

AI utan uppkoppling

Utforska Artificiell Intelligens

Aktiviteter och undervisningsmaterial om artificiell intelligens

Förord

Artificiell intelligens (AI) blir alltmer betydelsefull i vårt samhälle. Ett exempel på detta är den tyska¹ regeringens AI-strategi som publicerades i slutet av 2018. Men ännu viktigare är hur vi redan interagerar med AI-system som om det vore helt naturligt, exempelvis när vi använder röstassistenter som Siri eller Alexa. Trots detta visar undersökningar att över 50 % av tyskarna inte vet vad artificiell intelligens egentligen är.

För att möta denna kunskapsbrist har vi sammanställt en samling av aktiviteter utan dator, så kallade *Unplugged Activities*, som handlar om AI. Dessa aktiviteter ger elever i alla åldrar möjlighet att aktivt utforska idéer och koncept inom datavetenskap utan att behöva använda en dator.

Denna broschyr innehåller fem aktiviteter som du kan använda för att lära ut idéer och koncept kring artificiell intelligens till elever i alla åldrar.

Idag bygger AI främst på maskininlärning, men artificiell intelligens är mycket mer än så: AI handlar inte bara om tekniska aspekter, utan väcker också frågor med stor samhällsrelevans. Denna broschyr visar möjligheter på hur dessa ämnen kan diskuteras med både barn och vuxna.

Om du har frågor, kommentarer eller synpunkter om detta material, tveka inte att kontakta oss på hi@aiunplugged.org.

Vissa aktiviteter kräver extra material.

Utskrivbara mallar finns här:
<https://aiunplugged.org>



¹ **Översättarens anmärkning:** Detta dokument producerades ursprungligen i Tyskland, vilket förklarar referensen till den tyska regeringens AI-strategi.

Innehållsförteckning

• Klassificering med beslutsträd.....	4
• #deeplearning.....	8
• Förstärkningsinlärning	12
• Tillbaka till grunderna	16
• Turing testet.....	20
• Fler idéer.....	23

Klassificering med beslutsträd

Spelet "Bra Apa – Dålig Apa"

Målgrupp Grundskolenivå, gymnasienivå

Det handlar om detta

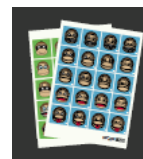
Hur kan en dator fatta självständiga beslut? Hur avgör en dator om en person är atletisk eller bör beviljas ett lån? Sådana klassificeringsprocesser är en vanlig tillämpning av AI. I den här aktiviteten får eleverna möjlighet att skapa sin egen klassificeringsmodell med hjälp av ett beslutsträd. Slutligen väljs den bästa modellen bland elevernas arbete för att användas i fortsatta klassificeringsuppgifter.

Idéerna bakom det

- AI klassificerar data baserat på mönster.
- AI använder den klassificeringsmodell som bäst passar den givna datan.
- Klassificeringsmodeller är inte perfekta.
- Vissa kombinationer av egenskaper indikerar en specifik kategori.

Det här behöver du

- Apa-kort (förberedelse: klipp ut mallen eller använd den digitala versionen)
- Tavla med magneter eller en anslagstavla



Så här går det till

Eleverna undersöker hur en serie exempeldata (träningsdata) kan kopplas till en viss kategori. För att göra detta utvecklar de kriterier som kan användas för att klassificera nya element. Därefter testas de framtagna modellerna med nya exempel (testdata), och noggrannheten i förutsägelserna utvärderas.

Sammanhang

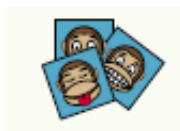
Vi är djurskötare på en djurpark och ansvarar för att mata aporna. Apona ser väldigt söta ut, men vi måste vara försiktiga eftersom vissa av dem biter. Vi vet redan vilka apor i djurparken som biter. Men snart kommer nya apor att ansluta sig till gruppen, och vi behöver ta reda på vilka av dem som biter och vilka som inte gör det – helst utan att komma för nära deras tänder.

Aktivitetsbeskrivning

Beroende på målgruppen kan du välja den grundläggande versionen av spelet med 20 bildkort (blå) eller den avancerade versionen med 40 bildkort (blå och gröna). Dessa 20 eller 40 apor representerar alla djur på zooet, och vi vet redan om de biter eller inte. Korten delas upp i träningsdata och testdata. Utifrån träningsdatan utvecklar vi kriterier som avgör om apona biter och testar sedan kriterierna tillförlitlighet med hjälp av testdatan. Träningsdatan delas in i två kategorier – "biter" och "biter inte" – och fästs på tavlan. Testdatan avslöjas inte i början.

Du kan skapa egna regler för att skilja apona åt eller använda något av de förslag som finns nedan (det går också att använda mindre urval av korten). Reglerna som används i exemplen illustreras med beslutsträd.

Version 1 (blå)



Version 1: Träningsdata

Biter: 6, 7, 8, 15

Biter inte: 1, 2, 4, 9, 12, 14, 17, 18

Version 1: Testdata

Biter: 3, 5, 11, 19

Biter inte: 10, 13, 16, 20

Börja med att göra eleverna medvetna om detaljer som de kan fokusera på genom att demonstrera proceduren med ett exempel. Jämför till exempel apa-korten 01 till 04 och 05 till 08. I detta exempel kan munens form vara en indikation på vilka apor som biter, medan ögonen inte har någon betydelse (Figur 1).

Alternativt, om du arbetar med äldre elever, kan du använda den enklare versionen av spelet (Version 1) för att demonstrera reglerna och de nödvändiga procedurerna.

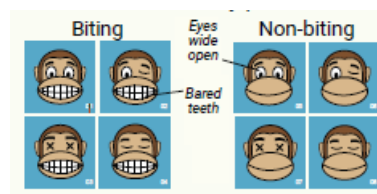
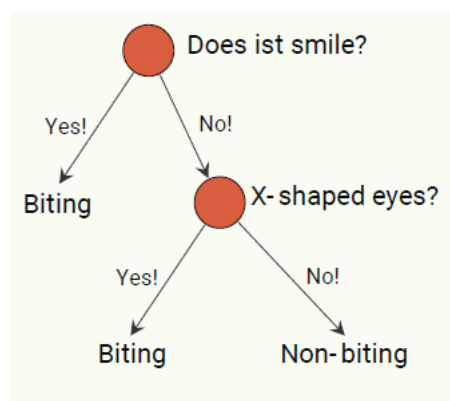
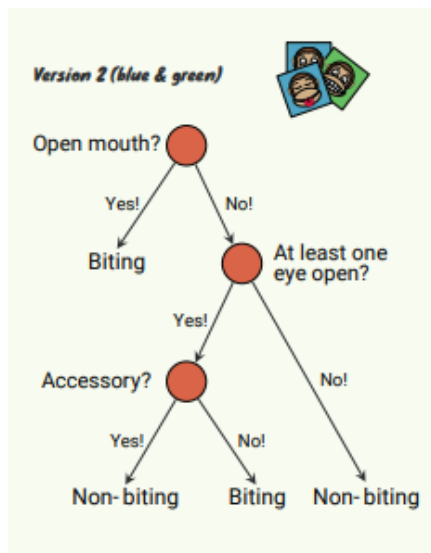


Fig. 1: I detta enkla exempel biter alla apor som visar tänderna.

