

Bij iedere open vraag is aangegeven hoeveel punten u met die vraag kunt behalen. In totaal kunnen de open vragen 75 punten opleveren.

De 10 multiple-choice vragen kunnen maximaal 25 punten opleveren. Een goed antwoord verhoogt uw score, een fout (of geen) antwoord verlaagt uw score niet. Vul dus altijd een antwoord in. De antwoorden op de multiple-choice vragen vult u in in de tabel achteraan. Als u een multiple-choice antwoord wilt verbeteren kruist u het foute antwoord door en vult u daarachter het goede antwoord in.

Uw cijfer is het totaal aantal punten gedeeld door 10, met een minimum van 1.0.

Mogelijk zijn delen van de vraag **vet** gedrukt of gemarkeerd met *sterretjes*. Het is aan te raden daar extra op te letten.

Als uw antwoord code bevat zal dit niet beoordeeld worden op 'kleine' syntaxfouten, d.w.z. die fouten die door de compiler gesignaleerd zouden worden, en waarvan de verbetering vanzelf spreekt.

Als een vraag (of bij multiple choice ook: een of meer antwoorden) volgens u een fout bevatten die u hinderen bij het geven van uw antwoord schrijf dit dan op.

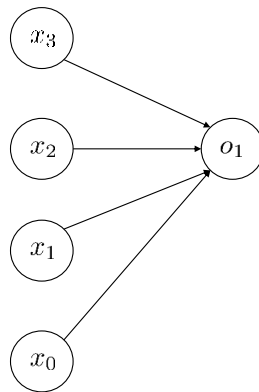
Het importeren van benodigde Python packages (bijv. functools, inspect, etc.) hoeft niet expliciet aangegeven te worden.

1) [25 punten] NN Topologie

Ontwerp op papier een neurale netwerk voor het uitvoeren van de volgende functie:

Input 1	Input 2	Input 3	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

- Geef aan hoeveel lagen en neuronen je netwerk nodig heeft (en waarom).
- Teken het netwerk.
- Geef de waarden van de gewichten, zodat je netwerk de bovenstaande functie perfect modelleert (d.w.z., als je het netwerk de waarden van 'input x' van regel uit de tabel presenteert, berekend het netwerk de waarde gegeven bij 'output' van die regel).

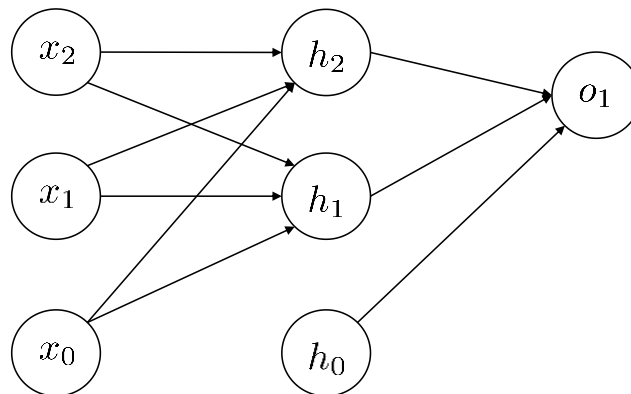


	x_0	x_1	x_2	x_3
o_1	1	-0.5	-0.5	-0.5

Géén hidden laag vereist omdat functie lineair schijdbaar is (dus op te lossen met een SLP).

2) [25 punten] Leeralgoritmen

Gegeven is het volgende neurale netwerk:



Met gewichten:

	x_0	x_1	x_2
h_1	-0.1	0.4	-0.2
h_2	0.2	0.3	0.1

	h_0	h_1	h_2
o_1	0.3	0.1	0.2

En activatiefunctie ReLU (zie onder).

- Bereken de uitkomst van het netwerk gegeven de input: $x_1 = 1, x_2 = 1$. De bias neuronen (met index 0) geven altijd een waarde 1.
- Bereken de nieuwe waarden van de gewichten gegeven dat de uitkomst bij de input van hierboven ($o_1 = 1$) had moeten zijn.

