**Teoretiska Frågor**

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

**A diagram of machine learning

Description automatically generatedArtificiell intelligens** är ett samlingsbegrepp för teknologier som kan uppfatta sin omgivning, resonera, lära sig och vidta åtgärder.

**Maskininlärning** är en dale av artificiell intelligens. Det fokuserar på att utbilda datorer att lära sig från data och förbättra sina, snarare än att vara specifikt programmerade för varje uppgift.

**Djupinlärning** är en del av maskininlärning som använder flera layers neurala nätverk, så kallade djupa neurala nätverk för att efterlikna den komplexa beslutsförmågan hos den mänskliga hjärnan, flesta av artificiell intelligens (AI) använder innom av någon typ av djupinlärning.

**Vi kan säga att deep learning är en teknik av maskininlärning, som är en del av det bredare fältet av artificiell intelligens.**

---------- ---------- ---------- -----------

1. Hur är Tensorflow och Keras relaterat?

**A screenshot of a computer

Description automatically generated**

**TensorFlow** är ett open source-bibliotek från Google för maskininlärning och djupinlärning, det har verktyg, bibliotek och resurser för att bygga och träna olika typer av neurala nätverk och andra maskininlärningsmodeller.

**Keras** är ett också open source-bibliotek för att bygga och träna neurala nätverk, designat för att vara enkelt och flexibelt. Det används ofta som ett högnivågränssnitt TensorFlow, utan att behöva detaljkunskaper om den underliggande arkitekturen.

**Alla Keras-funktioner och modeller är direkt tillgängliga i TensorFlow utan behov av extra installationer eller konfigurationer. Keras gör det enklare att utnyttja TensorFlows kraftfulla funktioner genom att erbjuda ett mer användarvänligt gränssnitt. TensorFlow från version 2.0 är Keras helt integrerat i TensorFlow-biblioteket.**

---------- ---------- ---------- -----------

3-Vad är en parameter? Vad är en hyperparameter?

A diagram of a math problem

Description automatically generated with medium confidence**En parameter** är en variabel inom modellen som justeras utifrån träningsdata. Den är en integrerad del av modellen och optimeras automatiskt under träningen för att minimera förlustfunktionen.

**En hyperparameter** är en förinställd inställning eller konfiguration som fastställs innan modellträningen påbörjas och styr sökprocessen för att hitta bästa parametrar.

Hyperparametrar påverkar modellens inlärningsprocess men justeras inte automatiskt under träning.

---------- ---------- ---------- -----------

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

A group of dogs and a sign

Description automatically generated**Träningsdata** används för att anpassa modellen genom att justera dess parametrar och minimera förlustfunktionen.

**Valideringsdata** används för att utvärdera modellens prestanda och justera hyperparametrar under träningen.

**Testdata** används för att bedöma modellens slutgiltiga prestanda på osett data och säkerställa att den kan generalisera korrekt.

Genom att använda tränings, validerings- och testdata på detta sätt kan man utveckla och utvärdera modeller.

---------- ---------- ---------- -----------

1. Den här koden skapar och tränar en neural nätverksmodell för binär klassificering med hjälp av Keras, en högnivå-API för neurala nätverk som är integrerad med TensorFlow. Här är en förklaring av vad varje del av koden gör:

En bild som visar text, Teckensnitt, skärmbild

Automatiskt genererad beskrivning

**n\_cols = X\_train.shape :**

Beräknar antalet kolumner i träningsdatan X\_train.

**nn model = Sequential():**

Skapar ett neuralt nätverk, vilket är den vanligaste typen av neurala nätverk i Keras.

**nn\_model.add(Dense(100, activation='relu', input\_shape=(n\_cols,))):**

Lägger 100 neuroner och aktiveringsfunktionen ReLU (Rectified Linear Activation) till nätverket. Det är det första lagret och kräver med att specificeras input\_shape.

**nn\_model.add(Dropout(rate=0.2)):**

Ett dropout med en dropout hastighet på 20%.

Det för att förhindra överanpassning genom att slumpmässigt inaktivera en del av neuroner under träningen.

**nn\_model.add(Dense(50, activation='relu')):**

Ett dolt lager med 50 neuroner och ReLU aktiveringsfunktionen.

**nn\_model.add(Dense(1, activation='sigmoid')):**

Ett utmatningslager med ett neuron och en sigmoid aktiveringsfunktion. Sigmoid används för binär klassificering för att producera en sannolikhetsutgång mellan 0 och 1.

**nn\_model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy']):**

Kompilerar modellen med en Adam-optimizer, binär korsentropi som förlustfunktion och noggrannhet som mätmetrik.

**early\_stopping\_monitor = EarlyStopping(patience=5):**

Skapar EarlyStopping som kommer att stoppa träningen om valideringsförlusten inte förbättras under 5 på varandra följande epoker.

**nn\_model.fit(X\_train, y\_train, validation\_split=0.2, epochs=100, callbacks=[early\_stopping\_monitor]):**

Tränar modellen på träningsdata X\_train och y\_train med en valideringsuppdelning på 20%, upp till högst 100 epoker, med tidig stoppövervakning.