Eleverna bildar par och använder träningsdatan för att utveckla kriterier som skiljer bitande apor från icke-bitande. Dessa kriterier måste dokumenteras tydligt så att en annan grupp senare kan använda dem för att bedöma nya exempel. Ett sätt att strukturera kriterierna är genom ett **beslutsträd**. Målet är att varje egenskap – beroende på om den finns eller saknas – tydligt ska avgöra vilken grupp en apa tillhör. Att använda beslutsträd är valfritt. Alternativt kan man också





skriva ner beslutskriterierna i form av tydliga regler.

När träningsfasen är avslutad byter grupperna sina formulerade kriterier med en annan grupp. Därefter visas bilder på de återstående aporna (testdata) en i taget. För varje bild ska grupperna, utifrån reglerna som deras klasskamrater har utvecklat, avgöra om apan biter eller inte. Varje grupp antecknar sina beslut.

Efter att alla apor har bedömts utvärderas vilka grupper som har lyckats bäst med att förutsäga apornas bitbeteende. Eleverna blir medvetna om att många klassificeringsmodeller kan identifiera de flesta apor korrekt, men att det är svårt att klassificera alla helt felfritt.

Som djurskötare är det därför klokt att använda den mest träffsäkra modellen när vi matar de nya aporna, även om det inte helt eliminerar risken för att bli biten.

I den avancerade versionen kan bild nr 21 (se Fig. 2) användas för att illustrera de utmaningar ett AI-system kan stöta på när en egenskap hos ett objekt

Version 2: Träningsdata

Biter: 1, 2, 5, 9, 10, 14, 15, 16, 17, 28, 33, 35, 36
Biter inte: 4, 7, 12, 19, 22, 23, 24, 25, 30, 32, 37, 38, 39, 40

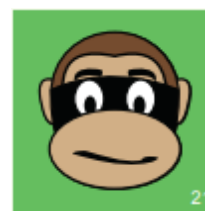
Version 2: Testdata

Biter: 6, 13, 18, 34
Biter inte: 3, 8, 11, 20, 21, 26, 27, 29, 31

skiljer sig avsevärt från träningsdatan. Vi har ingen tidigare erfarenhet av egenskaperna hos apan i bild nr 21, eftersom den har en ny, okänd munform. Därför är det inte möjligt att avgöra om den biter eller inte med hjälp av de tidigare fastställda kriterierna.

I praktiken är det svårt att förutse hur ett AI-system kommer att hantera sådana situationer. Istället för bild nr 21 kan man använda en bild av ett helt annat djur för att tydliggöra hur nya och oväntade egenskaper kan påverka klassificeringen.

Dessa exempel kan också kopplas till verkliga situationer. Till exempel kan en bank plötsligt neka en kund ett lån utan en tydlig förklaring, eller en självkörande bil kan misstolka löv på vägen som ett farligt hinder och tvärbromsa. I sådana fall kan AI-system skapa problem, särskilt om det är oklart hur besluten har fattats.



Figur 2: För apa 21 kan inga tydliga kriterier härledas från datan.

Bakgrund

Kategoribildning Kategorisering blir möjlig genom att känna igen återkommande mönster i individuella element. Men hur hänger detta samman med artificiell intelligens (AI)?

Inom det som kallas övervakad inlärning (supervised learning) analyserar ett AI-system en mängd indata och utdata (träningsdata) för att lära sig samband mellan dem samt identifiera typiska mönster för olika kategorier. Denna kunskap används sedan för att klassificera nya element i rätt kategori. Testdata, där kategorierna är kända för oss men inte för AI-systemet, används för att utvärdera hur väl den inlärd klassificeringsmodellen fungerar.

Samma princip används i neurala nätverk och andra AI-applikationer. Men denna process kan medföra olika utmaningar, eftersom ingen modell är perfekt. Beroende på träningsdatan kan klassificeringsmodellen ge för stor vikt åt vissa egenskaper eller förbise andra, vilket gör

att den får svårt att klassificera okända element korrekt. Mer träningsdata kan minska dessa problem, men det leder inte alltid till bättre resultat. För mycket träningsdata kan istället orsaka överfitting, där AI-systemet "lärt sig utantill" snarare än att generalisera, vilket gör det sämre på att hantera nya exempel.

Det är viktigt att lyfta fram dessa aspekter av maskininlärning i aktiviteten. Under testfasen kan eleverna förklara vilka egenskaper de använde för att klassificera aporna. Detta visar att olika grupper kan ha skapat olika regeluppsättningar. Påpeka att en klassificeringsmodell sällan är 100 % korrekt och att den modell som bäst klassificerar testdatan i slutändan kommer att väljas.

Avslutningsvis kan eleverna beskriva sin egen inlärningsprocess och jämföra den med hur en dator lär sig.



Figur 3: Träningsdata uppdelad i två kategorier

#djupinläring

Bildigenkänning med neurala nätverk

Målgrupp På högstadie- och gymnasienivå

Det handlar om detta

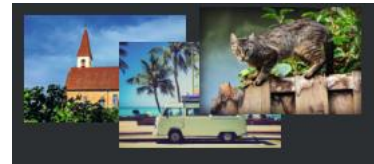
Hur kan en dator "känna igen" saker? Hur avgör en dator om ett foto föreställer en katt? Hur kan den skilja byggnader från människor? Att känna igen objekt baserat på deras form eller utseende är väldigt enkelt för människor. För en dator, som exempelvis kan användas i en självkörande bil för att identifiera objekt i sin omgivning, är detta en komplex uppgift. I denna aktivitet får eleverna möjlighet att utforska hur datorer känner igen innehållet i bilder.

Idéerna bakom det

- Neurala nätverk kopplar indata till specifika utdata: Rådata, såsom bilder, klassificeras genom att exempelvis tilldela termer till objekten i bilden.
- Neurala nätverk består av olika abstraktionslager som kan identifiera alltmer komplexa egenskaper.
- De klasser av objekt som ska kännas igen måste redan vara kända för AI-systemet.

Det här behöver du

- Fotokort med hus, katter och bilar för varje grupp.



Så här går det till

Eleverna återskapar bildigenkänningsprocessen för ett (förenklat) neuralt nätverk. De tar på sig rollerna som de olika lagren i ett sådant nätverk. De identifierar egenskaper från ett fotografi och klassificerar bilden. Under processen upptäcker de begränsningarna i systemet och överväger vilka förändringar som behövs för att förbättra resultatet i nätverket.

Sammanhang

Som människor förlitar vi oss starkt på vad vi ser. Om vi ser en katt, vet vi genast att det är en katt. Om vi ser en hund, känner vi direkt igen den. En dator kan däremot inte lika enkelt identifiera detta, men den kan lära sig – precis som vi gjorde som små. Genom att visa datorn många bilder av hundar, men också av andra djur, kan den lära sig vilka mönster i en bild som skiljer en hund från en katt. När datorn har tränats korrekt kan den inte bara automatiskt märka bilder, utan också upptäcka hudcancer eller – om den är inbyggd i en bil – reagera på hinder på vägen.

Aktivitetsbeskrivning

Börja med att diskutera hur en dator kan känna igen innehållet i en bild. Elevernas svar kommer ofta att handla om fördefinierade regler eller jämförelser med en bilddatabas, men idag fungerar datorer på ett annat sätt.

Dela in eleverna i grupper om tre. Varje grupp får en bunt med fotokort. I varje grupp har eleverna olika roller, där varje roll motsvarar ett lager i ett neuralt nätverk (se Fig. 4). Roller och uppgifter är följande:

A väljer en bild från buntan (B och C får inte se bilden!), skissar den på två olika sätt (30 sekunder per skiss) och lämnar dem vidare till B. Det är viktigt att **C** inte ser skisserna.

