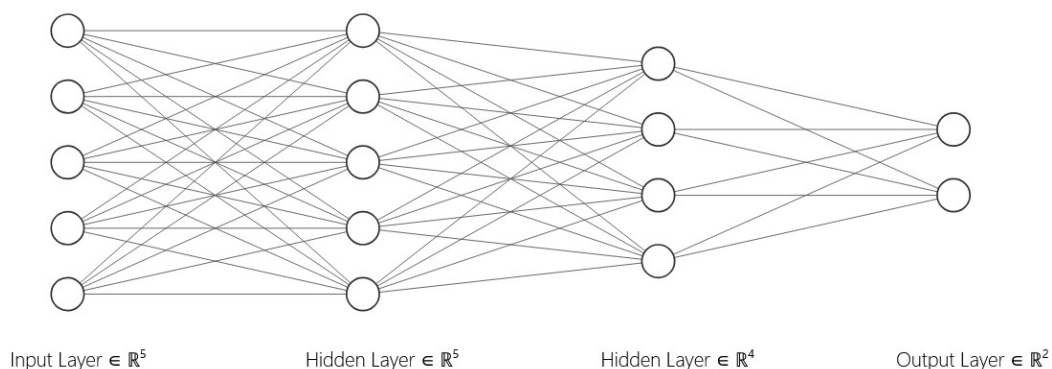


Machine learning

Machine learning (ML) er et stærkt værktøj til dataanalyse, og bliver brugt meget til at lave forudsigelser og løse komplekse funktioner som ikke nødvendigvis har en analytisk løsning (f.eks. differential ligninger). I virkeligheden er det faktisk et minimeringsproblem hvor det man vil mindske kaldes for **Cost function** (normal matematisk er der **loss function**). Cost funktionen kan i princippet være alt, men vil typisk inden for machine learning være: **mean square error** (MSE), **mean absolute error** (MAE), **cross-entropy** eller **hinge-loss**, altså kigger man primært på at minimere fejlen.

En machine learning algoritme består af nogle lag (layers) som består af en eller flere noder, ofte visualiseret som forgrenede træer af noder. Det første lag der håndterer inputs, bliver kaldt Input laget. Det lag der står for at komme med et konkret resultat kaldes for Output laget. Alle lag i mellem bliver omtalt som skjulte lag. Et network med to skjulte lag kan ses neden under.



Når ML algoritmer udvikles, udtænkes det fra et princip om at input af informationer der skal behandles af algoritmen, passerer en serie af disse noder i et forsøg på at identificere sammenhænge eller mønstre samt hvad der er mest effektivt at gøre med informationen. Måden informationen passerer gennem noderne er, rent konceptuelt blevet udviklet ud fra forestillingen om et menneskes neurale netværk. Her kommunikerer neuroner og nervebaner sanseindtryk og behov gennem kroppen og sørger for at alle signaler ender i hjernens beslutningsprocesser eller i kroppens organer der handler ud fra det stimuli de udsættes for. Tænk derfor delkomponenterne i en ML algoritme som en række processeringer, beslutninger om hvad der skal gøres i det næste trin og i sidste ende en unik sti gennem det nodetræ der udgør summen af en ML algoritme. På den måde kan en node i et algoritmisk træ sammenlignes med et neuron i menneskekroppen.

Et neuron vil typisk give sit output til mere end et neuron, og få input fra mere end et neuron. Neurons input er derfor en vægtet gruppe af alle inputs, mens dens output er en enkelt værdi. Neuronen får inputtet, bruger en **activation function** og returnerer sit output. Minimerings opgaven går da ud på at finde de vægtninger som skaber den mindste fejl (minimere cost funktionen hvor vægtningen er variabelen). Når læringsprocessen er færdig, gemmer man neuronstrukturen, samt de vægtninger man fik ud. Derfor kan man se machine learning som en enkelt funktion, der består af mange små funktioner.

Forskellige former for Machine learning

Der er tre hovedkategorier: **Supervised learning**, **Unsupervised learning** og **Reinforced learning**. Forskellen mellem de tre er hovedsageligt hvad de skal bruges til, og hvilke typer af datasæt der er til rådighed. Derved ændres læringsparadigmet sig baseret på typen af "feedback" man kan supplere systemet med.

