

Kunskapskontroll – Teoretiska Frågor

Besvara nedanstående frågor kort och koncist.

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

AI är ett omfattande område som strävar efter att skapa maskiner kapabla till mänsklig intelligens. Maskininlärning är en underkategori av AI som tillåter maskiner att lära sig från data utan explicit programmering. Deep Learning, en fördjupning av maskininlärning, använder djupa neurala nätverk för att analysera stora datamängder, vilket möjliggör mer komplexa insikter och förutsägelser.

2. Hur är Tensorflow och Keras relaterat?

TensorFlow är ett omfattande bibliotek för maskininlärning, utvecklat av Google. Keras fungerar som ett enklare och mer tillgängligt API som ligger ovanpå TensorFlow, och förenklar processen att bygga och träna neurala nätverk genom abstraktion av komplexa koncept

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

I ett neuralt nätverk refererar en parameter till både de koefficienter och bias som modellen lär sig under träningsprocessen, som båda justeras genom en iterativ spridningsprocess som är utformad för att minimera skillnaden mellan verkliga och inlärd data. Hyperparametrar är fördefinierade inställningar som tillämpas på en viss modell före träningen. I neurala nätverk påverkar detta både modellarkitekturen, till exempel att ange antalet dolda lager, samt träningsprocessen, inklusive element som batchstorlek. Hyperparametrar kan justeras efter den inledande träningen för att förfina modellens inlärningsprocess genom att justera inställningarna baserat på utvärderingar under validering eller korsvalidering.

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

Träningsdata: Används för att lära upp modellen.

Valideringsdata: Hjälper till att finjustera modellens hyperparametrar och verifiera att modellen inte överanpassar sig.

Testdata: Används för att ge en slutlig bedömning av modellens prestanda på osedd data, vilket ger en objektiv mätning av dess generaliseringsförmåga.

5. Förklara vad nedanstående kod gör:

```
1 n_cols = X_train.shape[1]
2
3 nn_model = Sequential()
4 nn_model.add(Dense(100, activation = 'relu', input_shape = (n_cols, )))
5 nn_model.add(Dropout(rate=0.2))
6 nn_model.add(Dense(50, activation = 'relu'))
7 nn_model.add(Dense(1, activation = 'sigmoid'))
8
9 nn_model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
10
11 early_stopping_monitor = EarlyStopping(patience = 5)
12 nn_model.fit(X_train, y_train, validation_split = 0.2, epochs = 100, callbacks = [early_stopping_monitor])
```

Koden anger en standardarkitektur för att bygga ett artificiellt neuralt nätverk eller ANN. Den avsätter först indatalagret från träningsdata som det första lagret som ska matas in i modellen. Sedan instansierar den en sekventiell modell där alla lager är helt anslutna och matas in i varandra en efter en. Det första lagret har 100 noder och en ReLU-aktiveringsfunktion för att lägga till icke-linjäritet i modellen

6. Vad är syftet med att regularisera en modell?

Regularisering används för att förhindra att modellen blir överanpassad till träningsdata genom att införa en kostnad (straff) för alltför komplexa modeller, vilket hjälper till att förbättra modellens generaliserbarhet.

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

Dropout innebär att under träningen inaktiveras slumpmässigt valda neuroner vilket förhindrar att nätverket blir för beroende av specifika neuroner. Detta uppmuntrar till robustare inlärning och minskar överanpassning.

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

Early stopping avbryter träningsprocessen när prestandan på valideringsdatan inte förbättras under en bestämd mängd epoker, vilket skyddar mot överträning och sparar tid och resurser.

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

Convolutional Neural Networks (CNNs) är särskilt effektiva för bildanalys då de är kapabla att upptäcka och klassificera visuella mönster direkt från pixlarna, genom användning av filter och pooling-lager.

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

CNN består av konvolutionslager som filtrerar inkommande data för att skapa feature maps, vilket fångar viktig information som kanter och texturer. Dessa egenskaper sammanförs i slutliga klassificerings- eller regressionslager för att göra förutsägelser.

11. Din vän har ett album med 100 olika bilder som innehåller t.ex. tennisbollar och zebror. Hur hade han/hon kunnat klassificera de bilderna trots att han/hon inte har någon mer data att träna en modell på?

En förtränad modell från ett dataset som ImageNet kan användas genom transfer learning för att klassificera bilder med specifika objekt (t.ex. tennisbollar och zebror) genom finjustering på den begränsade mängden tillgängliga bilder.

12. Vad gör nedanstående kod?

```
1 model.save('model_file.h5')
```

```
1 my_model = load_model('model_file.h5')
```

`model.save('model_file.h5')` sparar den tränade modellen i en fil, medan `load_model('model_file.h5')` återläser den sparade modellen. Detta möjliggör återanvändning av modellen utan behov av att träna om den.

13. Deep Learning modeller kan ta lång tid att träna, då kan GPU via t.ex. Google Colab skynda på träningen avsevärt. Läs följande artikel:

<https://blog.purestorage.com/purely-informational/cpu-vs-gpu-for-machine-learning/> och skriv mycket kortfattat vad CPU och GPU är.

CPU är hjärnan i datorn som utför generella beräkningsuppgifter medan GPU är en enhet optimerad för att hantera parallella beräkningar, vilket är idealiskt för att snabbt processa stora mängder data i deep learning.

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

- Att säkerställa att modellfilen fanns på rätt plats var en utmaning, men jag hanterade det genom att kontrollera sökvägen och ge felmeddelande vid behov.

2. Betyg och motivering:

- Jag förtjänar ett högt betyg på grund av min förmåga att lösa problem och slutföra projektet. Jag har visat uthållighet och praktisk kunskap inom ämnet.

3. Tips till mig själv:

- Öva regelbundet med kodexempel och projekt för att förbättra dina färdigheter.
- Var inte rädd för att be om hjälp och använd tillgängliga resurser när du behöver det