B tar emot skisserna från A och analyserar om de innehåller fyrkantiga, trekantiga eller runda former. Därefter skickar B vidare informationen till C.

C tolkar informationen med hjälp av en tabell och meddelar om originalbilden föreställer ett hus, en bil eller en katt.

Objekt	Fyrkantig form?	Triangulär form?	Rund form?
Hus	Ja	Ja	Nej
Bil	Ja	Nej	Ja
Katt	Nej	Ja	Ja

Till sist avgör A om svaret är korrekt.

Låt eleverna testa spelet i olika roller. Det går även att låta två elever dela på en roll, eller att varje roll tilldelas tre elever där varje person representerar en enskild neuron (dvs. en nod i lagret) istället för ett helt lager.

Efter en kort testomgång delar du ut bilder som antingen inte passar in i de kategorier som det neurala nätverket kan känna igen eller som har egenskaper som gör dem svåra att klassificera. Ett exempel är en bild på en hund, som nätverket inte identifierar korrekt – helt enkelt eftersom kategorin "hund" inte existerar i nätverket. Utifrån denna insikt får eleverna nu fundera över hur nätverket kan förändras och utökas för att i framtiden kunna känna

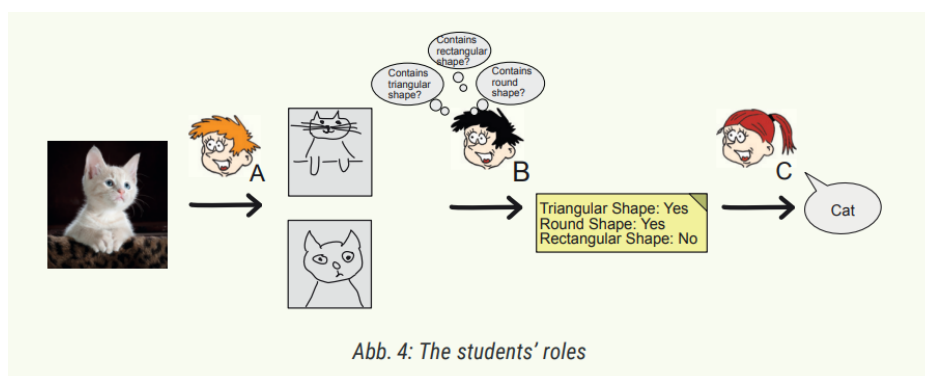


Abb. 4: The students' roles

Fig. 4: Elevernas roller

igen hundar eller andra objekt. Först och främst måste en ny utgångskategori läggas till. Samtidigt räcker inte längre de befintliga egenskaperna som nätverket identifierar. Därför måste antingen fler egenskaper läggas till för att kunna skilja mellan kategorierna, eller så behöver flera egenskaper kombineras för att skapa mer komplexa mönster.

Denna sammanslagning innebär i slutändan att ytterligare lager läggs till i det neurala nätverket. För att tydliggöra detta kan Fig. 6 användas för att visa hur enkla egenskaper kan kombineras till mer avancerade mönster.

Bakgrund

När det är svårt att översätta ett problem till logiska regler används ofta artificiella neurala nätverk för att lösa problemet. Typiska exempel är förståelse av texter eller igenkänning av objekt i bilder. Idén bakom artificiella neurala nätverk kommer från neurobiologin och bygger på den mänskliga hjärnans struktur.

Precis som en mänsklig nervcell, som bearbetar olika signaler och skickar en impuls vidare, hanterar ett artificiellt neuron olika indata och kan överföra en signal. Varje indata har en vikt som påverkar hur mycket det bidrar till resultatet. Detta gör att vissa indata har större betydelse än andra för neuronens utdata.

Ett artificiellt neuron fungerar som en enkel räknemaskin: det multiplicerar vikterna med indata, summerar resultaten och skickar sedan vidare en signal baserat på dessa beräkningar.

I likhet med det mänskliga nervsystemet är många artificiella neuroner kopplade samman och bildar ett nätverk. Dessa neuroner är organiserade i lager, och beroende på hur komplicerat problemet är kan ett nätverk ha två eller fler lager. I denna aktivitet utgår vi från en modell med tre lager. Om det finns fler lager mellan

indata- och utdatalagret kallas detta djupinlärning, eller deep learning.

För bildigenkänning och klassificering används ofta konvolutionella neurala nätverk. Dessa nätverk är specialiserade på att känna igen mönster och är därför mycket effektiva för att klassificera bilder. De använder en teknik som kallas **konvolution** för att extrahera egenskaper och mönster från indata. Idag kan dessa nätverk analysera och klassificera bilder snabbare än människor.

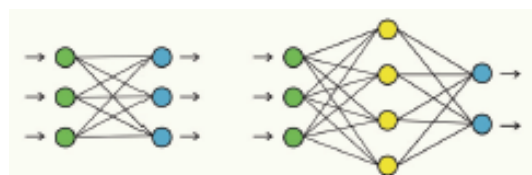


Fig. 5: Exempel på artificiella neurala nätverk - vänster: enkelt nätverk, höger: "djupare" nätverk

Hur fungerar detta egentligen? Digitala bilder består av små färgelement – pixlar – som är arrangerade i ett rutnät. Varje pixel har ett specifikt färgvärde. För en dator är foton dock inte bilder i sig, utan enbart numeriska värden. Vid bildigenkänning börjar neurala nätverk med att identifiera enkla egenskaper. För detta används filter som läggs över bilden. Detta liknar tekniken i fotoredigeringsprogram, där vi till exempel kan använda ett högpasfilter för att framhäva detaljer (se Fig. 7). I grund och botten handlar det om en matematisk beräkning, där flera pixlar analyseras tillsammans för att generera ett nytt



Fig. 7: Applying a filter

pixelvärde.

Fig. 7: Applicering av ett filter

Beroende på vilket filter som används

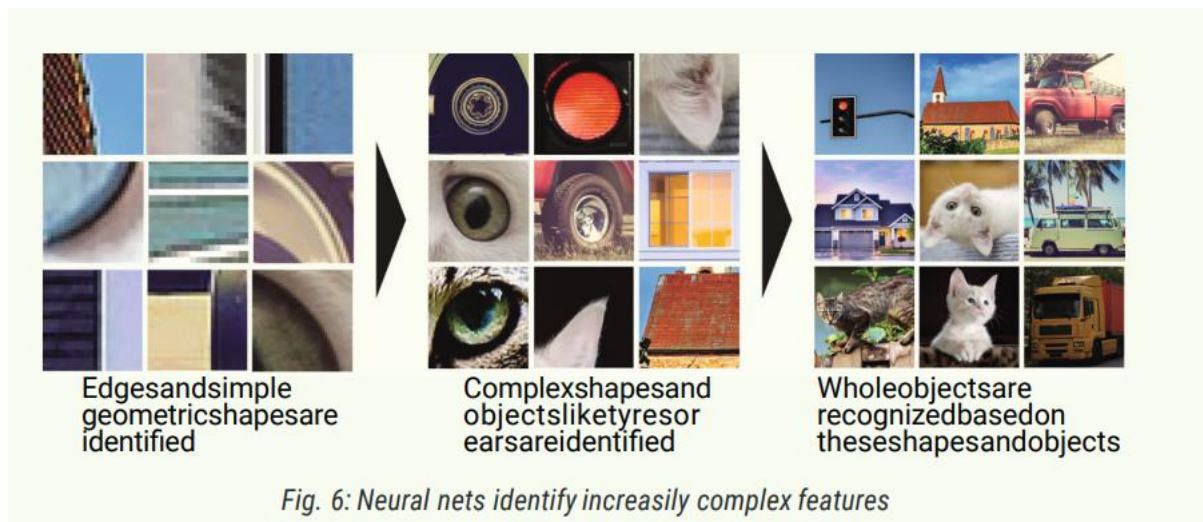


Fig. 6: Neurala nätverk identifierar alltmer komplexa egenskaper

kan man till exempel upptäcka att pixlar med liknande ljusstyrka kan grupperas för att bilda kanter. På ett senare lager extraheras mer komplexa egenskaper, såsom horisontella och vertikala linjer, cirklar eller hörn. Vanliga bildbehandlingsprogram, såsom GIMP, gör det möjligt att mata in sådana filter i form av matriser, vilket ger en bra möjlighet att själv utforska deras effekter.

