

SYNSPUNKT AI



Commodity AI – hvor svært kan det være?

Igennem en årrække har man på Erhvervsakademi Aarhus arbejdet med at gøre machine learning-projekter mere tilgængelige for studerende. De første studerende er nu ved at være igennem, og adjunkt Simon Laub fortæller om resultaterne i dette Synspunkt.

Simon Laub, Adjunkt fra Erhvervsakademi Aarhus

<https://pro.ing.dk/5276>

4. feb 2020 05:00

Ifølge ordbogen er 'a commodity' en uforarbejdet eller delvist forarbejdet vare. Korn, frugt & metaller er sådanne råvarer, mens f.eks. en service falder udenfor 'commodity'-kategorien.

Inden for it-verdenen er der på det seneste blevet argumenteret for, at AI er ved at blive en slags 'commodity' – inkluderet i et produkt som en slags halvfabrikata, andre it-services kan bruge. Og hvorfor ikke? It-programmer er ikke længere isolerede monolitter.

De fleste programmer inkluderer tredjepartssoftware, og programmører ved oftest kun, hvordan man bruger disse underprogrammer, ikke, hvordan de rent faktisk fungerer. Og de store spillere i it-verdenen (IBM, Google m.fl.) virker i disse år ganske ivrige efter hjælpe andre i gang med AI ved at gøre deres AI-komponenter tilgængelige for alle andre. Ifølge Kevin Kelly (Wired Magazine) er det nemt at forudsige, hvor vi er på vej hen: »Forretningsplanerne for de næste 10.000 start-ups er lette at forudsige: Tag X og tilføj AI«.

Hvor let er det egentligt 'at tage X og tilføje AI'? Kan alle gøre det?

Tja, som sædvanlig er djævelen i detaljerne. Hvad er 'X', og hvad mener man egentlig med AI? Hvilke færdigheder har

MEST LÆSTE

1. Energinet vil slippe elforbrugernes fortrolige data løs i anonym form
2. Vestager: GDPR-rettigheder skal virkeliggøres, hvis loven skal have troværdighed
3. Persondataskytselse og informationsikkerhed – standarden ISO/IEC 27701 binder det hele sammen

JOBFINDER

kamstrup

CRM Coordinator

LINAK 
WE IMPROVE YOUR LIFE

IT Leader for ERP and PLM

Ørsted

Product Owner for Agile Release Train

LINAK 
WE IMPROVE YOUR LIFE

Software Developer

man? Hvor lang tid har man?

Selvfølgelig: I et konkret projekt har man måske en idé om, hvor megen tid man selv skal bruge for at 'tilføje AI til X'. Måske er der én i ens team, som kan svare på det. Eller måske kan man ansætte en konsulent med præcis den rette ekspertise, der er brug for. Så langt, så godt, men hvad svarer man, hvis man mere generelt bliver spurgt, om det er muligt at 'tage X og tilføje AI' i et it-projekt, hvor ingen har en AI-baggrund og må nøjes med 'on-the-job' AI-oplæring. Kan et sådant team komme i mål med opgaven? På et par måneder? Et år?

Situationen er ikke mere hypotetisk, end at jeg flere gange i møder med små og mellemstore virksomheder har diskuteret, hvad der præcis skal til, for at man kommer i mål med at 'tage X og tilføje AI'. Og jeg har flere gange hørt mig selv sige: »Det afhænger jo af så meget. Hvem er det præcist, vi har i teamet? Hvad er X mere præcist?« osv. Ikke netop et modigt svar, og også lidt utilfredsstillende. Burde det ikke være muligt at sige noget lidt mere konkret?

Tid til nogle eksperimenter!

Når man arbejder i et uddannelsesmiljø, er det ikke så svært at finde projekter (vores 'X'er'), der kunne forbedres med lidt AI. Og det er også relativt let at finde studerende, der - som en del af et praktikophold eller en opgave - vil kunne udgøre ens team og udføre det meste af det praktiske arbejde.

Selvfølgelig ikke 100 procent et 'Tag X og tilføj AI'-projekt i en lille privat virksomhed. Men tæt nok på det til at være interessant. Kan arbejdsgivere forvente, at medarbejdere med en uddannelsesmæssig baggrund ca. som vores studerende vil komme i mål med en sådan AI-opgave? På hvor lang tid? Hvilke ekstra kvalifikationer mangler medarbejderne?

