Del 1: Teoretiska frågor

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

AI (artificiell intelligens) handlar om att utveckla system som kan utföra uppgifter som normalt kräver mänsklig intelligens, och det är ett övergripande område som omfattar flera olika tekniker och metoder. Maskininlärning är en del av AI där systemen lär sig från data. Deep Learning är en typ av maskininlärning som använder stora nätverk för att lära sig från mycket data. Så Deep Learning är en del av maskininlärning, och maskininlärning är en del av AI.

2. Hur är Tensorflow och Keras relaterat?

TensorFlow är ett öppet bibliotek för att bygga och träna maskininlärningsmodeller. Keras är ett verktyg som gör det lättare att skapa neurala nätverk och kan användas med TensorFlow. Keras hjälper till att skapa modeller snabbare, medan TensorFlow sköter beräkningarna och optimeringarna.

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

En parameter är något som modellen lär sig under träningen, som vikter i ett neuralt nätverk, och används för att göra förutsägelser. Hyperparametrar är inställningar vi väljer innan träning, som lärandets hastighet eller antal lager i nätverket. De styr hur modellen lär sig, men justeras inte under träningen.

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

Träningsdataset används för att lära modellen, det är den data modellen tränas på. Valideringsdataset används under träning för att justera hyperparametrar och förhindra överanpassning. Testdataset används för att utvärdera modellens prestanda efter träning, så vi kan se hur bra modellen fungerar på ny, 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])
```

Denna kod skapar ett neuralt nätverk där den första delen räknar antalet kolumner i träningsdata för att definiera inputformen. Modellen byggs upp med hjälp av en sekventiell struktur där vi lägger till olika lager ett efter ett. Det första lagret är ett fullt kopplat (dense) lager med 100 neuroner som använder ReLU som aktiveringsfunktion. För att förhindra överanpassning läggs även ett dropout-lager till, vilket tar bort 20% av neuronerna slumpmässigt under träning.

Sedan följer ett ytterligare dense-lager med 50 neuroner och ReLU som aktiveringsfunktion, och därefter ett utgångslager med en neuron som använder Sigmoid-aktivering. Detta används för att avgöra om något tillhör "den positiva klassen" eller inte.

När modellen har byggts, kompileras den med optimeraren "adam" och loss-funktionen binary_crossentropy. För att mäta prestanda används accuracy som en metrisk funktion. Early stopping definieras för att stoppa träningen om modellen inte förbättras på 5 epoker.

Vid träning används 20% av träningsdatan som en validering för att hålla koll på hur bra modellen presterar på nya data. Max 100 epoker väljs för att förhindra övertäningsproblematik, och early stopping används för att avbryta träningen om modellen inte förbättras efter ett visst antal epoker.

6. Vad är syftet med att regularisera en modell?

Syftet med regularisering är att förhindra överanpassning genom att göra modellen enklare och mer generaliserbar till nya data.

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

Dropout innebär att slumpmässigt "ta bort" vissa neuroner under träning för att förhindra att modellen blir för beroende av vissa funktioner och för att minska risken för överanpassning.

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

Early stopping innebär att träningen av modellen stoppas om den inte förbättras efter ett visst antal epoker, för att undvika överanpassning och spara tid.

9. Din kollega frågar dig vilken typ av neuralt nätverk som är populärt för bildanalys,

vad svarar du?

Jag skulle säga Convolutional Neural Networks (CNN) är de vanligaste, eftersom de är bra på att extrahera och känna igen mönster i bilder.

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

Ett CNN fungerar genom att använda konvolutionslager som letar efter specifika mönster eller funktioner i bilder, såsom kanter eller former, för att sedan använda dessa för att känna igen objekt i bilden.

11. Vad gör nedanstående kod?

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

Den här koden sparar modellen till en fil med namnet model_file.keras och sedan laddas den modellen igen med hjälp av load_model()-funktionen för att kunna användas senare utan att behöva tränas om.

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 (Central Processing Unit) är datorns "hjärna" som hanterar de flesta beräkningarna, men den är långsammare på att hantera stora mängder data samtidigt. GPU (Graphics Processing Unit) är specialiserad på att snabbt bearbeta stora mängder parallella beräkningar, vilket gör den mycket bättre för Deep Learning-träning.

Självutvärdering

1. Vad har varit roligast i kunskapskontrollen?

Det roligaste har varit att bygga chatbotten och verkligen se den fungera i praktiken. Jag har fått användning av både teoretiska och praktiska delar av kursen, som att förstå och implementera RAG-teknik och att använda Google Gemini för att skapa en interaktiv och användbar chatbot. Att se hur chatbotten svarar på frågor om examensarbetet har varit väldigt tillfredsställande och det känns som att jag verkligen lärde mig något nytt.

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

Jag tycker att jag borde få VG (Väl godkänt) för detta arbete. Jag har genomfört alla moment som krävs, inklusive att skapa en fungerande chatbot med RAG-teknik, implementerat semantisk sökning och använt Google Gemini API för att generera svar. Jag har också använt embeddings för att lagra och snabbt hämta relevant information från examensarbetet och jag har utvärderat chatbotten på ett strukturerat sätt. Jag har även testat och justerat chatbotten för att säkerställa att den kan hantera frågor på ett relevant och kontextuellt sätt.

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

Den största utmaningen har varit att justera semantisk sökning och embeddings så att chatbotten kan generera relevanta och korrekta svar. Det var svårt att få modellen att hantera komplexa frågor om examensarbetet på ett korrekt sätt utan att generera felaktiga svar. Jag hanterade detta genom att iterera och testa olika metoder för att förbättra svarens relevans, samt genom att skapa ett evalueringssystem för att objektivt mäta chatbotens prestanda.