Gebruik hiervoor:

Delta regel	$w_{i,j} = w'_{i,j} + \eta a_i g'(in_j)(y_j - a_j)$
Backpropagation	$w_{j,k} = w'_{j,k} + \eta a_j \Delta_k$ $\begin{cases} \Delta_k = g'(in_k)(y_k - a_k) & \text{(voor output neuron)} \\ \Delta_k = g'(in_k) \sum_p w_{k,p} \Delta_p & \text{(voor hidden neuron)} \end{cases}$
ReLU	$g(x) = \max(0, x)$
Afgeleide van ReLU	$g'(x) = \begin{cases} 0 & \text{als } x \leq 0 \\ 1 & \text{als } x > 0 \end{cases}$

	x0	x1	x2			h0	h1	h2
h1	-0,1	0,4	-0,2		o1	0,4	-0,1	0,4
h2	0,2	0,3	0,1					
x0	1		h0	1				
x1	1		h1	0,3	o1	0,57		
x2	0		h2	0,5				
d-o-1	-0,57			eta	0,1			
d-h-1	0,057							
d-h-2	-0,228							
	x0	x1	x2			h0	h1	h2
h1	-0,0943	0,4057	-0,2		o1	0,343	-0,1171	0,3715
h2	0,1772	0,2772	0,1					

3) [25 punten] Genetisch algoritmen

We willen een tafelverdeling maken voor het diner van de volgende HBO-ICT teamdag, maar helaas is het maken van een goede indeling een lastig probleem. Iedere docent wil natuurlijk bij zijn/haar partner aan tafel zitten, maar niet bij docenten waarmee hij/zij sterk van mening verschilt (bijv., geen BIM-ers bij TI-ers), maar juist weer wel bij die waarvan hij de meningen deelt.

Het gezelschap bestaat uit zo'n 80 mensen (docenten en partners) en er zijn 10 tafels (met elk 8 plekken) te verdelen.

Maak een representatie voor dit probleem om het door middel van een genetisch algoritme op te lossen. Je mag ervanuit gaan dat er een geschikte fitness functie gegeven is die voor een tafel kan bepalen hoe goed die indeling is.

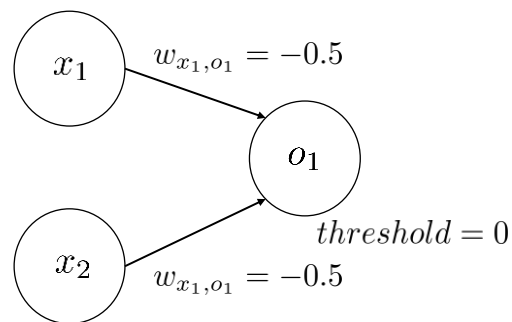
- a) Ontwerp een representatie voor dit probleem (encoding). Hoe ziet je genotype eruit?
- b) Wat raad je aan (motiveer!) als selectiestrategie en recombinate operators. Wat zijn de belangrijke eigenschappen van beiden (met betrekking tot het oplossen van dit probleem)? Waarom zijn die keuzes noodzakelijk?

Tafelschikkingen zijn een ordering probleem;

- Beschouw je genotype (chromosoom) als een reeks van individuen, waarbij de index in de reeks een stoel(nummer) aanduidt. Bijv. index 0 is de eerste stoel van de eerste tafel, index 1 is de tweede stoel van de eerste tafel, ..., index 8 is de eerste stoel van de tweede tafel, etc.
- Iedere persoon (docent of partner) wordt gelabeld (met een nummer of met een letter, bijv. de eerste twee letters van de achternaam, plus een 'd' voor docent, en 'p' voor de partner van een docent).
- Er zijn geen specifieke eisen voor de selectiestrategie. Ze zal echter, zoals altijd, voldoende selectiedruk en genetische diversiteit moeten leveren om het algoritme optimaal te laten werken. Hierom kies ik voor Tournament selection (beste genetische diversiteit); om de selectie druk op te voeren, spelen we 4-speler toernoois (75%, 19%, 5%, 1%). Individuen mogen meer dan 1 toernooi spelen.
- Wat betreft mutatie en crossover moeten we uiteraard een ordering-specifiek operator kiezen. Aangezien we per tafel scoren, heeft het niet zoveel zin om stukken van het genotype te verplaatsen, inverteren of te scramblen, dit heeft weinig effect op de (per tafel) schikking. Insertion (1 persoon bij een tafel weghalen en bij een andere tafel plaatsen) heeft echter wel een duidelijke invloed op het genotype. We kiezen daarom daarvoor.
- Ook wat crossover betreft zullen we de volgorde moeten bewaren. In principe werken order-based crossover en position-based crossover ongeveer hetzelfde. Wat onze tafelschikkingen betreft is het waarschijnlijk handiger om bepaalde personen per tafel over te erven naar de kinderen; position-based crossover bereikt dit.
- Crossover is in dit geval niet noodzakelijk, het versneld alleen de convergentie (exploitatie); het algoritme werkt ook prima met enkel mutatie.

→ Voer je antwoorden in op de laatste bladzijde, dat is de lijst die wordt nagekeken.

