

Teoretiska frågor

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

AI är det övergripande området som handlar om att få system att utföra uppgifter som normalt kräver mänsklig intelligens, till exempel att fatta beslut, lösa problem eller känna igen mönster.

Maskininlärning är en del av AI där system lär sig själva genom att analysera data och förbättra sin prestation utan att vara direkt programmerade för varje enskild uppgift.

Deep Learning är en underkategori av maskininlärning som bygger på neurala nätverk med många lager och används ofta för mer komplexa uppgifter, som bildigenkänning eller naturlig språkförståelse.

2. Hur är Tensorflow och Keras relaterat?

TensorFlow och Keras är båda verktyg för att bygga och träna neurala nätverk.

TensorFlow är ett kraftfullt ramverk utvecklat av Google för maskininlärning och deep learning.

Keras är ett användarvänligt gränssnitt ovanpå TensorFlow som gör det enklare att bygga modeller på ett mer lättläst och smidigt sätt.

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

En **parameter** är ett värde som modellen själv lär sig under träningen, till exempel vikter och bias i ett neuralt nätverk.

En **hyperparameter** är ett inställningsvärde som bestäms innan träningen, till exempel inlärningshastighet eller antal lager. Hyperparametrar styr hur modellen lär sig, men de justeras inte automatiskt av modellen utan av användaren.

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 och justera dess interna parametrar, så att den lär sig känna igen mönster och samband i datan.

Valideringsdata används under träningen för att utvärdera olika modellinställningar och hyperparametrar, samt för att upptäcka och undvika överträning.

Testdata används först när modellen är helt färdigtränad och vald. Testdatan ger en slutlig, objektiv utvärdering av modellens prestanda på ny data som modellen inte sett under vare sig träning eller validering.

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])
```

Koden bygger och tränar ett neuralt nätverk för binär klassificering i Keras/TensorFlow:

- Hämtar antalet indatafunktioner från träningsdatan.
- Skapar en sekventiell modell med tre dense-lager (100 och 50 noder med ReLU, samt ett utgångslager med sigmoid-aktivering) och ett dropout-lager för att minska risken för överanpassning.
- Kompilerar modellen med Adam-optimeraren, binary crossentropy som förlustfunktion och accuracy som utvärderingsmetrik.
- Använder "early stopping" som stoppar träningen om modellen inte förbättras på valideringsdata inom 5 epoker.
- Tränar modellen på träningsdata med 20 % som valideringsdata och maximalt 100 epoker (men kan avslutas tidigare tack vare early stopping).

6. Vad är syftet med att regularisera en modell?

Syftet med att regularisera en modell är att minska risken för överanpassning och därmed förbättra modellens förmåga att generalisera till ny data.

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

Dropout är en regulariseringsteknik där man slumpmässigt stänger av ett antal neuroner under träningen. Det förhindrar att nätverket blir beroende av enskilda noder och gör modellen mer robust.

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

Early stopping är en metod där träningen avslutas i förtid om valideringsfelet inte förbättras efter ett visst antal epoker. Det minskar risken för överträning.

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

Convolutional Neural Networks (CNN) är den vanligaste nätverkstypen för bildanalys, eftersom de är bra på att känna igen mönster och strukturer i bilder.

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

CNN analyserar bilder stegvis: konvolutionslager hittar lokala mönster, poolinglager minskar storleken, och fullt kopplade lager gör slutlig klassificering. De första lagren lär sig enkla mönster, senare lager kombinerar dessa till mer komplexa.

11. Vad gör nedanstående kod?

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

Koden sparar en tränad Keras-modell till en fil och laddar sedan in den igen för vidare användning, utan att behöva träna om modellen.

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 är datorns standardprocessor. GPU är specialiserad på parallella beräkningar och är därför mycket snabbare för deep learning.

Självutvärdering

1. Vad har varit roligast i kunskapskontrollen?

Det roligaste har varit att bygga en fungerande chattbot från grunden och se hela processen, från datainsamling till AI-svar. Jag har lärt mig mycket om RAG-metoden och hur man kombinerar olika verktyg, och det är kul att projektet har tydlig samhällsnytta.

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

Jag tycker att mitt arbete motsvarar VG, eftersom jag har gått igenom alla delar av uppgiften och lagt ner tid på att förstå och testa varje steg själv. Jag har reflekterat över både teknik, etik och användbarhet och försökt tänka på hur lösningen kan göra verklig nytta.

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

En av de största utmaningarna var att samla in och strukturera data eftersom Försäkringskassan inte har öppna API:er. Därför behövde jag samla in och organisera informationen manuellt, vilket både var tidskrävande och krävde noggrannhet. Av den anledningen valde jag att fokusera på tre förmåner. En annan svårighet var att förstå och använda chunking och embeddings. Jag hanterade detta genom att studera dokumentation, experimentera med olika inställningar och successivt testa lösningen för att hitta en fungerande metod.