## 1) Hur är AI, Maskininlärning och Deep Learning relaterat?

- AI: ett jättestort paraply – allt som får datorer att ”verka smarta”.  
- ML: en del av AI där man inte programmerar regler i detalj, utan modellen hittar mönster i data.  
- DL: ännu djupare nivå – neurala nätverk med många lager. Typ som att bygga hjärnkopior i liten skala.

## 2) TensorFlow vs Keras

TensorFlow = tungt maskineri för beräkningar.  
Keras = ett förenklat ”gränssnitt” ovanpå TF som gör att man kan bygga modeller utan att drunkna i kod.  
(Själv föredrar jag Keras, mycket smidigare i början).

## 3) Parameter vs Hyperparameter

- Parameter: saker modellen själv lär sig (vikter, bias).  
- Hyperparameter: sånt jag som tränare måste bestämma innan – typ ”hur fort springer vi?” (learning rate), ”hur många lager ska vi ha?”.

## 4) Träning/Validering/Test

- Träningsdata → modellen lär sig.  
- Validering → jag kollar om den överanpassar eller om hyperparametrarna behöver ändras.  
- Test → sista bossen, helt orört tills slutet, för att se hur bra modellen faktiskt generaliserar.

## 5) Keras-koden (Dense, Dropout, EarlyStopping)

Det här är en klassisk binär klassificerare:  
- Dense(100, relu) = första lagret.  
- Dropout(0.2) = stänger av lite neuroner slumpmässigt, bra mot överanpassning.  
- Dense(50, relu) = ett lager till.  
- Dense(1, sigmoid) = utgång med sannolikhet mellan 0–1.  
- Tränas med Adam + binary crossentropy.  
- EarlyStopping = räddar mig från onödiga epoker om modellen redan slutat bli bättre.

## 6) Regularisering – varför?

För att modellen inte ska bli ett ”minneskort” som bara rabblar träningsdata. Jag vill ha en modell som klarar nya exempel.

## 7) Dropout

Som att dra ut några sladdar i nätet vid varje träningssteg → nätet blir mer robust, kan inte förlita sig på samma ”genvägar” varje gång.

## 8) Early stopping

Lite som att säga ”ok, du är färdigpluggad, lägg ner boken nu”. När valideringsresultatet slutar bli bättre → avbryt.

## 9) Populärt nät för bilder

CNN, helt klart. De har varit standard i många år. (Fast numera snackas det mycket om Vision Transformers också).

## 10) Hur funkar ett CNN?

- Första lagren → hittar enkla grejer (kanter, färger).  
- Djupare lagren → sätter ihop det till former och objekt.  
- Pooling → krymper bilden men behåller det viktiga.  
- Sen FC-lager → avgör klassen (hund, katt, bil…).

## 11) model.save / load\_model

- save() = jag fryser ner modellen i en fil.  
- load\_model() = jag tinar upp den igen och kan fortsätta jobba.

## 12) CPU vs GPU

- CPU: allsidig, få men starka kärnor.  
- GPU: tusentals små kärnor, sjukt snabb på matrisoperationer → perfekt för deep learning.