

# Inleiding Adaptieve Systemen

deel 2, 25 juni 2014, 13.30-16.30, v. 1

Er is op vrijdag 27 juni nog een practicumssessie!

De aanvullende toets is op 4 juli, 13-15 uur.

## Competitie en cooperatie

1. Bekijk de volgende uitbetalings-matrix van een 2-player game:

	C	D
C	(5, 5)	(4, 7)
D	(7, 4)	(1, 1)

Dit spel laat zich analyseren als

- (a) Prisoner's Dilemma
- (b) Stag Hunt
- (c) • Chicken
- (d) Battle of the Sexes

★ antwoord:  $T > R > S > P$ , dus Chicken (slideset 9<sup>2014</sup>, slide 28)

2. De Tit-for-Tat strategie in het Iterated Prisoner's Dilemma houdt in:

- (a) begin met C, en wissel de keuze nadat de tegenstander D speelt
- (b) • begin met C, en kopieer daarna steeds de vorige zet van de tegenstander
- (c) begin met C, maar blijf D spelen als de tegenstander een keer D speelt
- (d) begin met C, maar speel D als de tegenstander op de vorige zet C speelde

★ antwoord: a is Pavlov, c is Unforgiving, d is een bizarre variant van TFT: afstraffen als de tegenstander juist aardig is.

3. We spelen een toernooi met meerdere kopieën van diverse strategieën voor het Iterated Prisoner's Dilemma: All-C, All-D, Tit-for-Tat, Pavlov, en Random. Na afloop veranderen we de populatie afhankelijk van het

succes. Door steeds opnieuw toernooien te blijven spelen evolueert de populatie.

We kunnen dit doen onder 'ideale' omstandigheden, of met introduceren van een kleine hoeveelheid 'ruis' (noise).

Welke strategie gaat de populatie domineren (en blijft soms zelfs als enige over)?

- (a) ideaal: All-C; met ruis: Pavlov
- (b) ideaal: All-C; met ruis: Random
- (c) ideaal: All-D; met ruis: Tit-for-Tat
- (d) • ideaal: Tit-for-Tat; met ruis: Pavlov

★ antwoord: Zie Flake, blz. 300. In moreel opzicht is All-C wel *idealistisch* te noemen, maar dat wil nog niet zeggen dat het in een ideale (ruis-vrije) wereld overleeft...

## Neurale netwerken

4. Bij het trainen van een multi-layer feedforward network met behulp van het backpropagation algoritme kunnen de gewichten het best worden geïnitieerd met:

- (a) Alle gewichten gelijk aan 1.
- (b) Alle gewichten gelijk aan 0.
- (c) Alle gewichten uniform random gekozen in het interval  $[-5, +5]$ .
- (d) • Alle gewichten uniform random gekozen in het interval  $[-0.5, +0.5]$ .

5. De optimale gewichten voor een lineaire perceptron kunnen berekend worden door middel van multivariate lineaire regressie. De  $N$  input voorbeelden  $X^1, X^2, \dots, X^N$  vormen de kolommen van de inputmatrix  $X$ :

$$X = [X^1, X^2, \dots, X^N].$$

De bijhorende outputs  $Y^1, Y^2, \dots, Y^N$  vormen de kolommen van de outputvector  $Y$ :

$$Y = [Y^1, Y^2, \dots, Y^N].$$

Voor de gewichtenvector  $W$  geldt:

$$Y = W^T X.$$

Geef aan hoe de gewichten berekend kunnen worden in termen van  $X$  en  $Y$ :

$$W^T X = Y \Leftrightarrow W^T = \dots\dots$$

★ antwoord: Je zou misschien links en rechts door  $X$  willen delen, maar dat kan niet want  $X$  is een matrix en daar kun je niet door delen. Een beter idee is al om links en rechts met  $X^{-1}$  te vermenigvuldigen:  $XX^{-1}$  is de identiteitsmatrix, dus dan is de gewenste oplossing  $W^T = YX^{-1}$  verkregen. Helaas, dat kan ook niet zomaar, want  $X$  is geen vierkante matrix en heeft dus niet een inverse.

De briljante ingeving staat in slideset 12 op slide 15: eerst links en rechts met  $X^T$  vermenigvuldigen. Nu staat er links  $XX^T$ , en dat is wèl een vierkante matrix die je kunt inverteren:

$$\begin{aligned} W^T X &= Y \\ \Leftrightarrow W^T X X^T &= Y X^T \\ \Leftrightarrow W^T &= Y X^T (X X^T)^{-1} \end{aligned}$$

6. Voor het supervised leren van een feedforward neurale netwerk beschikken we over 500 voorbeelden. Wat is de meest geschikte strategie van onderstaande opties:

- (a) Minimaliseer de error op de 500 voorbeelden. Rapporteer de gewichten met de laagste error.
- (b) Minimaliseer de error op de 500 random gekozen voorbeelden (random keuze met teruglegging, dus dubbele keuzes zijn toegelaten). Rapporteer de gewichten met de laagste error op opnieuw 500 random gekozen voorbeelden (eveneens random keuze met teruglegging).