---------- ---------- ---------- -----------

1. Vad är syftet med att regularisera en modell?

Regularization is a set of methods for reducing overfitting in machine learning models. Typically, regularization trades a marginal decrease in training accuracy for an increase in generalizability.

---------- ---------- ---------- -----------

1. ”Dropout” är en regulariseringsteknik, vad är det för något?

A diagram of a model of a deep learning

Description automatically generated

**Dropout** är en regulariseringsmetod som efterliknar samtidig träning av många olika neurala nätverk. Under träningen ignoreras slumpmässigt vissa neuroner i lagret, vilket får lagret att framstå som om det har ett varierande antal noder och anslutningar till föregående lager.

---------- ---------- ---------- -----------

1. ”Early stopping” är en regulariseringsteknik, vad är det för något?

A diagram of a graph

Description automatically generated with medium confidence

Inom maskininlärning är tidig stopp en form av regularisering som används för att undvika överanpassning när man tränar en inlärare med en iterativ metod, såsom gradientnedstigning. Sådana metoder uppdaterar eleven så att den bättre passar träningsdatan med varje iteration.

---------- ---------- ---------- -----------

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

Neural Networks (CNNs) är den typ av neuralt nätverk som är populär för bildanalys.

CNN är särskilt effektiva för att extrahera och lära sig hierarkiska funktioner från bilder, Det gör dem mycket lämpade för uppgifter som bildklassificering, objektdetektion och bildsegmentering.

---------- ---------- ---------- -----------

1. Förklara översiktligt hur ett ”Convolutional Neural Network” fungerar.

A diagram of a machine learning

Description automatically generated

Ett konvolutionellt neuralt nätverk (CNN) är en typ av artificiellt neuralt nätverk som främst används för bildigenkänning och bearbetning, på grund av dess förmåga att känna igen mönster i bilder. En CNN är ett kraftfullt verktyg men kräver miljontals märkta datapunkter för träning.

---------- ---------- ---------- -----------

1. 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?

Om din vän inte har tillgång till mer data för att träna en modell kan man använda en teknik som kallas transfer learning.

Med transfer learning använder man en förtränad modell som redan har lärt sig att extrahera generella mönster från en stor mängd data (till exempel tusentals bilder) och överför den kunskapen till en liknande uppgift, även om den inte har mycket träningsdata.

Din vän kan använda en förtränad modell, till exempel en CNN som tränats på en stor dataset som ImageNet, för att extrahera viktiga funktioner från de 100 bilderna med tennisbollar och zebror.

Den funktioner matas in i en ny klassificeringsmodell, som till exempel ett enkelt klassificerande neuralt nätverk, som tränas enbart på de 100 bilderna.

Genomdetta kan man utnyttja den generella kunskapen som redan finns i den förtränade modellen för att göra mer precisa prediktioner, även med begränsad mängd träningsdata.

---------- ---------- ---------- -----------

1. Vad gör nedanstående kod?

En bild som visar text, Teckensnitt, skärmbild, linje

Automatiskt genererad beskrivning

Första raden sparar den tränade modellen till en fil med namnet 'model\_file.h5'.

Andra raden laddar en sparad modellen från filen 'model\_file.h5'

efter det man kan användas för att göra prediktioner eller fortsätta träningen.

---------- ---------- ---------- -----------

1. 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.

A diagram of a computer hardware

Description automatically generated

**CPU** (Central Processing Unit) hanterar allmänna uppgifter snabbt på ett sekventiellt sätt. Den är kraftfull för olika typer av applikationer men mindre effektiv för parallella beräkningar.

Vi ka säga att CPU: Idealisk för sekventiella uppgifter, multitasking och allmänna beräkningar.

**GPU** (Graphics Processing Unit) är utmärkt på parallell bearbetning och delar upp komplexa uppgifter i mindre, samtidiga beräkningar. Ursprungligen designade för att rendera grafik, accelererar GPU nu maskininlärning genom att hantera distribuerade beräkningsprocesser mer effektivt.

Vi kan säga att GPU: Minskar avsevärt träningstiden för neurala nätverk och djupinlärning modeller genom att bearbeta flera operationer samtidigt, vilket gör dem överlägsna CPU för dessa applikationer.

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

Jag mötte utmaningar med dataobalans och överanpassning. Jag kunde hantera dataobalansen genom att utöka data med ImageDataGenerator.

Jag minimerar överanpassning genom att använda dropout-lager och EarlyStopping.

* Vilket betyg du anser att du skall ha och varför.

Jag tror att jag förtjänar högt betyg för att jag has löste komplexa problem, samt optimerade modellen för realtid.

* Tips du hade ”gett till dig själv” i början av kursen nu när du slutfört den.

Lär mer om regularisering tekniker och hyperparameter tuning.

Fokusera på att förbättra datakvaliteten.

Dokumentera alla steg i processen för att underlätta framtida arbete.