen voorgesteld, te weten

$$\begin{aligned} H_1 &= \{9,\ldots,15\} \times \{11,\ldots,15\} \text{ en} \\ H_2 &= \{4,\ldots,12\} \times \{3,\ldots,9\}. \end{aligned}$$

Bereken van elk van deze hypothesen het bereik (range), de overdekking (coverage), de match, en de nauwkeurigheid (accuracy).

√ Range: 35 en 63; coverage: 3 en 7; match: 0.42 en 0.67; accuracy: 1.0 en 0.88.

(b) Range: 3 en 7; coverage: 35 en 63; match: 0.42 en 0.67; accuracy: 1.0 en 0.88.

(c) Range: 3 en 7; coverage: 35 en 63; match: 1.0 en 0.88; accuracy: 0.42 en 0.67.

(d) Range: 35 en 63; coverage: 3 en 7; match: 1.0 en 0.88; accuracy: 0.42 en 0.67.

Antw. Je komt al snel bij het goede antwoord door alleen al naar de range te kijken. □

9. We passen kwalitatief gesuperviseerd leren met vector-kwantisatie toe op drie cluster-representanten $A = (1, 77, 1)$, $B = (2, 77, 3)$ en $C = (3, 77, 0)$. Bepaal de bewegingen van deze cluster-representanten als het leervoorbeeld $l = (2, 77, 2) \to A$ wordt aangeboden.

(a) Er gebeurd niets.

(b) Repr-A beweegt naar l toe.

√ Repr-A beweegt naar l toe en Repr-B beweegt van l af.

(d) Repr-A beweegt naar l toe en Repr-B en Repr-C bewegen van l af.

Antw. Letterlijke definitie van het algoritme. Merk op dat de tweede dimensie $y = 77$ kan worden weggegooid, zodat je het voorbeeld in R^2 kunt tekenen. □

10. We leren voorbeelden in R^3 met een standaard twee-dimensionaal Kohonen-netwerk. Het netwerk is random geïnitialiseerd binnen het bolletje om $(0, 0, 0)^T$ met straal 0.01. De leervoorbeelden $1 - 1,000$ bevinden zich uniform verdeeld over het oppervlak S van de eenheidsbol B. De leervoorbeelden $1,001 - 20,000$ bevinden zich uniform verdeeld over het oppervlak S' van de eenheidsbol B' die ontstaat door B te transleren over de vector $(5, 0, 0)^T$. De leerfactor is 0.4.

We kunnen het volgende zeggen over het resulterende netwerk.

√ Het spant zich om B' heen.

(b) Het spant zich, deels om B, en deels om B' heen.

(c) Het bevindt zich tussen B en B'.

(d) Het spant zich deels over het half-rond van B' dat naar B wijst, en deels over het half-rond van B dat naar B' wijst.

Antw. 95% extra iteraties om B' heen zorgt er wel voor dat het hele net om B' heen gaat zitten. B wordt vergeten. □

11. K-means clustering is van nature een batch-leertechniek.

(i) Maar, als er later voorbeelden bijkomen kan K-means daar wel mee omgaan.

(ii) Maar, als er later nieuwe cluster-centra bijkomen kan K-means daar wel mee omgaan.

Met "mee omgaan" bedoelen we dat de convergentie even verstoord zal worden, maar daarna weer zal herstellen tot een nieuw en waarschijnlijk anders optimum.

√ (11i) en (11ii).

(b) (11i) en niet (11ii).

(c) niet (11i) en (11ii).

(d) niet (11i) en niet (11ii).

12. Een probleem van recursieve clustering-methoden is de initialisatie: het kan gebeuren dat een neuron nooit winnaar wordt en dus niet leert.

De volgende wijzigingen worden gesuggereerd.

(a) Initialiseer elk neuron op een inputpatroon.

(b) Verander de gewichten van het winnende neuron k met de volgende leerregel:

$$w_k(t+1) = \frac{w_k(t) + \gamma(x(t) - w_k(t))}{\|w_k(t) + \gamma(x(t) - w_k(t))\|}$$

(c) Gebruik zg. "leaky learning".

(d) Voeg random noise toe aan de leerparameter γ.

Welke van deze wijzigingen helpen dit probleem op te lossen?

(a) (12a) en (12b).

(b) (12b) en (12c).

(c) (12c) en (12d).