I spelet fungerar elevernas skisser också som filter, eftersom de lyfter fram de viktigaste delarna av objektet som avbildas på fotot. Dessa skisser används sedan för att identifiera geometriska former i bilderna. Att endast använda tre egenskaper är dock en kraftig förenkling jämfört med ett verkligt neuralt nätverk, som innehåller miljontals neuroner i flera lager.

På de första nivåerna i ett neuralt nätverk används många enkla och mestadels geometriska filter. Dessa grundläggande mönster kombineras sedan, återigen genom filtrering, till mer komplexa strukturer. På djupare nivåer kan nätverket inte bara identifiera kanter och hörn, utan även delar av objekt, såsom ögon eller päls, och till slut även hela objekt, som hundar eller katter.

Under bearbetningen filtreras ofta överflödigt information bort, eftersom den exakta positionen för en diagonal linje i en bild i många fall inte är relevant för att känna igen ett objekt. Slutligen anger ett sannolikhetsvärde hur troligt det är att en bild tillhör en viss kategori.

Ett neuralt nätverk kan dock inte enkelt identifiera innehållet i vilken bild som helst. Användningsområdet är mycket begränsat, eftersom nätverket först måste tränas med ett mycket stort antal bilder, ofta flera tusen. Det lär sig vilka egenskaper som är avgörande för att en bild ska tillhöra en viss kategori. Därför kan det neurala nätverket endast korrekt klassificera bilder som tillhör en kategori det redan känner till.

Till exempel kan ett nätverk som är utformat för att skilja mellan hundar och katter inte känna igen andra djur, utan placerar dem istället i en av de två kända kategorierna. Ett tränat neuralt nätverk kan dock utföra sin uppgift mycket snabbare än en människa. Därför används metoder för bildigenkänning redan idag, till exempel i självkörande bilar för att identifiera olika objekt i trafiken, såsom mötande fordon och fotgängare, eller vid upptäckt av hudcancer.

Förstärkt Inläring

Slå Krokodilen

Målgrupp Grundskola, högstadium

Vad handlar det om?

Vi känner till datorer som kan spela schack och slå mänskliga spelare med överlägsen lätthet. Det kinesiska brädspelet Go har däremot länge ansetts vara så komplext att endast människor kan bemästra det – tills Google använde AlphaGo för att skrämman professionella spelare. I denna aktivitet undersöker vi hur datorer kan lära sig strategier för spel, trots att de endast känner till spelets regler.

Dessa idéer ligger bakom

- Datorer kan lära sig genom "belöning" och "bestraffning".
- Datorer utvärderar nyttan av slumpmässiga handlingar baserat på belöning och bestraffning.
- Datorer lär sig strategier eller handlingssekvenser genom att sträva efter maximal belöning.

Vad du behöver

- Per par av elever: 1 "mini-schack" bräde, 3 apkort och 3 krokodilkort, 1 översikt över möjliga drag
- Färgglada godisbitar (t.ex. chokladpolletter) eller pappersmarkörer för att utvärdera dragen i 4 olika färger (gul, röd, orange, blå; ca 20 per färg)

Så här går det till

Två elever spelar ett "mini-schack"-spel mot varandra. En elev antar rollen som en "papper"-dator. I början väljer datorn sina drag slumpmässigt, men lär sig gradvis med hjälp av ett belöningssystem baserat på godisbitar eller markörer vilka drag som leder till vinst och vilka som slutar i förlust.

Genom att utveckla sin strategi på detta sätt blir datorn allt bättre med tiden.

Denna aktivitet är baserad på en idé från CS4Fun.

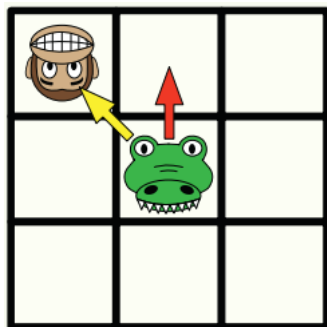
(<http://www.cs4fn.org/machinelearning/sweetlearningcomputer.php>)

Sammanhang

Hur lär sig människor att spela brädspel eller datorspel? Kanske genom att titta på när andra spelar, eller genom att själva prova olika handlingar eller drag för att se hur de påverkar spelet. Ju oftare vi vinner, desto bättre blir vi på spelet. Vi utvecklar strategier för att avgöra vilka drag som är mest framgångsrika i olika spelsituationer. På samma sätt lär sig en dator att spela spel.

Aktivitetsbeskrivning

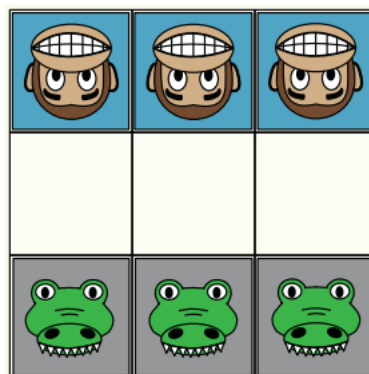
Spelet följer enkla schackregler: Varje spelpjäs rör sig som en bonde, det vill säga den kan bara röra sig rakt fram och kan bara slå motståndarens pjäser diagonalt. En elev tar rollen som aporna och agerar som mänsklig spelare. En annan elev spelar rollen som datorn i form av krokodiler.



Figur 8: Möjliga rörelser för en spelpjäs

En sida har vunnit om den lyckas...

- föra en spelpjäs till andra änden av spelplanen.
- slå ut alla motståndarens spelpjäser.
- se till att motståndaren inte kan göra några drag i nästa omgång.



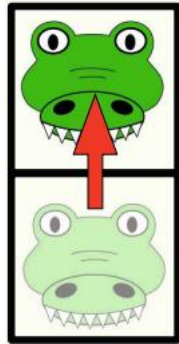
Figur 9: Spelbräde innan spelet börjar

Som förberedelse sprids utskrifter av datorns möjliga drag ut framför spelaren som styr krokodilerna. Därefter fördelas chokladpolletter (eller andra färgade markörer) på dessa drag. För varje färgad pil placeras en motsvarande färgmarkör i området till höger om varje spelsituation (se Fig. 11).

