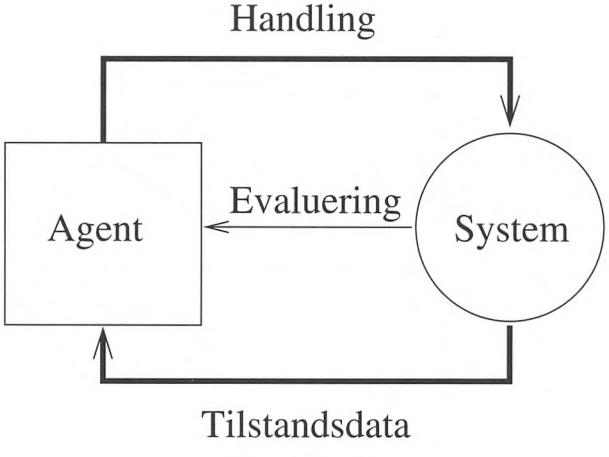
Hvordan lærer man en computer at køre på cykel?

Jette Randløv

Inden for kunstig intelligens er endemålet at forstå alle intelligente objekter og at kunne konstruere objekter med intelligens[1] – et meget ambitiøst mål, men modsat søgen efter overlyshastighedssignaler, anti-tyngdekraft, tidsrejser og evigt liv, kan vi være sikre på, at virkeligheden bakker os op. *Intelligens kan lade sig gøre.*

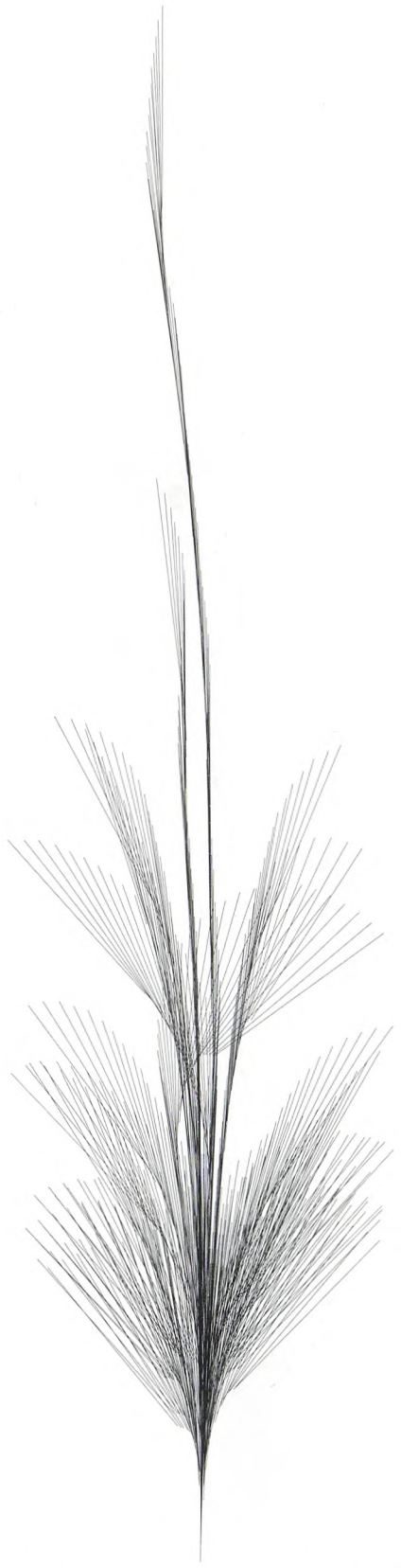
I løbet af de sidste 600 millioner år har biologiske systemer løst problemet med at vekselvirke med et foranderligt miljø ved at udvikle netværk af milliarder af forbundne nerveceller. Forsøget på at forstå disse processer trækker på viden fra mange felter – blandt andet matematik, fysik, datalogi, biologi og psykologi. Mange anvendelsesorienterede felter – maskinsyn, mønstergenkendelse, taleforståelse og robotsystemer – drager nytte af resultaterne. Ud af videnskaben om nervesystemer er der kommet et nyt computerteoretisk felt: Studiet af algoritmer, der kan bringe computere til at behandle data på en måde, som uden misbrug af ordet kan kaldes *indlæring*.



