Kunskapskontroll – teoretiska frågor

Svara kort och koncist på frågorna nedan.

1. Hur hänger AI, maskininlärning och djupinlärning ihop?

Djupinlärning är en delmängd av Machine Learning, som i sin tur är en delmängd av AI.

2. Hur är Tensorflow och Keras relaterade?

Keras är ett API som används för att träna neurala nätverksmodeller och använder Tensorflow i backend för att utföra beräkningar.

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 inlärda 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 du gör modellval och modellutvärdering kan du använda tränings-, validerings- och testdata. Förklara hur de olika delarna kan användas.

Träningsdata används för att bygga modellen där den lär sig mönster i en kontrollerad miljö. Valideringen använder en del av dessa träningsdata för att utvärdera modellen på osedda delar av träningsdata för att se om den behöver justeras ytterligare. Testdata är helt osynliga data som liknar verkliga data.

5. Förklara vad koden nedan 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 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.
- När lagren bygger på varandra i framåtpasset och matar från indata till utdata kombineras den här funktionen med summan av de viktade koefficienterna för att

hjälpa modellen att lära sig av träningsdata. När modellen går förbi indatalagret till det första av 2 ytterligare dolda lager använder den en form av regularisering som kallas dropout, som i det här fallet slumpmässigt släpper 20 % av noderna för att undvika överanpassning av tågdata. Sedan körs det andra dolda lagret igen genom 50 noder tills det slutligen matar ut en enda nod med hjälp av en sigmoidaktiveringsfunktion som används av en enda binär klassificeringsmodell.

- Modellen kompileras sedan genom att gå igenom backpropagation med hjälp av en Adam-optimerare, vilket är ett mer effektivt sätt att använda gradient descent för att minska förlusten genom att justera vikterna, med resultatet av denna process – under träning och testning – mätt med noggrannheten hos en binär korsentropijustering (lämplig för binära klassificeringsuppgifter).
- Slutligen tränas modellen baserat på den här arkitekturen, med 80 % för träning och 20 % för validering. Modellen går igenom iterationer av spridning framåt och bakåt av hela datauppsättningen 100 gånger (epoker) och är inställd på att sluta kontrollera valideringsfel när förlustnoggrannheten inte minskar efter 5 gånger.

6. Vad är syftet med att reglera en modell?

Syftet med att regularisera en modell är att minska modellens benägenhet för överanpassning av träningsdata.

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

Dropout inaktiverar slumpmässigt vissa neuroner under träningen för att undvika överanpassning genom att göra det möjligt för de andra neuronerna att lära sig mer effektivt i stället för att samanpassa sig till specifika funktioner i träningsdata, vilket förbättrar modellens förmåga att generalisera till nya data.

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

Tidigt stopp instruerar en träningsmodell att sluta lära sig när valideringsförlustfelet börjar öka efter ett visst antal batchar.

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

Ett konvolutionellt neuralt nätverk.

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

CNN är som ANN där de använder en hjärnliknande, skiktad struktur av neuroner för att lära sig komplexa mönster genom att kartlägga interaktionerna mellan visuella datapunkter. CNN är dock mer selektiva än ANN och använder en filterfunktion som liknar hur hjärnan gör för att dela upp en bild av till exempel en tiger som den försöker bearbeta i delar snarare än att titta på helheten. De införlivar det faktum att naturliga bilder har rumsliga korrelationer eller delade mönster. Varje filter är utformat för att identifiera dessa specifika mönster av funktionskartor, som markerar förekomsten av dessa funktioner på olika rumsliga platser. Modellen lär sig från dessa funktionskartor att vara selektiv eftersom varje lager av neuroner är anslutet, där varje karta eller lager sekventiellt plockar upp mer komplexa funktioner i den ursprungliga bilden. Den använder sedan ett koncept som kallas "pooling" för att minska storleken på varje partiell bild genom att välja de viktigaste aspekterna.

11. Din kompis har ett album med 100 olika bilder som innehåller till exempel tennisbollar och zebror. Hur kunde han/hon ha klassificerat dessa bilder trots att han/hon inte har mer data att träna en modell på?

Han/hon kunde manuellt ha tilldelat etiketter för att kategorisera de 100 bilderna och sedan till och med använda förtränade modeller och dataförstärkning för att lägga till fler "nya" bilder till varje objektklass.

12. Vad gör koden nedan?

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

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

Den första delen sparar din tränade modell som en h5-fil – behåller modellarkitekturen, vikterna och träningskonfigurationen – så att du sedan kan använda den andra delen i ett senare skede för att läsa in modellen för slutsatsdragning, finjustering eller ytterligare analys, utan att behöva köra den igen med hjälp av processorn.

13. Deep Learning-modeller kan ta lång tid att träna, då kan GPU:er via t.ex. Google Colab snabba upp träningen avsevärt. Läs följande artikel: https://blog.purestorage.com/purely-informational/cpu-vs-gpu-for-machine-learning/ och skriv mycket kort vad CPU och GPU är.

CPU är den centrala processorenheten i en dator och är i huvudsak datorns central- eller huvudmotor som använder RAM för att fungera, medan GPU är den grafiska processorenheten som är ett kraftfullare kort som använder separat minneslagring och därmed är effektivare att använda för vissa uppgifter.