Til vores 'X' besluttede jeg at kigge på klassificering af jobannoncer. Et populært samtaleemne omkring mig for 1½ år siden, da vi begyndte at lave disse små eksperimenter. Det lød udfordrende og med en tilpas 'AI feel' omkring det. Det vil sige, at jeg koblede mig på et nationalt forskningsprojekt indenfor erhvervsakademi-sektoren vedrørende klassificering af jobannoncer.

Projektet ledes af Erhvervsakademi Aarhus og CphBusiness. Og har til formål at afdække erhvervslivets kompetencebehov inden for forskellige erhverv, brancher, geografiske lokationer, fag, virksomheder og organisationer. Denne nye indsigt i behovene bruges fremadrettet til at tilrettelægge, opbygge og ombygge udbuddet af nationale uddannelser, valgfag og kursusforløb. Og det er i den kontekst, der anvendes automatiseret tekstanalyse af



Frontend Developer
with Backend
experience

[SE FLERE](#)

[OPRET JOB](#)

jobannoncer.

Det vil sige, at vores institution og lignende institutioner i Danmark er meget interesseret i at vide, hvilken slags færdigheder arbejdsgivere leder efter. Da det grundlæggende er de færdigheder, som vi alle skal give vores studerende.

En måde at blive klogere på dette er ved at søge efter bestemte ord – f.eks. ved hjælp af regulære udtryk – i annoncerne for derved at klassificere disse annoncer. En okay tilgang, men denne relativt ligefremme proces er ikke i stand til at klassificere alle annoncer 100 procent korrekt. Nogle jobannoncer bruger ikke de præcise nøgleord, som et program leder efter. Nej, automatiseret kategorisering af sproglige tekster er ikke så let. Og da slet ikke, når vi i Danmark har omkring 17.000 forskellige jobkategorier med et utal af forskellige måder at beskrive disse job på. Så projektet havde allerede indkaldt AI-konsulenter, før vi kom til, for at se, om de kunne hjælpe med nogle af disse lidt mere vanskelige problemstillinger, man var stødt på.

Vores studenterteams skulle arbejde parallelt med denne indsats og kigge nærmere på nogle af de samme 'jobannonceklassificeringsproblemer'. Så langt, så godt, det lød bestemt som et 'Take X og add AI'-projekt, hvor AI-komponenten her ville være et tekstklassificeringsprogram. Og ja, vi var i godt selskab, da tekstbehandling er core-business i dagens store internetfirmaer, virksomheder som Facebook, Google, Twitter, Baidu, Yahoo osv.

Startende i 2018 fik fire studerende 10-ugers praktikpladser til at hjælpe os med dette 'AI naturligt sprogklassificeringsproblem. Næsten færdige med deres 2½-årige datamatikeruddannelse havde de alle en solid baggrund inden for it, men ingen erfaring med machine learning eller AI.

De to første af vore studerende besluttede at arbejde sammen og følte sig hurtigt overbeviste om, at Googles Tensorflow og neurale net var vejen frem. Uden tidligere erfaring med neurale net var deres projekt fokuseret på at lære Tensorflow nærmere at kende og få jobannonceteksterne 'præprocesseret', så de neurale net kunne arbejde med dem. Helt som forventet: Forberedelse og klargøring af data er en stor del af de fleste dataprojekter, og man bruger ofte, helt som standard, 80 procent af et projekts tid og indsats i denne fase.

Men det blev alligevel hurtigt klart, at vi løb efter for mange forskellige ting i dette første projekt. De studerende blev ikke kun bedt om at anviser en god vej frem til at 'tage X og tilføje AI', de blev også bedt om at lave en fungerende

prototype til at gøre dette. I betragtning af tidsbegrænsningerne besluttede de studerende at fokusere på at opbygge en TensorFlow-prototype – uden at tænke så meget på, om der kunne være andre, og bedre, måder fremad.