Den mänskliga spelaren börjar och får röra sig fritt enligt spelets regler. Därefter är det krokodilernas tur. Spelaren jämför den aktuella spelplanen med de möjliga dragen och väljer den spelsituation som bäst matchar från de givna alternativen. För att underlätta orienteringen anges vilken spelomgång varje spelsituation tillhör. I första omgången behöver spelaren bara ta hänsyn till de två alternativen för drag 1, i andra omgången de 10 möjliga dragen för drag 2 och i tredje omgången de 7 möjliga dragen för drag 3. Symmetriska spelsituationer listas inte två gånger.

Krokodilspelaren blundar sedan och väljer slumpmässigt en av markörerna som ligger bredvid den aktuella spelsituationen och placerar den på det avbildade brädet. Färgen på markören avgör vilket drag som ska genomföras, och spelaren flyttar spelpjäsen i

enlighet med pilen i samma färg. Om exempelvis en röd markör dras, flyttas krokodilen i den riktning som den röda pilen visar.



Figur 10: Krokodilen flyttas längs den röda pilen

Denna process upprepas tills vinnaren av omgången har fastställts. Innan en ny omgång börjar justerar datorn sin strategi på följande sätt:

- **Krokodilerna har vunnit:** En extra markör i samma färg som det senaste vinnande draget placeras på rutan för det draget.
- **Aporna har vunnit:** Den chokladmarkör som avgjorde krokodilspelarens senaste drag tas bort. Apornas spelare får äta den.

Dessutom placeras alla markörer tillbaka till höger om spelplanen.

Valfritt: För att förenkla reglerna kan man välja att inte lägga till extra markörer när krokodilerna vinner.

Bakgrund

Till en början har datorn knappt någon chans att vinna eftersom den väljer sina drag helt slumpmässigt (genom att dra en markör med slutna ögon). Ju fler spel den genomför, desto bättre blir den: den "lärt sig" vilka drag som leder till vinst och vilka som bör undvikas eftersom de tidigare

har lett till förlust. På så sätt förbättras datorns strategi gradvis.

Eftersom datorn **bestraffas vid förlust och belönas vid vinst**, kallas denna metod för **förstärkningsinlärning** – inlärning genom belöning och bestraffning:

- **Bestraffning** = Att ta bort en godisbit efter ett drag som ledde till förlust.
- **Förstärkning** = Att lägga till en godisbit vid ett drag som ledde till vinst.

Denna metod sällar bort de drag som tidigare har lett till förlust, så att endast de "bästa" dragen återstår med tiden. I praktiken elimineras dock inte misslyckade strategier omedelbart, utan sannolikheten för att de väljs minskar gradvis. På detta sätt lär sig AI-systemet vilken strategi som fungerar bäst i olika situationer, men det utesluter inte omedelbart en strategi bara för att den inte alltid har varit framgångsrik.

Även om denna metod är förenklad i spelet genom att direkt ta bort drag som leder till förlust, kan det aldrig hända att alla möjliga drag elimineras för en viss spelsituation. För varje situation finns det alltid minst ett möjligt drag som inte leder till omedelbar förlust.

På detta sätt kan datorer lära sig att vinna ett spel enbart genom att känna till spelreglerna och möjliga indata. Till exempel, om en dator lär sig att spela tv-spelet Super Mario, kommer den till en början att trycka på knappar slumpmässigt. Detta kan leda till att den står stilla i flera minuter eller springer in i samma fiende om och om igen. Datorn analyserar pixlar och objekt på skärmen och reagerar genom att skicka kommandon.

Datorns mål är att maximera poängen, som fungerar som en belöning. Ju längre den rör sig åt höger, desto större blir vinsten. Med tiden lär sig datorn att hoppa över en större belöning om en fiende befinner sig direkt till höger, eftersom den då kommer längre i spelet genom att hoppa över fienden. På

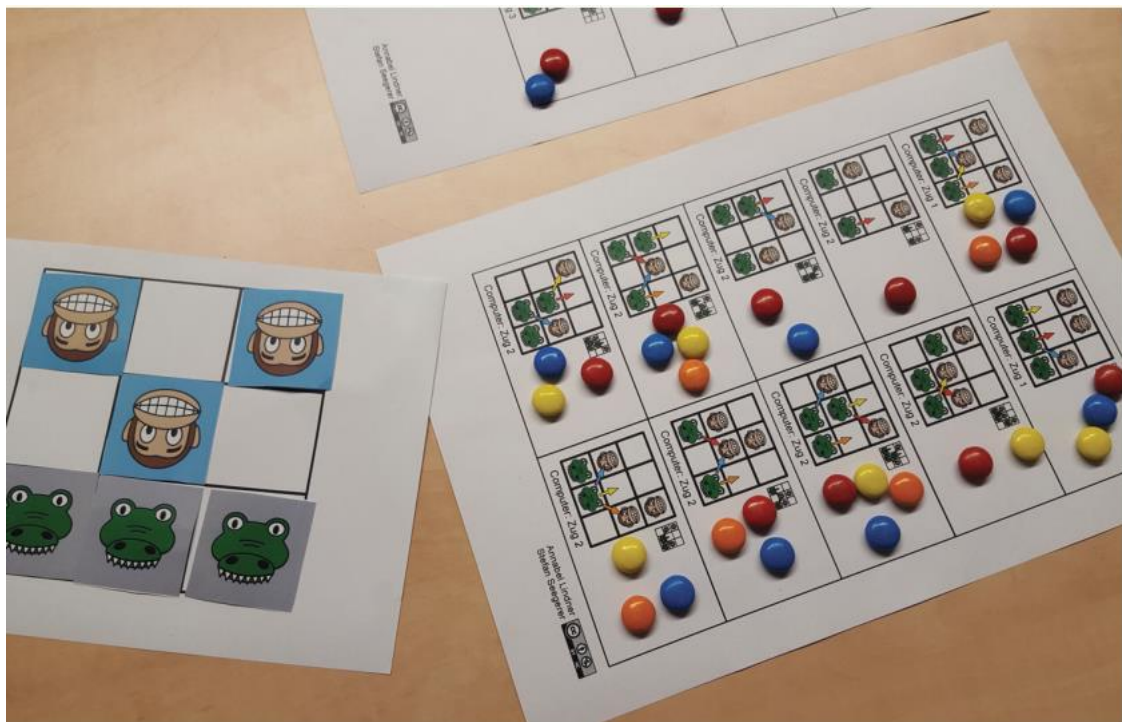
detta sätt förbättras AI-systemets prestation successivt, där systemet alltid försöker maximera sin belöning (eller mer exakt: en viss målfunktion).

Som en del av dekontekstualiseringen (dvs. att överföra kunskap från ett sammanhang till ett annat för att skapa djupare förståelse) ska eleverna analysera

Därefter kan en video om spelet Super Mario (se webbplats) användas för att visa hur förstärkningsinlärning fungerar i ett neuralt nätverk. Låt eleverna reflektera över begränsningarna i de strategier som systemet lär sig.

Denna aktivitet kan med fördel kombineras med "Back to the Roots: Crocodile Chess" och "Classic AI" för att tydliggöra skillnaden mellan självlärande system och traditionella regelbaserade AI-metoder.

hur datorns beteende utvecklas. Det bör bli tydligt att datorn utvecklar en effektiv spelstrategi genom att utvärdera helt slumpmässiga handlingar.



Figur 11: Speluppställning: Fördelningen av chokladmarkörer visar de inlärdade strategierna

Tillbaka till grunderna

Krokodilschack och klassisk AI

Målgrupp: Grundskolenivå, Högstadienivå

Det handlar om detta

De tidigare övningarna fokuserar i stor utsträckning på självlärande AI-system. Men AI handlar om mer än bara inläring – dess rötter finns i logik och idén att formalisera kunskap genom matematiska beskrivningar, vilket gör den tillgänglig för datorer.

