# Teoretiska frågor

## Besvarat av Emil Nilsson 2025-05-30

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

- **Artificiell Intelligens** är den generella termen för datorer som utför intelligenta uppdrag.
- **Maskininlärning (ML)** är en delmängd av AI där datorer tränas på data med hjälp av algoritmer för beslut eller prediktion.
- **Deep Learning** är en delmängd av **ML** och använder sig av neurala nätverk för att förstå komplexa data.

### 2. Hur är Tensorflow och Keras relaterat?

- **Keras** är ett bibliotek som används för att bygga och träna modeller.
- **TensorFlow** är motorn som kör dessa modeller i bakgrunden.
- **Keras** är integrerat i **TensorFlow**.

### 3. Vad är en parameter? Vad är en hyperparameter?

- En **parameter** är vad modellen har lärt sig t.ex. vikter.
- En **hyperparamterer** är hur den lär sig t.ex. LearningRate

### 4. När man skall göra modellval och modellutvärdering kan man använda tränings-, validerings- och testdataset. Förklara hur de olika delarna kan användas.

- **Träningsdata** används för att träna modellen.
- **Valideringsdata** används för att testa ens hyperparametrar.
- **Testdata** används för att utvärdera modellen på osedd data.

## 5. Förklara vad nedanstående kod gör:

```
n_cols = x_train.shape[1]

nn_model = Sequential()
nn_model.add(Dense(100, activation='relu', input_shape=(n_cols, )))
nn_model.add(Dropout(rate=0.2))
nn_model.add(Dense(50, activation='relu'))
nn_model.add(Dense(1, activation='sigmoid'))

nn_model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy' ])

early_stopping_monitor = EarlyStopping(patience=5)
nn_model.fit(
    x_train,
    y_train,
    validation_split=0.2,
    epochs=100,
    callbacks=[early_stopping_monitor])
```

- Skapar ett Sequential neural network med 3 olika lager.
  **Layer1** – 100 input noder med **relu** aktivering och **n_cols** antal features.
  **Layer2** – 50 noder i ett "hidden layer" med **relu** aktivering
  **Layer3** – 1 output node med **sigmoid** aktivering **(binär).**
- **Dropout** stänger av random nodes för att minska overfitting.
- Modellen kompileras med:
  **Optimizer='adam'** - Learningrate justeras automatiskt.
  **Loss='binary_crossentropy'** - används för att mäta skillnaden mellan de faktiska och predikterade värdena.
  **Metrics='accuracy'** - Mäter modellens prestanda under träning och validering.
- **Earlystopping** används för att stoppa träningen om den inte förbättras och sparar den bästa modellen.
- Modellen tränas på träningsdata och använder 20% av träningsdata för validering under 100 epoker. Modellen stoppas tidigare av Earlystopping om den inte förbättrats efter 5 epoker.

## 6. Vad är syftet med att regularisera en modell?

- Syftet med att regularisera en modell är att minska overfitting med t.ex. Dropout.

## 7. ”Dropout” är en regulariseringsteknik, vad är det för något?

- Dropout stänger randomiserat av en viss andel noder under träning vilket tvingar modellen att bli mer robust då den inte kan ta samma vägar.

## 8. ”Early stopping” är en regulariseringsteknik, vad är det för något?

- Early stopping används för att tidigare avsluta träningen om resultat inte längre förbättras inom angivna epoker. Vilket kan påskynda träningsprocessen.

## 9. Din kollega frågar dig vilken typ av neuralt nätverk som är populärt för bildanalys, vad svarar du?

- **Convolutional Neural Networks** är den modell som oftast används vid bildanalys.

## 10. Förklara översiktligt hur ett ”Convolutional Neural Network” fungerar.

- Ett CNN bryter ner en bild i mindre delar och lär sig detektera vissa attribut.
  En nod kan fokusera på att identifiera ögon medans en annan letar efter öron.

## 11. Vad gör nedanstående kod?

```
model.save("model_file.keras")
my_model = load_model("model_file.keras")
```

- **Rad 1:** Model.save sparar modellen tillsammans med vikterna.
- **Rad 2:** Modellen laddas in för användning.