Men hvordan? I naturlig sprogbehandling antager man ofte, at ord, der bruges og forekommer i samme sammenhæng, har en tendens til at have lignende betydninger. Så det blev besluttet at inkludere et såkaldt 'word-embedding'-lag i processen. Dette lag vil så placere semantisk lignende ord som nærliggende punkter i et multi-dimensionalt vektorrum. Dvs. systemet vil lægge 'Obama' tæt på ordet 'præsident', 'medier' tæt på 'presse', 'taler' tæt på 'hilser' osv., efter at man har trænet sit system på et stort tekstkorpus.

Meningen er så, at dette præprocesseringslag vil gøre det lettere for det neurale net bagefter at klassificere jobannoncer korrekt. Derefter gik de studerende i gang med de tekniske detaljer vedr. opsætning af selve det neurale net i TensorFlow. Tuning af hyperparametre til det neurale net, ting som: a) Hvor mange skjulte lag skal det neurale net have, med hvor mange neuroner? b) Hvilke aktiveringsfunktioner skal man bruge i systemet? c) Hvordan skal man optimere læringsprocessen for det neurale net? etc.

Overvældet af TensorFlow

10 uger går rigtigt hurtigt, når man har det sjovt! Og det første studenter-projekt var glade for at ende med en fungerende prototype baseret på TensorFlow. Et neuralt net var blevet trænet på jobannoncer (mærket med håndkraft til de korrekte kategorier) og kunne derefter fortælle os med 90 procents nøjagtighed, om en ny jobannonce var en 'Java-programmør'-annonce eller ej. Så langt, så godt.

De næste to studerende besluttede at arbejde hver for sig og var blevet bedt om at forbedre den eksisterende prototype.

Den ene studerende besluttede hurtigt, at 'Adam Optimizers', 'Lelu-aktiveringsfunktioner' mm. i TensorFlow-universet var en noget stor mundfuld, og besluttede derfor at skifte til et andet praktikprojekt. Den anden studerende kæmpede bravt videre, men blev lettere overvældet af mængden af TensorFlow-ord, han skulle lære bare for at forstå, hvad den eksisterende prototype gjorde. For slet ikke at nævne, at han rent faktisk var blevet bedt om at forbedre denne prototype.

At sige, at der var mange veje at gå herfra, ville være en bevidst underdrivelse. Et nyt neuralt net kunne fungere i et ensemble med det gamle neurale net og addere dets resultater til de gamle systems resultater. Dvs. man kunne træne et nye neuralt net på nye klassifikationer, såsom 'hvor

teknisk er jobbet? Et butiksassistent-job ville være 'ikke så teknisk', hvorimod vores Java-programmørjob ville få klassificeringen 'meget teknisk'. Sammen med det gamle neurale net kunne man så håbe, at man herved kunne forbedre det samlede systems generelle nøjagtighed.

Derudover kunne man sikkert også tilføje resultater fra det gamle 'ikke-AI'-system til ensemblet. Og ja, så er der jo også mange andre systemer, man måske kunne arbejde videre med. Tænk bare på f.eks. Facebooks open source-produkt 'FastText', der lover at kunne udføre generel tekstklassificering. Alle disse systemer kunne potentielt føjes til en ensemble-løsning. For slet ikke at tale om, at man også kunne arbejde med egne modeller, mere eller mindre sofistikerede. Simple mekanismer som bare at tælle forekomster af ord i en annonce, i en 'bag of words'-model, kunne måske også vise sig at være nyttige?

Den typisk tjekliste

Den studerende blev bedt om at se på den typiske tjekliste for machine learning-projekter: 1) Hvordan en minimumsløsning, der indfrier de forretningsmæssige mål, skal se ud 2) Hent og klargør data 3) Udforsk data og undersøg, hvordan problemet kan løses manuelt 4) Præprocessér data for at gøre data mere velegnet til processering i machine learning-regi 5) Udforsk mange forskellige modeller til at løse problemet.

Så, ja, der var tydeligvis mere end nok at arbejde på her. Igen på 10 korte uger.

Den rette måde at fortsætte på? En ordentlig dataanalyse, hvor man fuldt ud forventer, at denne vil tage de sædvanlige 80 procent af et data-science-projekts tid. Sandsynligvis. Men hvad så med resultater? Vi ønsker vel alle at præsentere resultater, nye overraskende indsigter fra en dataanalyse, prototyper eller lignende. Hurtigt ... Også i dette studenter-projekt var der nok at se til.