Denna aktivitet belyser skillnaderna mellan självlärande AI och traditionella metoder samt de begränsningar dessa system har. För att illustrera detta utförs den tidigare övningen i **förstärkningsinläring** nu istället med ett **expertsystem**, vilket tydligt visar de fundamentalt olika angreppssätten.

Dessa idéer ligger bakom det

- Kunskap måste kunna representeras på ett formellt sätt för att bearbetas automatiskt.
- Expertsystem kan kombinera regler och fakta för att generera ny kunskap.
- Sådana AI-system fattar inga självständiga beslut utan arbetar enligt logikens regler.
- AI-system har bearbetningsmekanismer för att automatiskt härleda information från befintlig kunskap.

Vad du behöver

- Per par av elever: 1 "minischack"-bräde, 3 apkort och 3 krokodilkort, 1 översikt över regler för nästa drag

Så här fungerar det

Precis som i aktiviteten om förstärkningsinläring spelar två elever en omgång "mini-schack" mot varandra, där en av dem tar rollen som en "pappersdator". Om denna aktivitet kombineras med den tidigare versionen av förstärkningsinläring kan det vara en bra idé att låta eleverna byta roller. Till skillnad från den självlärande versionen väljer datorn nu inte sina drag slumpmässigt, utan följer i stället fördefinierade regler, som delas ut i form av utskrifter.

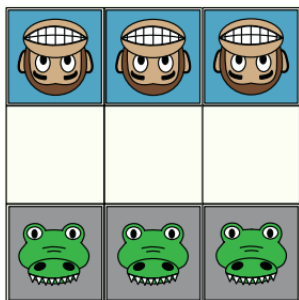
Sammanhang

Hur kan en dator programmeras för att spela bräd- eller datorspel? Datorer kan endast "förstå" spelregler och agera därefter om reglerna presenteras på ett sätt som de kan bearbeta. För att en dator ska kunna använda reglerna måste kunskap representeras formellt, till exempel genom matematiska termer. På så sätt kan datorn använda logik för att utvärdera information och avgöra sina handlingar. AI-system är därför inte egentligen intelligenta, utan de använder sig av olika metoder för att dra slutsatser utifrån den kunskap som finns tillgänglig.

Activity Description

Spelet följer enkla schackregler och har samma grundförutsättningar som i aktiviteten om förstärkningsinläring. Varje pjäs rör sig som en bonde – den kan endast gå rakt fram och slå motståndarpjäser diagonalt.

En elev tar rollen som aporna och spelar som mänsklig spelare. En annan elev tar rollen som datorn och spelar som krokodilerna.

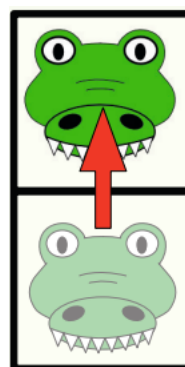


Figur 12: Spelbräde innan spelet börjar

En sida har vunnit om den lyckas...

- föra en spelpjä till andra änden av spelplanen.
- slå ut alla motståndarens spelpjäser.
- se till att motståndaren inte kan göra några drag i nästa omgång.

Som förberedelse får den elev som spelar krokodilerna en utskrift av reglerna för datorns drag. Dessa regler ersätter de slumpmässiga drag och markörer som användes i aktiviteten om förstärkningsinläring. Den mänskliga spelaren börjar och kan röra sig fritt enligt spelets regler. Därefter är det krokodilernas tur. Spelaren jämför den aktuella spelsituationen med "regelmatriken" och väljer det relevanta scenariot bland 10 möjliga alternativ (symmetriska situationer listas inte två gånger). Sedan gör spelaren det drag som reglerna anger.



Figur 13: Krokodilen flyttas längs den röda pilen.

Denna process upprepas tills en vinnare har fastställts. Spelet kan spelas i flera omgångar för att undersöka om datorn alltid kan vinna med hjälp av sina regler.

Därefter presenteras den utökade versionen av krokodilschack med en 4x4-spelplan (se webbplats) för eleverna. Påpeka att datorns regler inte längre är tillräckliga. Låt eleverna jämföra detta med mini-schackvarianten i aktiviteten om förstärkningsinläring, där datorn kunde lära sig strategier. Även här visar det sig att datorns "kunskap" inte räcker till. Detta är punkten där båda systemen når sina begränsningar.

Eleverna funderar nu på vilka åtgärder som krävs för att anpassa datorns "kunskap" till

4x4-versionen av spelet. De inser att den regelbaserade datorn måste justeras manuellt av människor genom att lägga till nya regler för det bästa draget. I jämförelse

kan det självlärande systemet lära sig det bästa beteendet för 4x4-spelplanen på samma sätt som i den tidigare aktiviteten, det vill säga genom att utvärdera slumpmässiga handlingar – förutsatt att alla nya möjliga drag har lagts till. Det självlärande systemet behöver alltså en ny träningsfas, där det utökade spelets regler implicit förvärfvas, medan nya regler måste explicit läggas till i det regelbaserade expertsystemet. Här får människan uppgiften att först bestämma det bästa draget för varje spelsituation utifrån alla möjliga alternativ och sedan formalisera samtliga regler helt och hållet. Detta är inte nödvändigt för det självlärande systemet.

Den explicita formella representationen av regler som styr systemets handlingar är dock knappast möjlig, eller ens genomförbar, när det gäller komplexa och flerstegade problem. I detta sammanhang har självlärande AI-system en stor fördel jämfört med expertsystem, eftersom de kan utveckla sådana processer "självständigt". Dessutom kan de upptäcka samband i data som människor inte kan identifiera, exempelvis eftersom tusentals datamängder behöver analyseras för att hitta dem.

Bakgrund

Logik och kunskapsbearbetning spelar en viktig roll inom många områden av datavetenskap och är dessutom centrala ämnen inom artificiell intelligens. Eftersom naturligt språk är tvetydigt och alltför varierande för att vara ett lämpligt medium för att göra kunskap tillgänglig för datorer, har frågan om den bästa möjliga representationen för maskiner varit avgörande ända sedan AI:s början.

I detta sammanhang bygger traditionella AI-metoder på symbolisk kunskapsrepresentation, det vill säga en explicit representation av kunskap i datorsystem, exempelvis genom logik. Detta möjliggör en tydlig, enhetlig och exakt

representation av kunskap, vilket är nödvändigt för att datorer ska kunna bearbeta information. Sådana representationsmetoder används exempelvis i regelbaserade expertsystem, som fortfarande har en roll i kommersiella tillämpningar idag.

I dessa system används logiska påståenden för att representera fakta och kunskap om regler, och dessa används automatiskt för att dra slutsatser om hur datorn ska agera. I denna aktivitet motsvarar fakta den aktuella spelsituationen, medan regelkunskapen utgör instruktionerna för vilket drag som ska göras.

Fakta representeras genom giltiga påståenden, och en uppsättning regler i "om ... så"-format bildar regelbasen. Formellt kan denna "om ... så"-struktur uttryckas med exempelvis prepositionell logik. Ett kontrollsystem (inferensmotor/slutledningsmotor) väljer lämpliga regler baserade på fakta, utvärderar dem och agerar därefter. I mer avancerade expertsystem kan slutsatser som dragits från en regel även fungera som indata för ytterligare regler och därmed bidra till att utöka faktabasen.