## 12. Deep Learning modeller kan ta lång tid att träna, då kan GPU via t.ex. Google Colab skynda på träningen avsevärt. Skriv mycket kortfattat vad CPU och GPU är.

- **CPU** – Datorns processor som används för generella beräkningar. Används för att köra t.ex. Operativsystemet eller när du använder din kalkylator.
  Bra på att beräkna enstaka men komplexa beräkningar.

- **GPU –** Dators grafikkort används främst för att hantera bild och video.
  Bra på att beräkna mindre komplexa uppgifter men kan beräkna dem parallellt vilket är fördelaktigt för maskininlärning.

# Diskussion

Ur denna diskussion analyserar vi hur en Retrieval-Augmented Generation (RAG) Chatbot med fokus på flygplans manualer, specifikt checklistor för Digital Combat Simulator (DCS) kan appliceras i en verklig miljö.

## Affärsmässig

Denna chatbot kan fungera som en virtuell assistent för piloter, flyginstruktörer, flygelever eller för underhållsarbetare speciellt inom militären men med lätt modifikation även för kommersiella plan. Utan att behöva slå upp sidor i en fysisk manual kan användaren lätt ställa en fråga till den virtuella assistenten som på någon sekund kan ge svar på frågan.
Även om assistenten föredrar frågor på engelska och för bäst precision kan den lätt översätta till ett antal olika språk vilket betyder att den fysiska manualen inte heller behöver översättas och tryckas i olika språk.

Ett system som detta kan också användas ihop med Speech-to-text (STT) där användaren kan prata in sin fråga och assistenten kan ge svar antingen i text eller även Text-to-speech (TTS).

## Potentiella Affärsmöjligheter

- **Flygskolor:** Kan köpa licens för användning som extra hjälp vid inlärning.
- **Simulator Utvecklare:** Digital Combat Simulator (DCS) eller t.ex. Microsoft Flight Simulator (MSFS) kan integrera chatboten.
- **Vidareutveckling:** Vidareutvecklas för andra transportmedel som bilar, tåg, båt.

## Etiska Svårigheter

- Chatboten får under inga omständigheter hallucinera. Om den inte kan svara enligt manualen måste detta hänvisas då missvisande information kan vara livsavgörande.
- Användaren kan överförlita sig på chatboten, att boten kan ha fel måste hänvisas.
- För militärt bruk måste det finnas extra säkerhet att modellen inte kan läcka känslig information.

## Framtida förbättringar

- **Konversationsminne:** För att följa upp frågor som t.ex. “What’s the next checklist?"
- **Flerspråkighet:** För att hantera fler översättningar.
- **Text-to-speech:** Integrera TTS för användning i t.ex. simulatorer.

# Sammanfattning

Detta projekt är en praktisk presentation för hur RAG interagerar med en språkmodell.

Medans denna version innefattar en väldigt liten del av en manual så innehåller den trots det som behövs för en person som nyligen köp ett av DCS:s flygplan, nämligen Mirage F1. För personen som inte använder sig utav Virtual Reality (VR) så hade denna chatbot implementeras i simulatorn som ett mod.
Detta hade fungerat även för den som använder VR men om Speech-to-text (STT) hade inkluderats hade användaren kunnat fråga chatboten via tal istället vilket hade varit betydligt lättare.

# Självutvärdering

## 1. Vad har varit roligast i kunskapskontrollen?

Att få arbeta med något eget var väldig kul och definitivt jag skulle kunna fortsätta med.

## 2. Vilket betyg anser du att du ska ha och varför?

Jag anser jag uppnått VG. Då jag besvarat allting tydligt med en fungerande bot.

## 3. Vad har varit mest utmanande i arbetet och hur har du hanterat det?

Det som varit mest utmanande var att lösa min preprocess av min pdf.
Mestadels kapitel 4 då detta kapitel bröt pdf strukturen och behövdes behandlas på annat vis.
Försökte med olika metoder såsom RecursiveTextSplitter men detta gav inte resultaten jag ville ha. Fick det att fungera med regex.