Når man tænker lidt over det, var situationen sandsynligvis ikke så forskellig fra noget, man kan opleve i de fleste små firmaer, der skal håndtere et nyt problem. Hvordan kan man producere resultater, samtidig med at man laver research og lærer undervejs? Vi fik en snak om det. Og blev enige om, at man ikke kan gøre alting på en gang.

Den studerende var meget ivrig efter at producere resultater, så vi fik besluttet, at han skulle fokusere på at foretage en lille gradvis forbedring af den eksisterende prototype. Men for at komme helt igennem de første fem trin på tjeklisten behøver man ikke kun støtte fra sin ledelse, og tid nok, man behøver også erfaring for at modstå pres (fra sig selv), der handler om, at man godt kan

udsætte resultat-fasen lidt. Selvom alle selvfølgelig helst vil præsentere resultater. Jo før, desto bedre.

Det endte med, at studenterprojekt nummer to kom frem til en forbedret prototype baseret på et ensemble af neurale net, der denne gang var i stand til at klassificere fire jobgrupper med en højere nøjagtighed end den første prototype. Hvilket såmænd var et ganske godt resultat.

Tilføjelse af flere tricks til ensemblet på en inkrementel måde vil derefter være et naturligt udgangspunkt for kommende studenter-projekter.

Men 'commodity AI'? Ikke rigtig. 'At tage X og tilføje AI' lyder muligvis let, men er sandsynligvis stadig ikke så nemt; især da ikke, hvis man ikke har en masse ML/AI-erfaring med i rygsækken endnu. For slet ikke at tale om, at man selvfølgelig stadig har brug for en masse god gammeldags it-erfaring. Vi er stadig ikke der, hvor det hele bare er superlet

...

I de senere år har alle talt om, at vi alle sammen skal være agile og have prototyper oppe at køre hurtigt. Men da ting som dataanalyse og dataforberedelse for det meste tegner sig for 80 procent af arbejdet i de fleste dataprojekter, vil små eksperimenter i en Jupyter-notebook (yndlingsværktøj i mange datascience-projekter) eller lign. nok være en realistisk forventning i forhold til den første prototype. Og selve analyse-fasen virker i hvert fald så vigtig som nogensinde. Hvor det før var en dårlig idé at reducere omkostningerne i analysefasen (som procent af de samlede projektomkostninger), virker det nu som en garanti for, at man ikke finder den helt gode løsning. Jvf. f.eks. Bjarne Kousholts bog 'Projektledelse – teori og praksis', som nævner, at under en bestemt tærskel, f.eks. 10 procent af de samlede omkostninger, vil hele ens projekt lide og sandsynligvis ende med at gå massivt over budgettet ...

Så jo, effektiv projektstyring er naturligvis også en god idé i 'tage X og tilføje AI'-projekter. Ligesom det er vigtigt at være præcis og realistisk med hensyn til, hvad man kan opnå på kort tid. 'At lære på jobbet' er fint, men det tager tid. Eksperimenter kan være nødvendige, men det tager naturligvis også tid, der også skal inkluderes i ens planer. Lige gyldigt hvor ivrige alle er efter at komme med resultater, samtidig med at man lige laver nogle eksperimenter og lærer lidt undervejs ...

Så hvor endte vi med 'Tag X og tilføj AI. Kan alle gøre det?'-spørgsmålet?

Kedeligt nok er svaret sikkert stadig: 'It depends'. AI/ML-erfaring er godt. Generel it-erfaring er godt. Men der er

stadig et stykke vej fra at kode og finde kode-fejl i traditionelle it-programmer til at snuse rundt i data, træningsæt og testsæt, labels og klassificeringer i et AI-univers. Så jo, realistisk projektplanlægning er så vigtig som nogensinde.

I vores lille team fortsætter nye studerende i dette projekt med at bygge oven på, hvor de gamle slap, og vi har også fundet et nyt AI-domæne at kaste vore studerende ud i her i foråret. Hvor vi så fortsætter med at indsamle erfaringer og forhåbentlig hjælper med til at fortælle historien om, at det er muligt at komme i gang, selvom man ikke er Google eller IBM. Og kommer man først i gang, har Kevin Kelly selvfølgelig ret: En masse nye og meget spændende ting bliver bestemt mulige. Ting, der ikke var mulige for kun få år siden.

AI

Simon Laub