√ Het goede antwoord staat er niet bij.

13. Marie beschikt over $1,000$ voorbeelden om haar neurale netwerk te laten leren. Wat is de meest verstandige aanpak?

(a) Laat maar zitten. Duizend voorbeelden is überhaupt te weinig om een netwerk significant te laten leren.

(b) Herhaaldelijk laten leren over alle $1,000$ voorbeelden. Stoppen als alle netwerkgewichten niet of nauwelijks meer veranderen.

(c) Herhaaldelijk laten leren op de eerste 900 voorbeelden. De laatste 100 voorbeelden gebruiken als testvoorbeelden. Stoppen met leren als de performance over de testvoorbeelden gaat dalen.

√ Voorbeelden opdelen in tien groepen. Dan tien keer laten leren over negen groepen met telkens één andere groep als contrôle. Telkens stoppen met leren als de performance over de testvoorbeelden gaat dalen. Run vervolgens een elfde keer met het gemiddeld aantal runs van alle vorige tien keren.

Antw. Zoals bediscussieerd in Chapt. 4, Artificial Neural Networks, Mitchell. □

14. Waar word de hypotheseruimte in een neuraal netwerk door vertegenwoordigt?

√ Door de gewichten.

(b) Door de netwerk-topologie, d.w.z., de knopen en de links.

(c) Door de topologie en de gewichten.

(d) Door de topologie, en de respectievelijke leervoorbeelden.

Antw. Alleen de gewichten, de rest ligt tijdens het leren vast. □

15. (Backpropagation.) Voor eindknopen geldt:

$$-\frac{\partial E_d(\vec{w})}{\partial net_j} = (t_k - o_k)\, ok(1 - o_k).$$

Deze factor korten we af met δ_k. We bepalen de correctiefactor δ_j voor inwendige knopen. Omdat net_j alleen via $Downstream(j)$ invloed heeft op E_d mogen we schrijven

$$\begin{aligned}
\delta_j &= -\frac{\partial E_d(\vec{w})}{\partial net_j} \\
&= \sum_{k \in Downstream(j)} -\frac{\partial E_d(\vec{w})}{\partial net_k}\frac{\partial net_k}{\partial net_j} \\
&= \sum_{k \in Downstream(j)} \delta_k \frac{\partial net_k}{\partial net_j} \\
&= \sum_{k \in Downstream(j)} \delta_k \frac{\partial net_k}{\partial o_j}\frac{\partial o_j}{\partial net_j} \\
&= \sum_{k \in Downstream(j)} \delta_k x_{kj} o_j(1 - o_j) \\
&= o_j(1 - o_j) \sum_{k \in Downstream(j)} \delta_k x_{kj}.
\end{aligned}$$

(a) De net_k moet een $\sigma(net_k)$ zijn.

(b) Op $\partial net_k/\partial net_j$ mag de kettingregel niet worden toegepast om $\partial net_k/\partial o_j \cdot \partial o_j/\partial net_j$ te krijgen.

√ De x_{kj} moet een w_{kj} zijn.

(d) Omdat $o_j(1 - o_j)$ afhangt van k is de laatste herschrijving niet toegestaan.

16. In reinforcement leren is value-iteration een techniek om via iteratie grootheden aan te passen. Daarbij gelden de volgende beweringen.

(i) V verandert.

(ii) Q verandert.

(iii) Π verandert.

√ (16i), (16ii) en (16iii).

(b) (16i), (16ii) en niet (16iii).

(c) (16i), niet (16ii) en (16iii).

(d) (16i), niet (16ii) en niet (16iii).

Einde van alle opgaven. Heb je je antwoorden gecontroleerd? Op de vaksite verschijnen deze week de uitwerkingen. Deze opgaven mogen worden meegenomen.

