

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

AI är ett övergripande begrepp som syftar till att skapa datorprogram som kan utföra uppgifter som normalt kräver mänsklig intelligens. Maskininlärning är en central del inom AI där datorprogram har förmågan att lära sig från data. Deep Learning är en underkategori inom Maskininlärning som använder neurala nätverk med många lager, som inspireras av neuroner i våra hjärnor, för att utvinna komplexa mönster från data.

2. Hur är Tensorflow och Keras relaterat?

Tensorflow är en framework och Keras är en hög nivå bibliotek, vilket betyder att det är enklare att koda med Keras än att koda direkt med Tensorflow. Keras körs på Tensorflow – Tensorflow är som en motor som går att köras på egen hand, men Keras behöver en motor, till exempel Tensorflow, för att kunna användas.

3. Vad kännetecknar regressionsproblem?

I ett regressionsproblem är den predikterad variabeln numeriskt kontinuerlig som tar värden på ett kontinuerligt intervall.

4. Vad kännetecknar klassificeringsproblem?

I ett klassificeringsproblem är den predikterad variabeln diskret och tillhör en eller flera förutbestämda kategorier.

5. Vad är en parameter? Vad är en hyperparameter?

Inom Maskininlärning och Deep Learning är det viktigt att veta skillnaden mellan begreppen parameter och hyperparameter: en parameter är en variabel som modellen lär sig från, en hyperparameter är något som påverkar modellens beteende och prestanda. Man kan tänka så att en parameter är "vad" modellen ska lära sig om, och en hyperparameter är "hur" modellen ska lära sig.

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

Man kan dela in data i 3 olika delar: träningsdata, valideringsdata och testdata.

Man använder träningsdata för att skapa och träna olika modeller, med hjälp av valideringsdata utvärderar man och väljer man ut den bästa modellen som har skapats och tränats på träningsdata, när man har valt den bästa modellen så tränar man om den valda modellen på både träningsdata och valideringsdata. Till sist testar man den valda modellen på testdata för att se hur väl modellen generaliserar på datan som modellen inte har sett förut.

7. 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 ovan är en sekventiell neural nätverks modell för binär klassificering, nedan förklarar jag vad varje rad av kod gör:

Rad 1: Räknar antalet kolumner/funktioner i träningsdata 'X_train' och sparar dem i variabeln 'n_cols'.

Rad 3: Skapar en sekventiell modell och sparar modellen i variabeln 'nn_model'

Rad 4: Lägger till ett Dense lager med 100 neuroner, relu aktiveringsfunktion och har 'n_cols' antal funktioner.

Rad 5: Lägger till en regulariseringslager Dropout där 20% av neuroner slumpmässigt väljs ut för att motverka överanpassning.

Rad 6: Lägger till ett till Dense lager med 50 neuroner och relu aktiveringsfunktion.

Rad 7: Lägger till ett output lager med endast 1 output och sigmoid aktiveringsfunktion. Detta innebär att outputen är en binär klassificering där resultatet blir 0 eller 1.

Rad 9: Kompilerar modellen med adam som optimizer, binary_crossentropy som loss funktion samt accuracy som utvärderingsgrund för att se modellens prestanda.

Rad 11: Lägger till en early stopping för att avbryta träningen om valideringsförlusten inte förbättras på 5 epoker. Detta för att motverka överanpassning.

Rad 12: Tränar modellen med datan 'X_train' och svaren 'y_train'. Datan delas upp i en träningsset och en valideringsset där 20% är valideringsset. Träningen sker i 100 epoker (eller tills early stopping-villkoret uppfylls enligt rad 11). Early stopping används som callback under träningsprocessen.

8. Vad är syftet med att regularisera en modell?

Syftet med att regularisera en modell är att motverka modellen överanpassar sig till data.

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

Dropout är en regulariseringsteknik som "droppar" utvalda delar av neuroner under en träningsomgång. Detta leder till att neuronerna tvingas till att "lära sig själva" och inte "samarbeta". Om en modell överanpassar sig så får man höja "dropout rate".

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

Early stopping är en annan regulariseringsteknik som används för att motverka överanpassning. Denna teknik kollar på en modells valideringsfel för att tillämpa regularisering. När valideringsfelet slutar minska i ett visst antal epoker så kan vi stoppa träningen av modellen. Vi specificerar antalet epoker genom att ge parameter "patience" ett värde.

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

CNN, Convolutional Neural Network.

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

I en CNN modell identifieras först "low level features" såsom enklare former och färger. Sedan kombineras dessa enklare egenskaper för att skapa "high level features", såsom delar av ögon, mun och så vidare. Beroende på vilka egenskaper CNN modellen hittar, görs prediktioner om ifall ett objekt finns på bilden eller inte. CNN modeller använder "convolutional layers" i kombination med "pooling layers".

Ett "convolutional layer" består av flera filter där varje filter söker eller betonar vissa lokala attribut. I praktiken används flera filter för att hitta flera olika attribut och modellen lär sig själv vilka filter som ska användas, dvs vilka vikter som används. Ett "pooling layer" fokuserar på de viktigaste delarna i en bild, såsom ett maxvärde. Resultatet blir att bildens storlek minskar.

13. 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?

Han/hon hade kunnat använda en förtränad modell, ResNet-50, som är en CNN som har 50 lager, för att klassificera de bilderna.

14. Din kollega frågar dig vilken typ av neuralt nätverk som är lämpligt för att genomföra en sentimentanalys, vad svarar du?

RNN, Recurrent Neural Network, mer specifikt NLP (eller en seq-to-vector RNN variant).

15. Förklara översiktligt hur ett "Recurrent Neural Network" (RNN) fungerar.

För sekventiella data där ordningen spelar roll, som t ex text eller tidserier, kan man använda RNN. I en RNN modell beror outputen på inputen, men även föregående output. Pga att en output beror på föregående output som i sin tur beror på föregående output, så har RNN ett "minne". RNN möjliggör för neuronerna att behålla information om tidigare steg i sekvensen och använda den informationen när de bearbetar aktuella steg.

16. Vad gör nedanstående kod?

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

Första raden av kod sparar en modell som är färdigtränad med filnamnet 'model_file.h5'.

Andra raden av kod laddar en tidigare sparad modell med filnamnet 'model_file.h5' och sparar modellen i variabeln 'my_model' som en ny modellinstans, vilket gör det möjligt att använda den sparade tränade modell för att göra prediktioner samt att träna vidare utan att behöva bygga om modellen från grunden.