AI: Sådan laver forskere computere, der kan tænke

Deep learning kan ikke lave fordomsfri algoritmer. Derfor er forskere i gang med at udvikle nye metoder, som kan lære computerne at tænke over beslutningerne.

For at opnå det bedste resultat i implementeringen af kunstig intelligens i samfundet må vi udvikle de forskellige teknikker i stedet for at fokusere på en enkelt. Det mener Luís Cruz-Filipe. (Foto: Shutterstock)

Luís Cruz-Filipe
Adjunkt, Institut for Matematik og Datalogi, Syddansk Universitet

02 marts 2020 FORSKERZONEN INNOVATION ROBOTTER

Vi lever i en tid, hvor computere ikke kun er en uundgåelig del af vores hverdag, men også begynder at erstatte mennesker på opgaver, som vores liv afhænger af.

Vi stoler på computere til at styre næsten al lufttrafik; vi bygger biler, som køres af computere; vi bruger computere til at give medicin til patienter på hospitaler; og vi

begynder at bruge computere til at diagnosticere sygdomme.

Men hvordan ved vi, at vi kan stole på de programmer, der understøtter alle disse aktiviteter? Hvordan kan vi være sikre på, at vores fly ikke styrter ned? Hvordan ved vi, at vores selvkørende bil ikke kører den gamle dame over, der krydser gaden?

Er det virkelig bedre at stole på en computer end på en sygeplejerske eller en læge?

Svaret er ja – hvis de programmer, der udfører disse opgaver er certificerede.

Er det et fly? Er det en fugl? Ja, det er en fugl

Fakta
Forskerzonen

De traditionelle systemer i kunstig intelligens er baserede på deep-learning. Princippet ved deep-learning er, at en computer kan lære fra eksempler – ligesom børn gør.

For eksempel kan vi træne en computer til at genkende billeder af fugle ved at vise den et stort antal billeder, hvor nogle viser fugle, og andre ikke viser fugle.

For hvert billede skal computeren gætte, om der er en fugl på billedet eller ej, og den får at vide, om den har gættet rigtigt. Computerens gæt bliver bedre og bedre, indtil den (næsten) ikke tager fejl mere.

På samme måde kan vi træne en computer til for eksempel medicinsk diagnose, sådan at den faktisk laver færre fejl end en menneskelig læge.

Der er dog to problemer. For det første kan der forekomme fordomme i computerens svar, og for det andet er der ingen forklaringer på dens svar.

LÆS OGSÅ: Hvad er kunstig intelligens egentlig?

Kønsdiskriminerende robotter

Fordomme har typisk årsag i skjulte mønstre i de eksempler, man har brugt til at træne computeren med, som ikke var tydelige i forvejen.

Et skræmmende eksempel på, hvor galt det kan gå, skete dengang Amazon forsøgte at udvikle et computerprogram, der kunne kigge igennem ansøgeres cv'er til en stilling.

Efter flere års udvikling blev det tydeligt, at programmet fravalgte kvinder.

Det skyldtes, at programmet var blevet trænet med data fra de forrige 10 års ansatte. Da industrien er domineret af mænd, 'lærte' computeren, at kvinder ikke var gode kandidater.

I dette tilfælde fandt man heldigvis tidligt ud af, at der var et problem, og hvad dets årsag var – og programmet kom derfor aldrig i brug. Men det er ikke altid så nemt at forstå, hvad der er gået galt.

Vi bliver nødt til at vide, hvorfor fejlene sker

Da computeren lærer ved at finjustere nogle tal, som vi basalt set ikke kender betydningen af, er der principielt ingen måde at forstå, hvorfor den svarer, som den gør.