(c) Verdeel de data in 300 trainingsvoorbeelden en 200 testvoorbeelden. Minimaliseer de error op de trainingsvoorbeelden. Neem de gewichten met de laagste error op de *trainings*voorbeelden en rapporteer hun fout op de testvoorbeelden.

(d) • Verdeel de data in 300 trainingsvoorbeelden en 200 testvoorbeelden. Minimaliseer de error op de trainingsvoorbeelden. Neem de gewichten met de laagste error op de *test*voorbeelden en rapporteer hun fout op de testvoorbeelden.

★ antwoord: (Zie Mitchell, pagina 111)

7. We willen online stochastische gradient descent toepassen op een lineair neuron met output  $Y_d$ , target  $T_d$ , input  $X_{d,i}$  en gewichten  $w_i$  voor het voorbeeld  $d$ :  $Y_d = \sum_i w_i X_{d,i}$ . De error is  $E_d(\vec{w}) = \frac{1}{2}(T_d - Y_d)^2$ . Geef aan hoe de gewichten update  $\Delta w_i$  berekend kan worden uit  $T_d$ ,  $Y_d$  en  $X_{d,i}$ :

$$\Delta w_i = -\alpha \frac{\partial E_d(\vec{w})}{\partial w_i} = \dots\dots$$

Hint: je kunt niet direct de afgeleide van  $E_d$  naar  $w_i$  bepalen, maar wel de afgeleide van  $E_d$  naar  $Y_d$ , en de afgeleide van  $Y_d$  naar ...

★ antwoord: staat in slideset 12 op slide 21:

$$\begin{aligned} \Delta w_i &= -\alpha \frac{\partial E_d(\vec{w})}{\partial w_i} \\ &= -\alpha \frac{\partial E_d(\vec{w})}{\partial Y_d} \frac{\partial Y_d}{\partial w_i} \\ &= -\alpha (T_d - Y_d) (-1) X_{d,i} \\ &= +\alpha (T_d - Y_d) X_{d,i} \end{aligned}$$

8. Gegeven is een 1-dimensionaal Kohonen netwerk met buurfunctie:

$$g(x, y) = \begin{cases} 1 & \text{als } |x - y| = 0, \\ 0.5 & \text{als } |x - y| = 1, \\ 0 & \text{anders.} \end{cases}$$

Het netwerk ligt in  $R^2$  zoals weergegeven in de figuur. Bepaal de ligging van dit netwerk na updates met de volgende twee voorbeelden, waarbij de leerfactor,  $\alpha$ , gelijk aan 0.25 is. Schrijf de tussenresultaten van je berekening op in de tabel op het antwoordblad. De voorbeelden worden aangeboden in deze volgorde: voorbeeld 1: (0, 4), en voorbeeld 2: (3, 3).

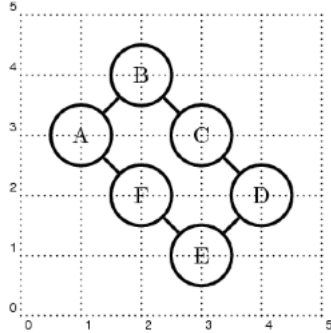


Fig. 1: Kohonen netwerk.

★ antwoord: Bij het eerste datapunt is  $A$  de knoop die het dichtst bij het datapunt staat. Dus  $A$  kruipt  $\alpha = 0.25$  van de afstand tot het datapunt naar dat datapunt toe. De buur-knopen  $B$  en  $F$  kruipt ieder  $0.5\alpha$  in de richting van het datapunt.

Het tweede datapunt valt samen met knoop  $C$ . Die hoeft zelf dus niet meer te kruipten, maar sleept wel zijn buur-knopen  $B$  en  $D$  mee.