Vergeet je niet de vak-evaluatie in te vullen? Bedankt voor je deelname en misschien tot ziens! ♠

Figuren

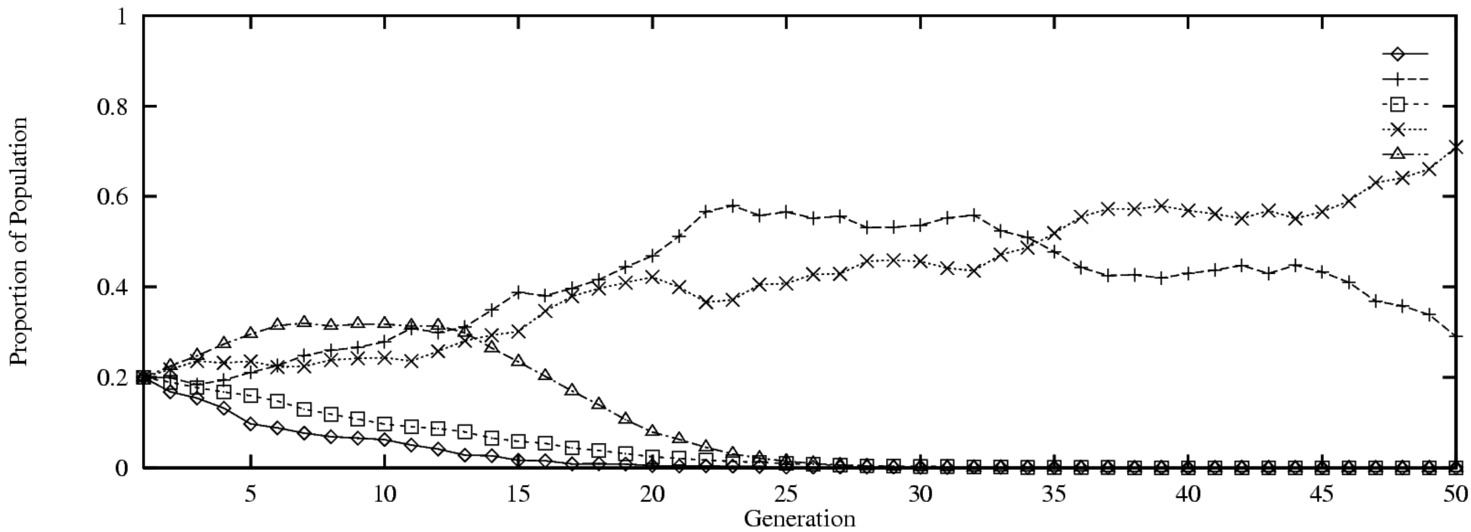

Fig. 1: Ecologische versie van het IPD, met 0.5% noise.

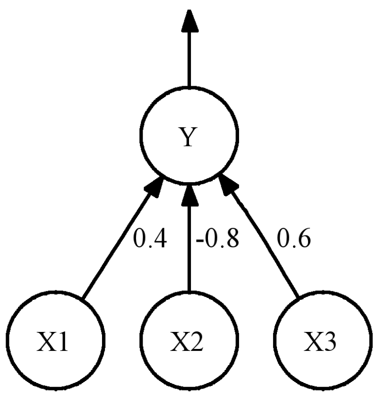

Fig. 2: Lineair perceptron. Input $X_3 \equiv 1$.

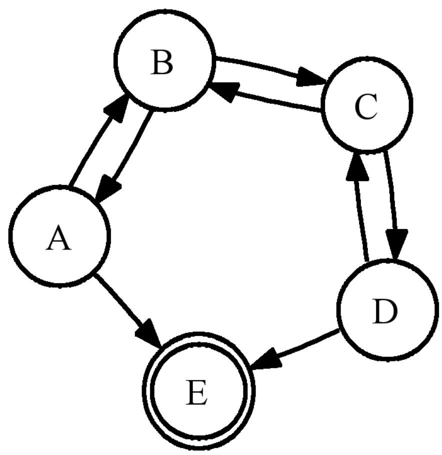

Fig. 3: Toestandengraaf.

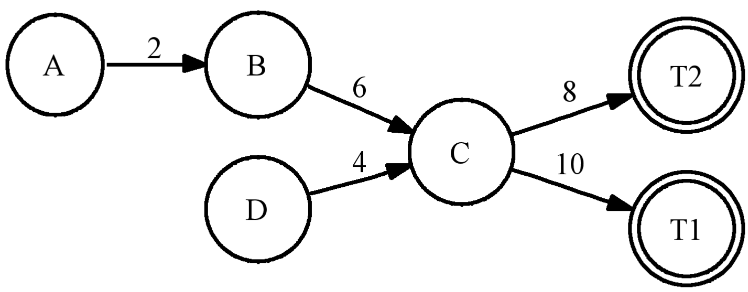

Fig. 4: Twee epochs.