I denna aktivitet tas kontrollsystemets roll över av den elev som spelar datorn. Med hjälp av draginstruktionerna, som representerar regelbasen, måste eleven dra slutsatser från den aktuella spelsituationen och genomföra nästa drag.

I ett regelbaserat system kallas denna metod för datadriven eller framåtkedjad inferens (forward chaining), eftersom systemet försöker uppnå ett okänt mål baserat på kända fakta. I kontrast till detta finns bakåtkedjad inferens (backward chaining), där man försöker bevisa en hypotes.

I spelet tillämpar vi ett bakåtresonemang – vi börjar med en spelsituation där datorn har vunnit och försöker sedan dra

slutsatser om vilka drag som krävdes för att nå dit.

Även om maskininlärning idag är den dominerande metoden inom artificiell intelligens och i många områden har ersatt expertsystem och andra traditionella AI-metoder, kan den inte helt undvika behovet av kunskapsrepresentation. I teknologier som neurala nätverk sker detta dock implicit, vilket gör att de klassas som subsymboliska system. I stället för att explicit representera regler tränas nätverket till ett systematiskt beteende, där

en form av implicit kunskap om underliggande samband byggs upp.

Det är dock svårt att få insyn i hur lösningarna faktiskt tas fram i dessa nätverk, eftersom de regler som styr datan bara representeras indirekt – exempelvis genom vikterna mellan neuronerna och deras aktiveringströsklar.

Mer information om neurala nätverk finns i aktiviteten #deeplearning. CS4FN erbjuder också en annan aktivitet där ett expertsystem för luffarschack modelleras på ett unplugged sätt.

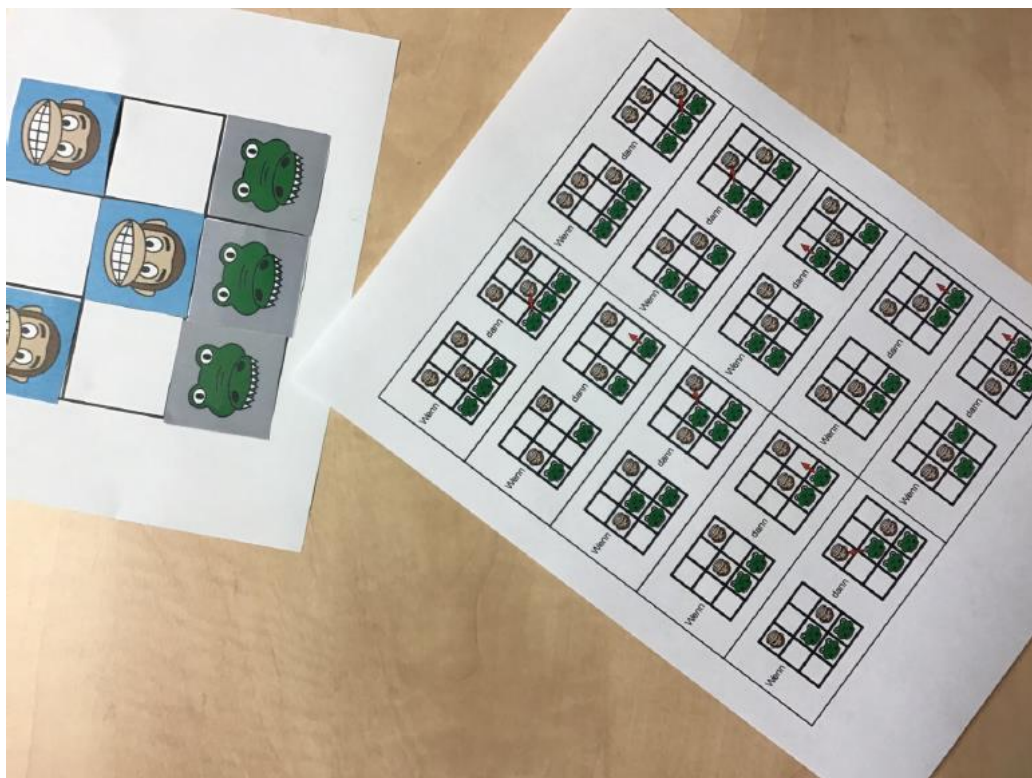


Fig. 14: Game setup: The rules for the person taking over the computer are clearly defined.

Figur 14: Speluppställning: Reglerna för personen som tar över datorn är tydligt definierade.

Turing testet

"Åh! Jag är så glad att ingen visste att jag är en dator!"

Målgrupp: Högstadiet och gymnasiet

Vad handlar det om?

Hur måste en maskin bete sig för att betraktas som intelligent? Vad innebär egentligen artificiell intelligens? Forskare har arbetat med dessa frågor sedan artificiell intelligens började utvecklas. År 1950 kom Alan Turing på en idé om hur man kan avgöra om en maskin är intelligent – det som idag kallas Turing testet. Denna aktivitet återskapar Turing testet tillsammans med elever och syftar till att stimulera diskussion om huruvida datorer verkligen kan uppvisa något som liknar mänsklig intelligens. Aktiviteten visar också hur lätt det är att bli lurad av en maskin genom noggrant valda exempel på "intelligens".

Idéerna bakom detta

- Intelligent system använder specifika strategier för att efterlikna mänskligt beteende.
- Särskilda metoder krävs för att utvärdera maskiners intelligens.
- Definitionen av (artificiell) intelligens är inte tydlig.

Vad du behöver

- Arbetsblad/bilder med förutbestämda frågor från Turing testet för hela klassen.
- En kopia av svaren på frågorna i Turing testet.
- Fyra frivilliga elever i rollerna som dator (1x), människa (1x) och springare (2x).

Here's how it works

I den här aktiviteten får eleverna spela en fråga-svar-lek där de försöker avgöra vem som är en människa och vem som är en dator – baserat på hur de svarar på frågor. En elev spelar rollen som en dator, medan en annan svarar som en vanlig människa. Resten av klassen ställer frågor och analyserar svaren för att lista ut vem som är vem. Kan ni avslöja "datorn" genom smarta frågor?

Denna aktivitet kommer ursprungligen från det ursprungliga CS Unplugged-materialet. Dessa är licensierade under Creative Commons CC-BY-SA av Bell, Witten och Fellows. Det ursprungliga materialet har anpassats i denna beskrivning, som också är licensierad under CC-BY-SA.

Sammanhang

I århundraden har filosofer diskuterat huruvida en maskin kan uppnå mänsklig intelligens eller om den mänskliga hjärnan kanske bara är en mycket avancerad maskin. Vissa anser att artificiell intelligens är en absurd idé, medan andra tror att vi så småningom kommer att utveckla maskiner som är lika intelligenta som vi människor.

Artificiell intelligens har stor potential, men samtidigt väcker idén om intelligenta maskiner också oro och rädslor.

