

# Lab 2: Dyp Læring og CNN

## ELMED219-2026

ELMED219

Vår 2026

# Oversikt

- 1 Nevrale Nettverk
- 2 CNN: Convolutional Neural Networks
- 3 Medisinsk Bildeanalyse
- 4 Notebooks

# Inspirasjon fra Biologien

Dyp læring (Deep Learning) er basert på kunstige nevrale nettverk (ANN).

**Det kunstige nevronet (Perceptron):** Et nevron mottar input  $x_i$ , vekter dem med  $w_i$ , summerer dem, og sender dem gjennom en aktiveringsfunksjon.

$$z = \sum_{i=1}^n w_i x_i + b$$

$$a = \sigma(z)$$

- $w$ : Vekter (synapsestyrke) - dette lærer modellen.
- $b$ : Bias (terskel).
- $\sigma$ : Aktiveringsfunksjon (non-linearitet).

## Hvorfor Dyp”?

- Fordi vi stabler mange lag med nevroner etter hverandre.
- Dette lar nettverket lære hierarkiske representasjoner (kanter - $\downarrow$  former - $\downarrow$  objekter).

# Aktiveringsfunksjoner

Uten ikke-lineære aktiveringsfunksjoner ville hele nettverket bare kollapset til én lineær regresjon!

- **Sigmoid:**  $\sigma(z) = \frac{1}{1+e^{-z}}$ . Klemmer output mellom 0 og 1. (Klassisk, men problem med "vanishing gradient").
- **ReLU (Rectified Linear Unit):**  $f(z) = \max(0, z)$ .
  - De-facto standarden i moderne DL.
  - Rask å beregne, løser mange treningsproblemer.
- **Softmax:** Brukes i siste lag for multiklasse-klassifikasjon. Gir sannsynligheter som summerer til 1.

# Hvordan lærer nettverket?

## Backpropagation (Tilbakeforplantning):

1. **Forward Pass:** Data sendes gjennom nettverket, vi får en prediksjon  $\hat{y}$ .

2. **Loss Calculation:** Vi beregner feilen (Loss)  $L(y, \hat{y})$ .  
3. **Backward Pass:** Vi beregner gradienten til feilen med hensyn på hver vekt:  $\frac{\partial L}{\partial w}$ . (Kjerneregelen for derivasjon er essensiell her!).

4. **Optimizer Step:** Vi oppdaterer vektene for å redusere feilen:

$$w_{ny} = w_{gammel} - \eta \cdot \frac{\partial L}{\partial w}$$

( $\eta$  er læringsraten).

# Hvorfor CNN for bilder?

Vanlige Densenettverk fungerer dårlig på bilder fordi:

- De ignorerer romlig struktur (pixsel (0,0) er nabo med (0,1)).
- Antall vekter eksploderer (et 1000x1000 bilde til 100 nevroner = 100 millioner vekter!).

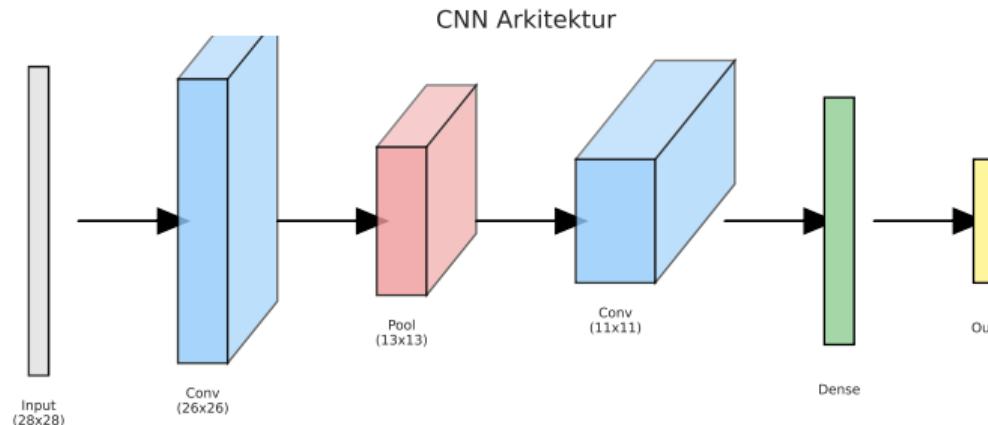
**Løsningen: Konvolusjon** Vi bruker små filtre (kernels) som glir over bildet.