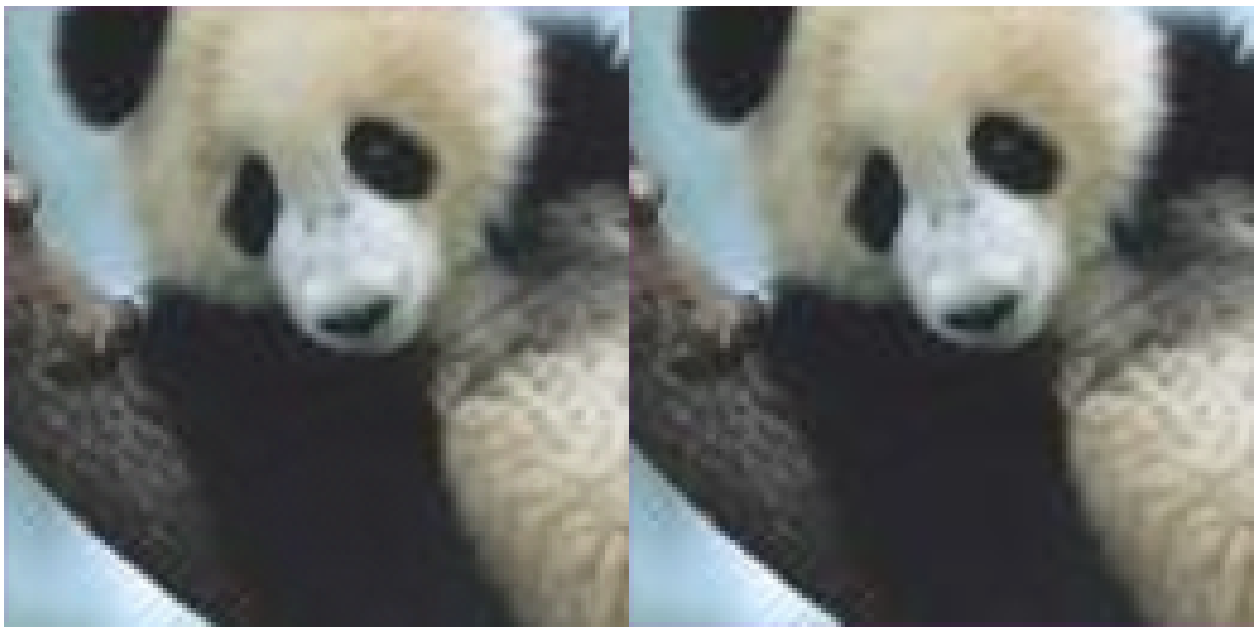
Og det er et stort etisk problem. Det ville simpelthen ikke være acceptabelt at fortælle pårørende, at patienten døde, fordi computeren gav et forkert svar af en ukendt årsag.

Deep learning har haft stor succes på mange områder. Dog har disse bekymringer motiveret en søgning efter andre måder til at udvikle systemer, der får følsomme opgaver.

De mest populære alternativer falder i to kategorier:

- Programmer, der kan forklare deres svar (explainable/forklarende AI).
- Programmer der ikke kan tage fejl (certificerede programmer).

Idéen bag begge alternativer er, at viden programmeres direkte ind i programmet.



Disse to billeder viser, hvordan man kan snyde et program baseret på deep-learning. Selvom de ligner hinanden meget, vil sådan et program mene, at det til venstre viser en panda, og det til højre viser en gibbon. (Billederne bruges med tilladelse af Ian Goodfellow, Nicolas Papernot, Sandy Huang, Rocky Duan, Pieter Abbeel og Jack Clark)

I stedet for at vise computeren eksempler og lade den finde ud af, hvad der er rigtigt, og hvad der er forkert, lærer vi computeren, hvordan man tænker om problemet – for eksempel hvordan man beslutter sig for en flyrute, eller hvordan man diagnosticerer en bestemt sygdom.

'Du skylder mig en forklaring, computer!'

Med andre ord: i stedet for at computeren lærer, som børn lærer (via eksempler), lærer den som voksne (via forklaringer og metoder). Derudover er viden organiseret sådan, at computeren også kan forklare sin tankegang samtidig med, at den giver et svar.

Hvis vi tvivler på svaret, har vi nu mulighed for at analysere forklaringen og forstå, om computeren måske har lagt mærke til noget, vi har overset. Vi kan også finde ud af, at der er en fejl i computerens program, som vi nu også har information nok om til at rette.

Ideelt vil vi også kunne bevise, at der ikke findes fejl i programmet. Her taler vi ikke om, om vi har testet programmet og ikke har kunnet finde fejl: Vi *ved* derimod, at der er et matematisk bevis for programmets egenskaber, og det bevis har vi typisk gennemført på papir.

LÆS OGSÅ: Pilot banker kunstig intelligens i droneræs

Et logisk sprog

Dette gør vi ved først at beskrive de egenskaber, vi ønsker (at flyet ikke skal styrte ned, at bilen ikke skal køre den gamle dame over, at patienten ikke får for meget medicin eller en forkert diagnose) i et bestemt sprog.