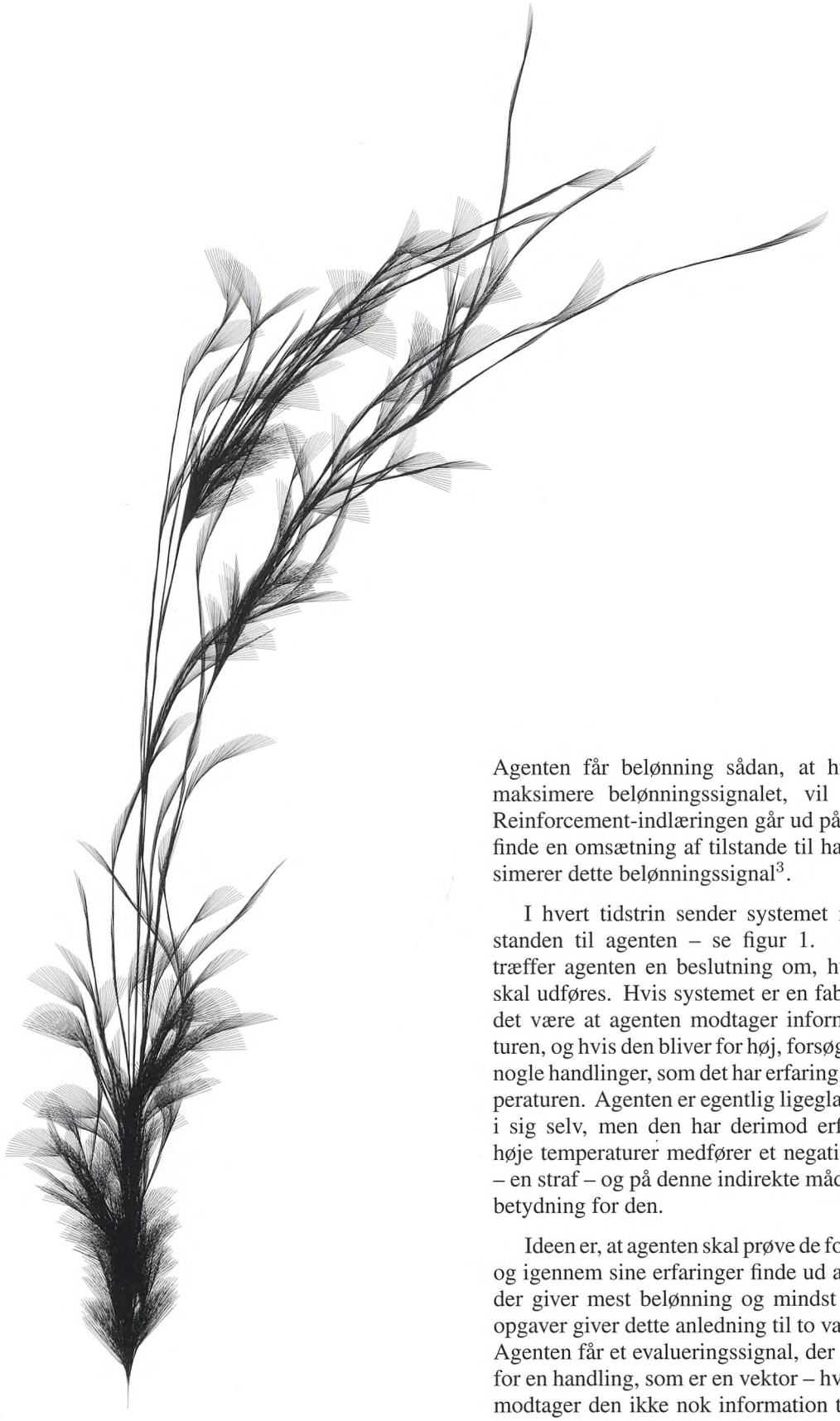
Figur 1: Informationsstrømmen mellem agenten og omgivelserne/systemet, som den skal handle i.

Reinforcement-indlæring

Reinforcement-indlæring er et lovende bud i viften af maskinindlæringsmetoder. Et centralt begreb er "agenten", som er en enhed, der kan sanse og handle. (Egentlig er en agent en matematisk abstraktion, men man kan med fordel tænke på den som en lille blikrobot, ikke ulig en af de gamle Nilfisk støvsugere, der suser rundt i et fase- eller tilstandsrum.) Ønsket bag reinforcement-indlæring er, at kunne programmere en agent til at løse en opgave gennem belønning og straf, men uden at fortælle den, hvordan opgaven skal løses.



Figur 2. Cyklens rute for de 151 første indlæringsrunder af en kørsel. Den længste cykletur er på 7 meter.