- **Parameter sharing:** Samme filter ser etter samme mønster over hele bildet.
- **Translation invariance:** En katt er en katt, uansett hvor i bildet den er.

# CNN Arkitektur

## Prompt

En isometrisk visualisering av et Konvolusjonelt Nevralt Nettverk (CNN) som prosesserer et medisinsk røntgenbilde. Røntgenbildet er til venstre. Det passerer gjennom flere firkantede lag (konvolusjonslag) som blir mindre og dypere, og trekker ut egenskaper. Til høyre er resultatet en merkelapp som sier 'Diagnose: Positiv'. Ren, skjematiske 3D-stil, myk lyssetting, pedagogisk diagram."



# Byggeklossene i et CNN

- ① **Konvolusjonslag (Conv):** Trekker ut egenskaper (features) som kanter, teksturer.
- ② **Pooling Lag (Max Pool):** Reduserer størrelsen på bildet (downsampling).
  - Gjør beregningene lettere.
  - Gjør modellen mer robust mot små forskyvninger.
- ③ **Flatten Dense Layers:** Til slutt flatets feature-mappen ut til en vektor og sendes gjennom et klassisk nettverk for å gjøre selve klassifiseringen.

# Spesielle utfordringer i medisin

## Datatilgang:

- Det finnes milliarder av kattebilder, men få annoterte MR-bilder.
- Personvern (GDPR).

## Datakvalitet:

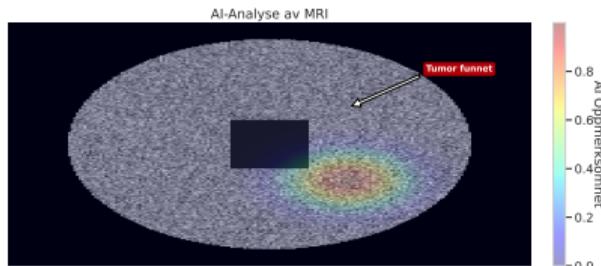
- Støy i bilder (MR/CT/Røntgen).
- Variasjon mellom ulike skannere/sykehus.

## Ubalanserte data:

- De fleste pasienter er friske (eller har ikke den sjeldne sykdommen vi leter etter).

## Prompt

*Et futuristisk medisinsk grensesnitt som viser en MRI-skann av en hjerne. Overlagt på skannen er varmekart (Grad-CAM) som viser hvor AI-en ser. Rundt skannen flyter data-widgets og Python-kodesnutter. Cyberpunk medisinsk estetikk men ren nok for en presentasjon. Høy kvalitet, 4k."*



# Tolkbarhet: Grad-CAM

Det er ikke nok at AI-en sier "Syk". Legen må vite **hvorfor**.

## **Grad-CAM (Gradient-weighted Class Activation Mapping):**

- En teknikk for å visualisere hvilke deler av bildet CNN-et fokuserte på.
- Den bruker gradientene som flyter tilbake i siste konvolusjonslag.
- Resultatet er et "varmekart" overlagt bildet.

*Viktig for tillit (Trustworthy AI).*

# Lab 2 Oversikt

Denne labben er omfattende og delt i serier:

- **A-serien (MNIST):** Vi starter med håndskrevne tall.
  - Fra Random Forest (A4) til MLP (A5) til CNN (A6).
  - Ser hvordan CNN overlegen slår de andre på bildedata.
- **B-serien (Helse):**
  - Klassifikasjon av hjertesykdom (Tabulære data).
  - EKG-analyse (Tidsserier - 1D CNN).
- **C-serien (Fordypning CNN):**
  - Bygge CNN fra bunnen i PyTorch Lightning.
  - Implementere Grad-CAM.

# Valgfrie Moduler

For de som vil gå dypere:

- **D-serien:** MR Demens-klassifikasjon. 3D-bilder! (Krever litt mer regnekraft).
- **E-serien:** Emosjonsgjenkjenning fra ansiktsbilder.
- **F-serien:** TabPFN - state-of-the-art for små tabulære datasett (Transformer for tabeller).

# Oppsummering

- Deep Learning, og spesielt CNN, har revolusjonert bildeanalyse.
- Vi bygger nettverk ved å stable enkle operasjoner (Konvolusjon, ReLU, Pool).
- I medisin er tolkbarhet (XAI) kritisk - vi må kunne se "hvorfor".

I neste og siste lab (Lab 3) skal vi se på **Generativ AI og Store Språkmodeller** - teknologien bak ChatGPT.