1) Welke functie wordt benaderd door het onderstaande perceptron?



- A: AND
- B: NOR
- C: OR
- D: NAND

2) Wat is de functie van de bias in feed-forward netwerken?

- A: Glad/effen maken van de activatiefunctie.
- B: Verbeteren van het lerend vermogen door gebruik van backpropagation.
- C: Inhibitie van neuronen toevoegen, die ervoor zorgt dat neuronen meer input vereisen om 'aan' te gaan.
- D: Het verwijderen van de threshold in de activatiefunctie.

3) Waarom is het gebruik van een sigmoid-functie aan te raden als je NNen wilt trainen?

- A: De sigmoid activatiefunctie zorgt dat de uitvoer van een neuron beter past bij de inputs.
- B: Sigmoid neuronen zijn standaard in lerende NNs.
- C: Sigmoid activatie zorgt voor niet-lineaire scheiding van de output.
- D: Sigmoid activatiefuncties zorgen voor een continue output van neuronen.

4) Zet op volgorde van meest invloedrijk tot minst invloedrijk; welke van de volgende heeft de meeste invloed op de performance van KNN?

- A: Minder validatiedata.
- B: Minder trainingsdata.
- C: PCA om het aantal (relevante) attributen te reduceren.
- D: Introductie van prototypische individuen.

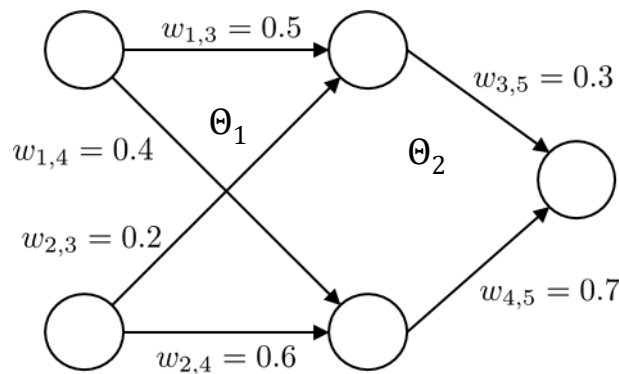
5) Wat maakt KNN zo gevoelig voor outliers?

- A: KNN beschouwd alle trainingsvoorbeelden bij het maken van een classificatie, fouten daarin hebben dus altijd een invloed op het algoritme.
- B: Outliers zijn vaak meetfouten, en *garbage in = garbage out*.
- C: Outliers verstoren de classification boundary, die daardoor grilliger van vorm wordt.
- D: Outliers vergroten de kans op overfitting.

6) Welke van de volgende eigenschappen geldt voor de clusters opgeleverd door KMeans?

- A: Polythetic.
- B: Soft borders.
- C: Hierarchical.

7) Gegeven het volgende neurale netwerk, welke code hieronder berekend correct de output van dit netwerk (gegeven een inputvector v)?



- A: `sigmoid(sigmoid(theta2*v)*theta2)`
- B: `sigmoid(theta2.dot(sigmoid(theta1.dot(v))))`
- C: `sigmoid(sigmoid(v*theta1)*theta2)`
- D: `sigmoid(sigmoid(v.dot(theta1)).dot(theta2))`

8) Welke van de volgende matrixberekeningen levert een foutmelding (wegens incorrecte dimensies)?

```
M = numpy.array([[1, 2, 3], [4, 5, 6]])
N = numpy.array([[3, 2, 1], [6, 5, 4]])
u = numpy.array([1, 2, 3])
```

- A: $M+N$
- B: `M.dot(u)`
- C: `M*u`
- D: `M.dot(N)`

9) Waarom zou je eerder voor Tournament selection kiezen dan voor Roulette wheel selection?

- A: Tournament selection levert een hogere genetische diversiteit.
- B: Tournament selection heeft een hogere selectie druk.
- C: Roulette wheel selection kan te vroeg convergeren.
- D: Roulette wheel selection is te random.

10) Wat is de belangrijkste reden om géén Evolutionair algoritme toe te passen?

- A: Probleem is slecht gedefinieerd, maar de vorm van de zoekruimte is bekend.
- B: Er zijn (meerdere) lokale optima.
- C: Fitness-functie is niet exact (noisy).
- D: De fitness-functie is gradueel steigend met een absoluut maximum.

001 : B
002 : D
003 : D
004 : C – D – B – A
005 : C

006 : A
007 : B
008 : D
009 : C
010 : D

Multiple Choice Vraag	Antwoord (schrijf duidelijk: A, B, C, of D)
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	