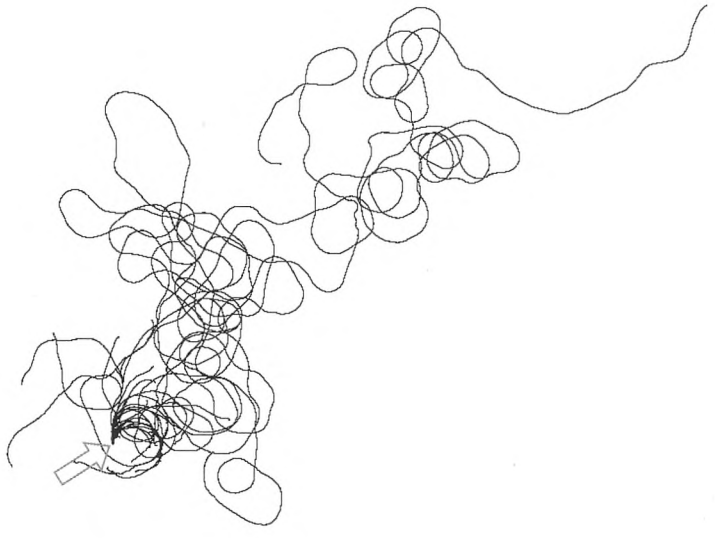


Figur 3. Cyklens rute noget tid senere, hvor agenten er i stand til at balancere på cyklen 30–40 meter af gangen.

Agenten får belønning sådan, at hvis den forstår at maksimere belønningssignalet, vil den løse opgaven. Reinforcement-indlæringen går ud på at få agenten til at finde en omsætning af tilstande til handlinger, som maksimerer dette belønningssignal[3].

I hvert tidstrin sender systemet information om tilstanden til agenten – se figur 1. På baggrund heraf træffer agenten en beslutning om, hvilken handling der skal udføres. Hvis systemet er en fabriksmaskine, kunne det være at agenten modtager information om temperaturen, og hvis den bliver for høj, forsøger agenten at udføre nogle handlinger, som det har erfaring med, vil sænke temperaturen. Agenten er egentlig ligeglad med temperaturen i sig selv, men den har derimod erfaring for, at meget høje temperaturer medfører et negativt evalueringssignal – en straf – og på denne indirekte måde har temperaturen betydning for den.

Ideen er, at agenten skal prøve de forskellige handlinger og igennem sine erfaringer finde ud af, hvilke handlinger der giver mest belønning og mindst straf. I komplekse opgaver giver dette anledning til to vanskelige problemer: Agenten får et evalueringssignal, der er en skalar, tilbage for en handling, som er en vektor – hvis den begår en fejl, modtager den ikke nok information til at regne ud, hvad den skulle have gjort. Lad os som eksempel sige, at en agent, der skal styre en cykel, forsøger at dreje styret lidt til højre og straks modtager en straf. Måske skulle den slet



Figur 4: Cyklens rute for samtlige indlæringsrunder. Afstanden fra billedets øverste til nederste kant svarer til trekvart kilometer. Pilen markerer cyklens startposition.

ikke have drejet styret eller den skulle have drejet kraftigt til højre. Umiddelbart er det ikke til at vide. Et andet mere dybtgående problem er, at de enkelte handlingers indflydelse rækker videre end blot til det næste evalueringssignal og kan have indflydelse på den næste tilstand og de næste mange signaler. Belønningen for en handling kan være forsinket – den falder først, når konsekvenserne af hele sekvensen af handlinger er tydelige. Som eksempel på det andet problem kunne man tænke sig, at agenten først har drejet styret lidt til den ene side, lidt til den anden, tilbage igen og derpå modtaget en straf, fordi cyklen er væltet. Nu er det meget vanskeligt at sige, hvilke handlinger der egentligt var skyldige, og hvilke handlinger der forbedrede situationen.

