

Del 1: Teoretiska frågor

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

AI är hela området som handlar om att skapa system som kan utföra uppgifter som normalt sett är ett mänskligt beteende. Maskininlärning är en underkategori till AI som handlar om att man lär AI mönster från data istället för att man programmerar exakt vad den ska göra. Inom maskininlärning finns också olika grenar varav Deep Learning är en del. Inom Deep Learning kan man hantera mer komplexa problem (som ex bildigenkänning och språkförståelse) och man använder sig av neurala nätverk med många lager för detta.

2. Hur är Tensorflow och Keras relaterat?

Keras är ett högnivå-API som gör det enklare att bygga neurala nätverk. Det fungerar som ett gränssnitt till underliggande ramverk som TensorFlow (från Google). När man använder Keras i TensorFlow, använder man TensorFlows motor för att träna och köra modeller men med ett enklare och mer intuitivt kodgränssnitt.

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

- Parameter är något modellen lär sig automatiskt från den data man tränar på, som till exempel vikter i ett neuralt nätverk.
- Hyperparameter är något vi bestämmer innan träningen och som styr träningsprocessen, som exempelvis antal lager eller aktiveringsfunktioner.

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

- Träningsdata är den data som används för att träna modellen
- Valideringsdata är en intern utvärdering som man kan använda när man tränar modellen för att utvärdera modellen och finjustera hyperparamterar för att undvika överanpassning.
- Testdata rör man inte/tittar på under träningen utan den används för att testa modellen när man tränat färdigt för att kunna mäta modellens generaliseringsförmåga.

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])

```

Det som sker är att man skapar och tränar ett neuralt nätverk med hjälp av Keras/Tensorflow för att lösa en klassificeringsuppgift. Det är två dense lager (100 och 50 neuroner) och använder dropout på 20 % för att minska överanpassning samt avslutas med en sigmoid-funktion som ger en sannolikhet som output, dvs den binära klassificeringen. Det tränas med Adam-optimerare och stoppas i förtid med early stopping om det inte förbättras inom 5 epoker. Modellen tränas på träningsdata där 20 % används som valideringsdata.

6. Vad är syftet med att regularisera en modell?

Syftet med att regularisera är att minska risken för överfitting.

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

DropOut är en teknik där man slumpmässigt stänger av ett visst antal neuroner i varje träningssteg. Detta för att modellen inte ska bli beroende av enskilda neuroner och ökar generaliseringsförmåga för modellen.

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

Early stopping är en teknik man kan använda för att avbryta träningen om valideringsförlusten inte förbättras efter ett visst antal epoker. Syftet är dels att spara tid och dels att förhindra överfitting.

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

CNN (Convolutional Neural Network)

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

Ett Convolutional Neural Network (CNN) fungerar som ett specialiserat neuralt nätverk designat för att hantera bilder. Det fungerar genom att systematiskt skanna av bilden och identifiera olika mönster, som exempelvis kanter, former eller texturer.

Detta görs i flera steg:

1. Mönsterigenkänning: CNN:et använder särskilda filter (konvolutionslager) som "glider" över bilden för att upptäcka och lära sig olika mönster. I de första stegen identifieras enklare mönster och i djupare lager kombineras dessa till mer komplexa drag.
2. Förenkling: Mellan mönsterigenkänningen sker en förenkling där nätverket minskar mängden data men behåller de viktigaste upptäckta mönstren. Detta hjälper till att modellen fokuserar på det väsentliga.
3. Klassificering: Slutligen, baserat på de identifierade mönstren och de viktigaste dragen, kan nätverket klassificera vad bilden föreställer (t.ex. en katt, en bil eller ett specifikt objekt).

11. Vad gör nedanstående kod?

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

Koden sparar en modell och laddar sedan in modellen i en variabel som heter my_model. Det gör att man kan använda en tränad modell om och om igen istället för att behöva träna om den varje gång.

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 och GPU är båda processorer i datorn men har olika användningsområden.

CPU (central processing unit) är den viktigaste komponenten som datorn inte fungerar utan, den hanterar många olika allmänna uppgifter i datorn. Man kan säga att det är den som berättar vad som ska göras i datorn.

GPU (graphic processing unit) är mer specialiserad processor som traditionellt använts till grafik men det har också visats sig vara bra till exempelvis träning av deep learning-modeller.

Vissa datorer har en integrerad GPU men exempelvis gamindatorer som ska leverera bra grafik i krävande spel behöver ett eget grafikkort. Eller om man vill träna deep-learning modeller, om man inte har det kan Google Colab vara just ett bra alternativ.