	$A$	$B$	$C$
0	(1, 3)	(2, 4)	(3, 3)
na vb. 1	$(\frac{3}{4}, 3\frac{1}{4})$	$(1\frac{3}{4}, 4)$	(3, 3)
na vb. 2	$(\frac{3}{4}, 3\frac{1}{4})$	$(1\frac{29}{32}, 3\frac{7}{8})$	(3, 3)
	$D$	$E$	$F$
0	(4, 2)	(3, 1)	(2, 2)
na vb. 1	(4, 2)	(3, 1)	$(1\frac{3}{4}, 2\frac{1}{4})$
na vb. 2	$(3\frac{7}{8}, 2\frac{1}{8})$	(3, 1)	$(1\frac{3}{4}, 2\frac{1}{4})$

9. Elman Neurale Netwerken (ENNs) zijn recurrente netwerken waarvoor geldt:

- (a) ENNs koppelen de activatie van output neuronen terug naar input neuronen.
- (b) • ENNs koppelen de activatie van hidden neuronen terug naar input neuronen.
- (c) ENNs gebruiken inputs van vorige tijdstappen als huidige input neuronen.
- (d) ENNs koppelen alle neuronen in het netwerk terug naar elkaar.

10. Welke uitspraak is waar bij batch vs. online gradient descent van de gewichten van een neural network:

- (a) Online leren volgt precies de fout-functie.
- (b) • Batch leren volgt precies de fout-functie.
- (c) Online en batch leren volgen beiden precies de fout-functie.
- (d) Online en batch leren volgen beiden *niet* precies de fout-functie.

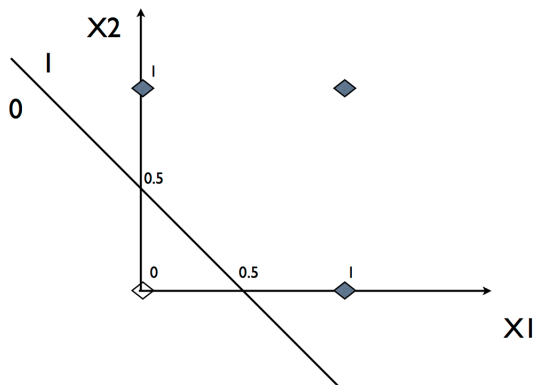
11. We passen kwalitatief gesuperviseerd leren met vector-kwantisatie (LVQ) toe op drie cluster-representanten:  $A = (0,0)$  met label Wit,  $B = (2,4)$  met label Zwart, en  $C = (3,2)$  met label Zwart. Bepaal de bewegingen van deze cluster-representanten als het leervoorbeeld  $X = (1, 1)$  met label Zwart wordt aangeboden. Gebruik de Euclidische afstandsmaat.

- (a)  $A$  beweegt naar  $X$  toe.
- (b)  $C$  beweegt naar  $X$  toe.
- (c) •  $C$  beweegt naar  $X$  toe en  $A$  beweegt van  $X$  af.
- (d)  $B$  en  $C$  bewegen naar  $X$  toe en  $A$  beweegt van  $X$  af.

12. Bereken de gewichten en bias van de lineaire perceptron met inputs  $X_1, X_2$  en output  $Y$  die het OR-probleem, zoals beschreven in

onderstaande tabel en figuur, correct classificeert

$X_1$	$X_2$	$Y$
0	0	0
0	1	1
1	0	1
1	1	1



★ antwoord: (zie Mitchel, blz. 87) Kies  $w_1$  en  $w_2$  positief, het maakt niet eens uit wat de waarde precies is. Kies de bias negatief, zodat de uitkomst negatief wordt bij  $X_1 = X_2 = 0$ , maar wel zo dat de uitkomst positief blijft zodra  $X_1 = 1$  of  $X_2 = 1$ . Bijvoorbeeld:  $w_1 = w_2 = 1$  en bias =  $-0.5$ . Maar ook  $w_1 = w_2 = 0.5$  en bias =  $-0.3$  is een mogelijkheid, en vele andere.

### Reinforcement learning

13. Beschouw een reinforcement leerprobleem met toestanden  $A, B, C, T_1$  en  $T_2$ , waarbij  $T_1$  en  $T_2$  eindtoestanden zijn. Er worden twee runs uitgevoerd, resulterend in  
 Epoch 1 =  $A, B, C, T_2$  en  
 Epoch 2 =  $A, B, C, A, B, T_1$ .

De directe rewards  $R(X, Y)$  die worden geobserveerd bij een toestandsovergang van toestand  $X$  naar toestand  $Y$  zijn als volgt:

$$\begin{aligned} R(A, B) &= -1 & R(B, T_1) &= 4 \\ R(B, C) &= -1 & R(C, T_2) &= 8 \\ R(C, A) &= -1 & & \end{aligned}$$

Na temporal difference learning met verdisconteringsfactor  $\gamma$  en leerfactor  $\alpha$  hebben de toestanden de volgende waarden (de waarden bij Epoch 2 zijn tevens de eindwaarden):

	$A$	$B$	$C$	$T_1$	$T_2$
Epoch 1	-0.8	-0.8	6.4	0	0
Epoch 2	-0.416	3.52	-0.032	0	0

Welke van onderstaande uitspraken met betrekking tot de gebruikte parameters is correct?