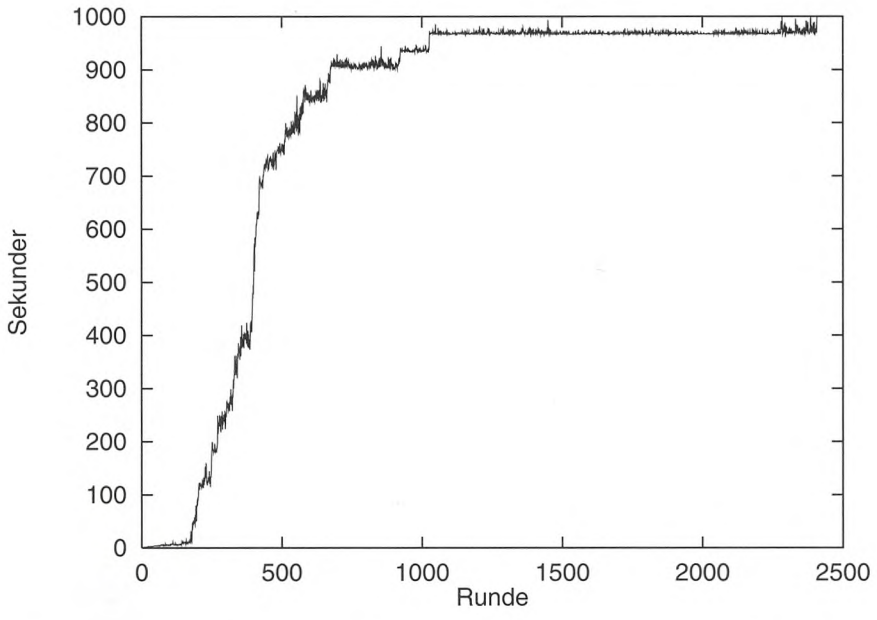
Kunstige neurale netværk

Hvordan bygger man så en agent? En metode er at skrue den sammen som et system af en masse simple enkeltdele som for eksempel neuroner i et kunstigt neuralt netværk og så håbe på, at man kan hitte på nogle regler for, hvordan enkeltdelene skal påvirke hinanden, for at agenten som helhed lærer noget nyttigt. I 1988 opstillede Richard Sutton et ligningssystem, der i dag er kendt som TD(λ)-ligningerne, som løser dette problem[2].

Grundideen i disse ligninger er e-sporene, som holder rede på hvor aktivt et handlingsvalg har været i en bestemt tilstand. E-sporene bruges til at opdatere værdierne af vægtene i netværket, og vægtene bruges til at beregne værdien af at udføre en bestemt handling, når systemet er i en given tilstand. Agenten vælger hvilken handling (eller kombinationer af handlinger) den vil udfører, ved at vælge handlinger med størst mulig værdi, eller handlinger som leder til størst mulig værdi. Værdien af en handling eller en tilstand er et udtryk for i for høj grad agenten regner med, at handlingen eller tilstanden vil lede til straf og belønning, hvor hurtigt de vil komme, og hvor store de vil være. Hvergang agenten vælger en handling bliver det tilhørende e-spor større, hvorefter det henfalder eksponentielt med en faktor λ. Når straffen eller belønningen falder er værdierne af e-sporene et godt bud på, hvor skyldige de enkelte handlingsvalg er i udfaldet. E-sporene kan altså i en vis forstand kaldes en hukommelse over de seneste handlingsvalg. Halveringstiden vælges i anvendelser typisk til cirka 35 tidstrin. I 1992-93 blev det vist, at under vise tekniske omstændigheder vil TD(λ)-ligningerne føre til, at agentens opførsel før eller senere konvergerer der hen, hvor vi ønsker den.

Men har vi så ikke opnået det vi ønskede? Agenten kan nu på egen hånd lære en nyttig opførsel ud fra sine erfaringer. For det første er disse tekniske omstændigheder mere snævre end vi kan være tilfredse med, og for det andet betyder "før eller senere" i praksis, at man ofte skal være særdeles tålmodig.

Den meget sparsomme evaluering af agentens handlinger, der tildels er skyld i den langsomme indlæring, har imidlertid også en enorm anvendelsesmæssig fordel i forhold til andre former for indlæring af kunstige neurale netværk. Den bedst kendte metode til indlæring er nok vejledt indlæring baseret på backpropagation. Denne indlæringsmetode går meget groft sagt ud på, at man serverer nogle eksempler for et netværk, og når det har gættet på nogle svar, serverer man også de rigtige svar



Figur 5: Antal sekunder, agenten kan balancere på cyklen, som funktion af antallet af indlæringrunder. Kurven er fremkommet som gennemsnit af 40 agenters indlæring.

for det. Derefter kan gradienten af fejlen beregnes, og netværkets vægte kan justeres i den rigtige retning. Modsat denne type indlæring behøver personen, der indlærer agenten, ikke at have den fjerneste idé om, hvad svaret på et problem er, dvs. hvad den rette handling er i en bestemt situation. Personen skal blot kunne specificere, hvilke tilstande af systemet, der er gode, og hvilke man ønsker at undgå. Det betyder, at netværket kan lære at løse opgaver, som mennesker ikke kender løsningen til! (En person med særdeles god tid og tålmodighed vil dog kunne løse den samme opgave ved slavisk at gennemkøre den samme indlæring som agenten.)