Supervised learning dækker over ML algoritmer der er blevet trænet i de mønstre/sammenhænge de skal se ud fra datasæt der er gennemgået af mennesker. Her kunne et eksempel være anvendelsen af forudsigelse af ondartede tumorer i kræftbehandling, hvilket historisk set er blevet analyseret af læger. I dette tilfælde ved udviklerne allerede hvad det er for et mønster der kigges efter i inputtet, som ofte er tilfældet i supervised learning. Her vil en algoritmisk model få et datasæt der er labeled med ondartede og godartede tumorer, hver node vil være designet ud fra et kendetegn ved en ondartet tumor. Hvis algoritmens output er det samme som det labbelede datasæt har algoritmen truffet de samme beslutninger som de læger der diagnosticerer patienter med cancer.

Unsupervised learning dækker over ML algoritmer hvor man ikke på forhånd har klassificeret dataet. Herved lader man systemet selv finde relevante grupperinger og mønstre. Disse algoritmer bruges til f.eks. cluster analyser, hvor man kan bruge en k-means algoritme til at finde k grupper af data. Man kan også bruge den til dimmensionalitetens reduktion ved f.eks. independent component analysis (ICA) eller principal component analysis (PCA). Disse algoritmer har et hav af forskellige brug, lige fra stemme genkaldelse, billede generering og støj reduktion i signal analyse. I unsupervised learning vil der typisk være backpropagation af data gennem neuronetværket, men det varierer efter netværket.

Reinforcement learning dækker over ML algoritmer der er blevet trænet i at udfører en sekvens af handlinger der maksimerer dens "belønning". På den måde skaber man en maskine der kan træffe intelligente beslutninger. Dette gøres ved et såkaldt straffe- og belønningssystem. Her har man ikke nødvendigvis labelled data, men data som beskriver det stadie systemet er i. Helt konkret splitter man systemet op i et sæt af

"actions" og "states" og diskrete skridt f.eks. tid. States bliver brugt af systemet til at fortælle hvad der er af mulige "actions" den kan vælge, og "actions" er en beskrivelse af selve handlingen. Rutinen går derfor ud på at give systemet et state. Derefter vælger systemet en tilfældig "action". I næste skridt får systemet det state der er skabt ud fra den action den foretog samt en belønning. Belønningen her dækker over både en straf og belønning. Disse er defineret af menneskerne og variere alt efter hvilken opgave systemet skal klare. Det kan f.eks. værre en skak robot der bliver belønnet for at tage modstanders brikker, skaber et bedre board state eller straf i forbindelse med dårligere board states eller at lade en brik hænge.

Når man har kørt alle skridt igennem kigger man på den samlede belønning systemet har opnået og bruger det til at sammenligne med andre iterationer. Typisk vil man sætte mere end et system i gang f.eks. 10 og vælge den bedste af de ti. Derefter tager man den som udgangspunkt men variere lidt i dens adfærd indtil man har opnået sit ønsket resultat

Yderligere information

For mere information kan følgende ressourcer benyttes:

- Deep Dive into LLMs like ChatGPT:
<https://www.youtube.com/watch?v=7xTGNNLPyMI>
- But what is a neural network? | Deep learning chapter 1:
https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=1
- Neural Networks Demystified [Part 1: Data and Architecture]:
<https://www.youtube.com/watch?v=bxe2T-V8XR&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU>
- Learning To See [Part 1: Introduction]:
<https://www.youtube.com/watch?v=i8D90DkCLhI&list=PLiaHhY2iBX9ihLasvE8BKns2Xg8AhY6iV>
- How to Build Your First Neural Network in Python and Keras:
https://www.youtube.com/watch?v=lcqgceCr_bk
- AI can't cross this line and we don't know why.:
<https://www.youtube.com/watch?v=5eqRuVp65eY>
- Tensorflow playground:
<https://playground.tensorflow.org/>
- Google Colab:
<https://colab.research.google.com/#>