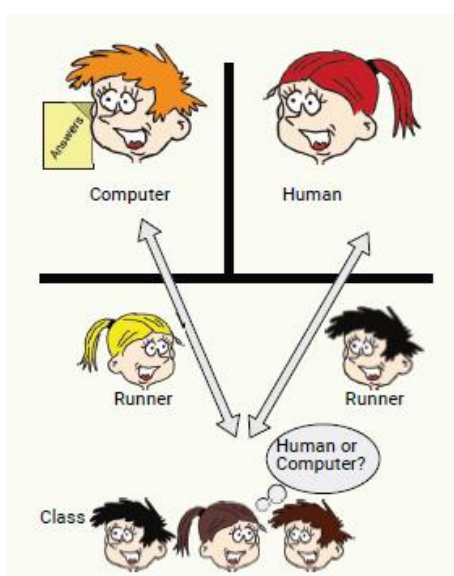


Fig. 15: Design av Turing testet

Aktivetsbeskrivning

Innan spelet börjar, diskutera med eleverna om de anser att datorer är intelligenta eller om de tror att datorer någon gång kommer att bli intelligenta. Be dem fundera på hur man kan avgöra om en dator är intelligent och introducera kort Turing testet, som simuleras i denna aktivitet.

För att förbereda aktiviteten väljs fyra frivilliga elever som tar på sig rollerna som en dator och en människa (se Figur 15). Dessutom finns det två springare som ser till att spelet genomförs rättvist och är utrustade med ett papper och en penna för

att anteckna svaren. Roller som "människa" och "dator" tilldelas i hemlighet av läraren innan dessa två elever lämnar klassrummet och går in i två separata rum (alternativt kan skärmar användas, men det är viktigt att eleverna inte ser varandra).

Eleven som spelar datorn får en kopia av svaren på frågorna i Turing testet. Varje springare är ansvarig för en av rollerna, vilken roll de har hålls också hemligt. Klassen ska nu lista ut vilken av eleverna som spelar datorn. För att göra detta väljer de en fråga per runda från arbetsbladet som ska ställas till både datorn och människan.

Efter att en fråga har valts ska eleverna förklara varför de anser att denna fråga är lämplig för att skilja datorn från människan. Denna argumentation är den centrala delen av uppgiften, då klassen reflekterar över hur svaren från en människa och en "intelligent" dator kan skilja sig åt.

Därefter ställer springarna frågan till sina klasskamrater i de andra rummen och svaren förs tillbaka till klassen. Människan måste svara kort och ärligt – med andra ord ge ett mänskligt svar. Datorn väljer däremot det lämpliga svaret från arbetsbladet. Om instruktionerna är skrivna i kursiv stil måste datorn själv komma på ett svar (t.ex. den aktuella tiden). När svaren överförs måste springarna vara särskilt noga med att inte avslöja vem de interagerar med.

Klassen diskuterar nu vilket svar som troligen kommer från en dator. Upprepa processen med några fler frågor, om möjligt tills klassen kan fatta ett tydligt beslut om vem som är datorn. Om klassen inte på ett tillförlitligt sätt kan skilja mellan människa och dator har datorn klarat Turing testet.

Bakgrund

Även om inget nuvarande datorprogram har något som liknar allmän intelligens, är frågan om datorer i grunden är kapabla till det fortfarande obesvarad. Detta beror

främst på att själva definitionen av intelligens är kontroversiellt diskuterad.

Mot denna bakgrund föreslog den brittiske matematikern Alan Turing 1950 en metod för att fastställa en maskins intelligens utan att behöva en exakt definition av intelligens. Detta så kallade Turing test låter datorn demonstrera sin "intelligens". Testets scenario liknar den aktivitet som beskrivs ovan: en frågeställare interagerar både med en människa och en dator via en chatt. Om hen inte på ett tillförlitligt sätt kan skilja mellan de två har datorn klarat Turing testet. Eftersom kommunikationen sker via chatt kan datorn inte avslöja sig genom fysiska egenskaper, såsom röstläge.

Ett välkänt exempel på ett sådant interaktionssystem är chatboten Eliza. De svar som en elev ger i rollen som dator liknar de svar som ett "intelligent" datorprogram kan ge. Vissa svar avslöjar snabbt datorn – en människa kan knappast ge kvadratroten av 2 med 20 decimalers noggrannhet. Andra frågor, där datorn alltid använder ett visst svarsmonster, avslöjar den först efter en viss tid. Till exempel är svar på frågor av typen "Tycker du om X?" inte misstänkta när de betraktas enskilt.

Men om flera frågor av denna typ kombineras blir det tydligt att datorn använder en formel för att generera svar från frågorna.

Svaren kan också visa att datorn har misstolkat en fråga, även om detta även kan hända en människa. Många svar är vaga och vid vidare efterforskning skulle det bli tydligt att datorn egentligen inte förstod innehållet i frågan. Dessutom är det ofta säkrare för datorn att svara "Jag vet inte" (t.ex. på frågan om kvadratroten av 2). Detta kan imitera mänskliga drag, men kan också avslöja datorn om denna taktik används för ofta eller på alltför enkla frågor. Fördröjda och felaktiga svar, till exempel på aritmetiska problem, kan också vilseleda frågeställaren under en längre tid.

Datorer kan alltså ge sken av att kunna föra samtal genom formelbaserade svar, genom att spegla samtalspartners uttalanden, reagera på nyckelord, använda idiom och återkoppla till tidigare ämnen, men detta är bara en fasad som är lätt att genomskåda.

Fler idéer

Länkar och detaljer om dessa aktiviteter finns på vår webbplats.

Ansiktsigenkänning

Vår ytterdörr kan skilja oss från brevbäraren, och vår fotohanteringsapp kan automatiskt tagga våra vänner – ansiktsigenkänning är en vanlig tillämpning av AI. Tekniken måste vara flexibel nog att känna igen oss både på vintern när vi har mössa och på sommaren när vi bär solglasögon. Denna aktivitet illustrerar principen genom tecknade figurer.

Monkey, Sherlock Monkey

Hur kan kunskap representeras på ett sätt som gör att en dator kan "förstå" den och dra logiska slutsatser? Logik och formell kunskapsrepresentation spelar en stor roll här! AI-system är därför inte riktigt "intelligenta", utan använder smarta sätt att strukturera och hantera kunskap. Denna typ av kunskapsrepresentation kan också användas i logiska pussel. Sådana pussel kräver att man kombinerar olika fakta enligt bestämda regler för att sedan hitta en lösning.

Brain-in-a-Bag

I denna aktivitet simulerar eleverna hur ett neuralt nätverk fungerar genom att använda snören och toalettrullar. Det färdiga nätverket kan sedan spela ett spel.

Oövervakad inlärning

Förutom övervakad inlärning och förstärkningsinlärning finns det även metoder för oövervakad inlärning. Här lär sig datorer utan att ha förutbestämda mål eller belöningar. Genom att analysera en samling data kan datorn själv identifiera kategorier (t.ex. kunder med hög köppotential i webbutiker) eller avvikelser (t.ex. misstänkta aktiviteter på webbsidor).

Ett sätt att illustrera detta är att rita ett koordinatsystem med krita på skolgården. Låt eleverna placera sig i rutnätet utifrån två olika egenskaper. Beroende på vilka axlar som används kan de hitta grupperingar (kluster), men också avvikare (anomalier).



Imprint

Editor:

Professorship for Computer Science Education
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3

91058 Erlangen

<https://aiunplugged.org>

Editing and design:

Annabel Lindner, Stefan Seegerer

Translation:

Jahangir "Jay" Almasi 2025

All text och grafik i denna broschyr (förutom FAU-logotypen eller där annat anges) är licensierade under **CC BY-NC 3.0**. Detta innebär att du får redigera, reproducera och distribuera materialet i valfritt format eller medium, men **inte i kommersiellt syfte**. Det enda kravet är att du anger upphovspersonen.