En person, der ikke kender andet til skak end reglerne, kan derfor i princippet lære en agent at spille skak. Under spillet vil agenten ikke modtage nogen evaluering af situationen, men ved spillets afslutning vil den modtage en straf eller en belønning alt efter, om det har tabt eller vundet (eller 0 for remis). Ingen har dog gjort forsøget med skak. Et andet brætspil er derimod blevet prøvet. Gerald Tesauro har siden 1991 arbejdet med at få en agent til at spille backgammon[4,5]. I 1995 nåede agenten en spillestyrke, der var tæt på backgammon-eksperters. Ganske overraskende viste det sig, at agenten spillede en særlig åbningssituation anderledes end eksperterne gjorde. Da de to forskellige strategier blev analyseret med sandsynlighedsregning, viste det sig, at den, agenten havde fundet, var den smarteste. Oplæringen af Tesauros agent har krævet mere end en million spil – flere end et menneske kunne nå at spille i løbet af et helt liv. Men det var ikke noget problem: Tesauro lod blot agenten spille mod sig selv.

Cykelkørsel

På trods af de mange dybtgående problemer, der stadig venter på deres afklaring, er det i dag muligt at få gode resultater ud af at anvende reinforcementindlæring på simple systemer. Der findes desuden nogle få eksempler på ganske komplekse problemer, der er blevet knækket – Tesauros backgammon er et af dem. Et andet eksempel er en agent, der har lært at balancere på en cykel alene ud fra at blive straffet, når cyklen væltede. Dette arbejde blev udført i år i reinforcement-gruppen på Niels Bohr Institutet, som er under vejledning af Preben Alstrøm. Agenten modtog information om, hvor meget cyklen hældede, den første og anden afledte af denne hældning, hvor meget styret var drejet og vinkelhastigheden. Agenten skulle så i hvert tidstrin beslutte, hvilket drejningsmoment, der skulle anvendes på styret. Når en cyklist med god fart svinger, drejer cyklisten ofte ikke på styret, men svinger i stedet ved at læne sig en lille smule ind i svinget, dvs. ved at forskyde massecentrum af systemet uden for cyklens plan. Denne handlingsmulighed fik agenten også.

Figur 5 viser indlæringskurven for vores agent. Indlæringen blev stoppet, når agenten kunne holde balancen i 1000 sekunder.

Middelindlæringstiden ligger altså på omkring 500 runder: 500 gange skal agenten vælte på cyklen, før den har fanget ideen. Modelcyklen, som agenten blev trænet på, var en computersimulering af en rigtig cykel med en simplifikation: Cyklens forgaffel var lodret for at forsimple udledningen af ligningerne for cyklens dynamik på trods af, at dette gjorde opgaven sværere for agenten.

Figur 2 viser cyklens rute set fra oven i starten af indlæringen, idet cyklen i hvert tidstrin er markeret med en streg. Cyklen bliver startet forfra det samme sted, hver gang den vælter.

Figur 3 viser cyklens rute på et senere tidspunkt, og

endelig viser figur 4 hele ruten, efter at agenten er udlært.

Konklusion

"At lære at handle på en måde, der bliver belønnet, er et tegn på intelligens," skrev John Watkins i sin Ph.D.-afhandling fra 1989 og undrede sig samtidig over at denne simple tanke var overset i forskningen i kunstig intelligens. Det er den ikke længere.

Referencer:

1) Stuart Russell og Peter Norvig: *Artificial Intelligence – A Modern Approach*, Englewood Cliffs, NJ: Prentice Hall, 1995. (En omfattende og god introduktion til hele kunstig intelligens-området.)

2) Richard S. Sutton: *Learning by the Methods of Temporal Differences*, Machine Learning, Kluwer Academic Publishers, vol. 3, side 9–44, 1988.

3) Richard S. Sutton og Andrew G. Barto: *Introduction to Reinforcement Learning*, MIT Press/Bradford Books, 1998. (Den kanoniske bog inden for reinforcement-indlæring, på trods af at papir-udgaven først udkommer januar 98. Den elektroniske udgave findes på http://www-anw.cs.umass.edu/~ rich/book/the-book.html)

4) Gerald Tesauro: *Practical Issues in Temporal Difference Learning*, Machine Learning, vol. 8, side 257–277, 1992.

5) Gerald Tesauro: *TD-Gammon, A Self-Teaching Backgammon Program, Achieves Master-Level Play*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, 1994. ftp://archive.cis.ohio-state.edu/pub/neuroprose/tesauro.tdgammon.ps.Z

Jette Randløv er ph.d.-studerende ved Niels Bohr Institutet. Hun interesserer sig for reinforcement-indlæring, kunstig intelligens og dertil knyttede filosofiske problemstillinger.