

DEEP LEARNING – DS24

Kunskapskontroll 2

Per Olevik

Svar teoretiska frågor

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

Deep Learning är en del av ML som är en del av AI

2. Hur är Tensorflow och Keras relaterat?

Tensorflow är motorn som sköter alla beräkningar medan Keras mer är ett användargränssnitt som hjälper användaren att styra så att Tensorflow gör rätt beräkningar.

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

En parameter är något som modellen beräknar och lär sig under träning, och som beskriver samband i datan. En hyperparameter styr hur modellens beräknas, och är alltså ett metodval som fördefinieras av användaren.

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

Modellens tränas på träningsdata och lär sig de samband som finns i den datan. Eftersom träningsdatan sannolikt innehåller vissa unika särdrag som inte är generella för populationen riskerar dock modellen att bli overfitted, dvs att de tar hänsyn till de unika egenskaperna vilket minskar generaliseringsförmågan. För att identifiera sådan problematik utvärderas modellen på valideringsdatan. Dessa steg kan göras med flera olika modeller och olika hyperparametrar, vilka kan utvärderas mot varandra baserat på prediktionsförmåga på valideringsdata. Den modell som presterar bäst tränas om på den sammanslagna tränings+valideringsdatan och utvärderas sedan på testdata, för att bedöma den slutgiltiga prediktionsförmågan.

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])

```

5.1 Först plockar vi ut antalet kolumner i x_train datasetet, dvs antalet förklaringsvariabler.

5.2 Vi definierar nn_model som en sekventiell modell, dvs att lagren läggs på ett efter ett.

5.3 Vi lägger till ett lager med 100 neuroner, RELU-aktivering och specificerar att input till lagret kommer att vara ett antal värden motsvarande antalet förklaringsvariabler, definierade i 5.1. Lagret är "dense", vilket innebär att alla neuroner i lagret tar input från alla neuroner i föregående lager, eller från all input i detta fall.

5.4 Adderar ett dropoutlager och specificerar att sannolikheten för att en neuron ska uteslutas är 0,2.

5.5 Adderar ytterligare ett dense-lager med 50 neuroner och RELU-aktivering.

5.6 Adderar ett sista dense-lager med en neuron och Sigmoid aktivering. Sigmoid ger ett värde mellan ett och noll, dvs en sannolikhet. Att det bara är en neuron tyder på att vi har en modell med binär output. Modellen kommer att spotta ur sig en sannolikhet för utfall "1".

5.7 Modellen sammanställs

5.8 Som optimeringsfunktion, dvs hur vikterna kommer att justeras, används "adam", som är ett snabbare alternativ än den traditionella "gradient descent".

5.9 Som loss-funktion används binary_crossentropy, vilken är lämplig vid binär klassificering.

5.10 Som utvärderingsmått av modellen används accuracy, dvs andel korrekta prediktioner av totala prediktioner.

5.11 Vi definierar en early_stopping med patience=5 vilket innebär att modellen kommer att stanna efter att loss har försämrats fem omgångar i rad.

5.12-14 Modellen tränas på x_{train} och y_{train} med

5.15 validation split 0,2, dvs den tränas på 80% av datan och utvärderas på 20% och

5.16 100 epoker, dvs de 80 procent av datan som används går över i sin helhet 100 gånger. MEN

5.17 early stopping aktiveras, vilket möjliggör att modellen stannar tidigare om accuracy försämras 5 epoker i rad.

6. Vad är syftet med att regularisera en modell?

Overfitting har förklarats under svaret på fråga 4. Syftet med regularisering är att hantera overfitting.

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

Vid Dropout fördefinieras en sannolikhet. Vid varje iteration av modellen slumpas, utifrån den sannolikheten om varje neuron ska inkluderas i körningen eller strykas. Inputen till neuronerna i hidden layers blir således mer oförutsägbart och respektive neuron tränas således till att bli mer självständig.

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

Early stopping innebär att man sätter en gräns för hur många epoker modellen ska köras utifrån loss function/valideringsfel. Så länge modellens loss minskar/förbättras fortsätter modellen att köras. Men när loss function har ökat n gånger (där n =patience definieras i förväg) bryts modellen den $n+1$:a gången.

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

Convolutional Neural Network är den typ av neurala nätverk som ofta förknippas med bildanalys.

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

För att en modell ska klassas som Convolutional Neural Network (CNN) krävs att minst ett hidden layer är ett Convolutional layer. Ett Convolutional layer innehåller flera filter vilka vart och ett fångar upp olika typer av egenskaper/mönster/attribut. Varje filter har sin egen vikt och modellen tränar själv och anpassar vikterna för att minimera loss.

Filtren jobbar på en delmängd av bilden i taget och söker igenom hela bilden.

Convolutional layers kombineras ofta med pooling-layers vilka plockar ut den viktigaste informationen ur varje delmängd av bilden och slänger bort resten, vilket således skapar en förminskad bild, och snabbar på modellen. De små lokala attribut som de olika filtren hittar kombineras i modellens senare lager ihop och skapar sammanhang som modellen kan använda för att förstå helheten.

11. Vad gör nedanstående kod?

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

Koden sparar en antagligen förtränad modell inklusive vikter, och laddar sedan tillbaka den i scriptet igen som *my_model*. Detta tillvägagångssätt innebär att modellen kan användas utan att 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 utför datorns beräkningar. Traditionellt utför en processor beräkningarna sekventiellt. Moderna CPU kan ha flera processorer vilket möjliggör begränsade parallella beräkningar.

GPU är likt CPU, men innehåller ett stort antal kärnor som parallellt tar sig ann olika delar av de beräkningar som ska utföras, och kan därmed arbeta betydligt snabbare.