- (a)  $\alpha = 1.0$  en  $\gamma = 0.8$
- (b)  $\alpha = 0.5$  en  $\gamma = 0.8$
- (c)  $\bullet \alpha = 0.8$  en  $\gamma = 0.5$
- (d)  $\alpha = 0.8$  en  $\gamma = 1.0$

14. Welke van onderstaande uitspraken over de Markov eigenschap is correct?

De Markov eigenschap...

- (a) schrijft voor in welke toestand je welke actie moet uitvoeren
- (b)  $\bullet$  zegt dat, gegeven het heden, de toekomst onafhankelijk is van het verleden
- (c) zegt dat wanneer je actie  $a$  uitvoert in toestand  $s$ , elke toestand  $s'$  met een bepaalde kans bereikt kan worden
- (d) wordt in reinforcement leren alleen door modelgebaseerde (indirecte) algoritmen gebruikt

★ antwoord: (b) is correct, het verwoordt de formule  $P(s_{t+1} | s_t, a_t) = P(s_{t+1} | s_t, a_t, \dots, s_1, a_1)$ .

- (a) beschrijft niet de Markov eigenschap maar een policy in een deterministische wereld
- (c) zou je kunnen geven als je nog wel vaag iets van de formule weet, maar niet meer van de betekenis ervan.
- (d) is niet waar omdat bijv temporal difference learning door het meenemen van alleen kennis over huidige en volgende toestand ook de Markov property uitbuit; dit

geldt niet voor Monte Carlo sampling: die neemt de hele toekomst mee.

15. Welke van onderstaande uitspraken is *niet* correct?

- (a) Monte Carlo sampling is alleen toepasbaar voor eindige trials/runs
- (b) Q-learning is een vorm van temporal difference learning
- (c) • Reinforcement learning algoritmen lossen de Bellman vergelijking op
- (d) Met Q-learning kan, uiteindelijk, een *optimale* policy bepaald worden

★ antwoord: (a) is waar omdat je in de berekeningen steeds meeneemt hoeveel beloning je nog krijgt tot het einde van een epoch; dit is de daadwerkelijk te verkrijgen beloning in het epoch en niet de te verwachten beloning.

(b) is waar: je doet ipv hetzelfde als bij TD, alleen met Q ipv V

(c) is niet waar: als we al voldoende informatie hebben op de vergelijking op te lossen (een model) dan is dat doorgaans te complex; RL algoritmen benaderen daarom de Q-functie. Overigens is het bij Q-learning wel zo dat als alle toestand/actie paren oneindig vaak bezocht worden, de Q-functie convergeert naar de optimale Q-functie en gegeven de optimale Q-functie kunnen we ook een optimale policy vaststellen; vandaar dat antwoord (d) juist wel waar is.

### Leren van regels

16. Er is een verschil tussen het leren van één regel of het leren van meerdere regels voor een gegeven data-set. Welk van de volgende verschillen is correct?

- (a) bij het leren van één regel is er een grotere kans op inconsistenties
- (b) bij het leren van meerdere regels is het lastiger om te gaan met ruis

(c) • bij het leren van meerdere regels kun je een hogere accuratesse bereiken

(d) bij het leren van één regel heb je een kleiner bereik

17. Welk van de volgende beweringen over overfitting bij het leren van regels is *niet* correct?

(a) overfitting is een algemeen probleem bij leertechnieken binnen de AI

(b) • overfitting is alleen een probleem bij supervised learning techniques

(c) overfitting kan worden tegengegaan door bij heuristisch zoeken de accuratesse van hypothesen te verlagen

(d) overfitting kan worden tegengegaan regels te zoeken van een logische taal met minder uitdrukingskracht

18. Welk van de volgende beweringen over het heuristisch zoeken naar regels is *niet* correct?

(a) • bij heuristisch zoeken naar regels weet je nooit zeker of een bestaande consistente oplossing ook zal worden bereikt

(b) heuristisch zoeken naar regels is een betere afspiegeling van hoe de mens regels induceert (bijvoorbeeld in de natuurkunde, bij het herkennen van patronen, bij het inschatten van strategieën van anderen, etc.) dan niet-heuristisch zoeken

(c) heuristisch zoeken van regels kan omgaan met inconsistente data

(d) heuristisch zoeken resulteert vaak in een bredere zoekboom dan niet-heuristisch zoeken