Vi kalder dette sprog for en logik og de 'sætninger', der beskriver egenskaber, for udsagn eller formler.

I logikken laver vi også en model af programmeringssprog, det vil sige, at vi har udsagn, der beskriver, hvordan programmet virker – for eksempel hvis programmet gemmer værdien fem på en variabel x og derefter læser x's værdi, så får det garanteret 5 tilbage.

Det næste trin er at bruge logisk værktøj for at vise, at de egenskaber, vi har beskrevet, gælder.

Her bruger vi også specialiserede computerprogrammer, der samarbejder ved at tjekke beviserne - muligvis kan de selv klare nogle trin uden hjælp.

Hvordan ved vi, at disse programmer ikke laver fejl? Fordi vi har tidligere bevist matematisk, at de er fejlfrie.

Typisk er der en lille del af programmet, som vi har bevist korrekt på papir, og derefter brugt til at bevise, at resten af programmet også er korrekt.

Det bliver ekstremt kompliceret

Processen kan blive enormt kompleks, da vi kan lave en kæde af programmer, som hver især tjekker hinanden.

Det er ikke usædvanligt at have tre eller fire niveauer, hvor program A bevises fejlfrit ved brug af program B, som bevises fejlfrit ved brug af program C, som...

Bare for at have en idé om, hvor stort problemet kan blive: Listen med alle egenskaber for software til lufttrafikkontrol fylder cirka 1000 sider; og nogle af de største beviser, der findes kræver mere end 1 petabyte (dvs. 1 million gigabytes) for at skrives ned.

Det er heller ikke altid muligt at bevise alle egenskaber, som man ønsker skal gælde. Derfor er det altid en god idé, at programmer stadigvæk returnerer en forklaring på sit svar.

LÆS OGSÅ: Bør vi frygte kunstig intelligens?

Hver løsning har sine svagheder

I nogle tilfælde er det også muligt, at vi ikke kan certificere det program, som beregner svaret, men stadigvæk kan garantere at svaret er korrekt ved at tjekke forklaringen.

Og ja, det kan også gøres ved brug af endnu et computerprogram.

Så vi kan køre med to programmer: Et, der finder svar sammen med forklaringen, og som muligvis tager fejl en gang imellem og et andet, der tjekker forklaringerne og fortæller os, om svarene er korrekte - og som vi har bevist aldrig laver fejl.

Deep learning, explainable AI og certificerede programmer er meget forskellige måder at danne computerprogrammer til at løse komplicerede opgaver, og hver har sine fordele og ulemper.

Deep learning har haft stor succes ved at oplære computere bedre end mennesker på mange områder (er det en fugl eller ej), men vi kan befinde os i et etisk rod, hvis programmet svarer forkert (medicinsk diagnose).

Certificerede programmer kan ikke tage fejl, men det kan være enormt besværligt (eller umuligt) at skrive og bevise alle egenskaber, som man ønsker skal gælde.

Explainable AI kræver til gengæld, at vi analyserer forklaringen for at bestemme, om vi stoler på svaret eller ej.

Det bedste ville være en kombination

Det bedste scenarie er et, hvor de forskellige metoder kombineres på basis af, hvor meget de kan bruges, og hvor vigtigt det er, at der ikke sker nogen fejl.

I medicinsk diagnose får man de bedste resultater ved at bruge både et computerprogram og en menneskelig læge.

Når lægen og computeren ikke er enige, kan lægen analysere computerens forklaring for at forstå dens svar – og bestemme, hvem der har ret.

Sådan bliver det faktisk næsten umuligt at tage fejl – det sker kun, hvis både læge og computer uafhængigt af hinanden når den samme forkerte konklusion (eller hvis lægen laver fejl i sin analyse af, hvem der har ret).

Til gengæld, da vi kan bevise de vigtigste egenskaber af luftfartskontrolsystemet, har vi ikke brug for at tjekke dets individuelle svar. Og ved en app, der analyserer et billede af en blomst og fortæller, hvad type blomst den er, kan vi godt acceptere, at den tager fejl en gang imellem.

Derfor bliver vi nødt til at udvikle de forskellige tekniker for at få de bedste resultater, i stedet for kun et fokusere på én af dem.

LÆS OGSÅ: Kunstig intelligens: Bliver mennesker overflødige?

LÆS OGSÅ: Kunstig intelligens lider også af menneskelige fejl

Kilder

- Luís Cruz-Filipes profil (SDU)
- Mere om AI og etik