

1. Hur är AI, Maskininlärning (ML) och Deep Learning (DL) relaterat?

Svar: **Maskininlärning** och **Deep Learning** är metoder för **artificiell intelligens** som tillämpar algoritmer på data. ML är en delmängd av AI som använder algoritmer för att lära sig mönster från data. DL är en delmängd av ML som använder artificiella neurala nätverk för komplexa uppgifter.

2. Hur är Tensorflow och Keras relaterat?

Svar: **TensorFlow** är en end-to-end-plattform med öppen källkod, ett bibliotek för flera maskininlärningsuppgifter, medan **Keras** är ett neuralt nätverksbibliotek på hög nivå som körs ovanpå TensorFlow.

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

Svar: **Parametrar** är koefficienterna i en modell som algoritmen justerar under träning för att minimera felet mellan den förutsägda output och den faktiska output. **Hyperparametrar** är parametrar vars värden styr inlärningsprocessen och bestämmer värdena på modellparametrar som en inlärningsalgoritm lär sig.

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

Svar: **Träningsdata** är den datan som används för att träna och få modellen att lära sig de dolda funktionerna/mönstren i datan. Under varje epok matas samma träningsdata till den neurala nätverksarkitekturen upprepade gånger, och modellen fortsätter att lära sig funktionerna i datan. **Valideringsdata** är datan, separat från träningsdata, som används för att validera vår modellprestanda under träning. Denna valideringsprocess ger information som hjälper oss att justera modellens hyperparametrar och konfigurationer därefter. **Testdata** är en data som används för att testa modellen efter avslutad Träning. Det ger en opartisk slutlig modellprestandamått när det gäller accuracy, precision, etc.

5. Förklara vad nedanstående kod gör:

Svar: Koden börjar med att definiera **'input shape'**. Efter det konstruerar den en **'Sequential'** neural nätverksmodell. Lager läggs sedan till det neurala nätverket: det initiala lagret består av 100 **'Dense'** noder med en **'relu'**-aktiveringsfunktion och den specificerade input shape. Därefter tillämpas en **'Dropout'** Rate på 20 % för att slumpmässigt inaktivera noder för förbättrad träning. Efter dropout-lagret läggs ytterligare ett lager som består av 50 noder, som kulminerar i ett outputlager med en enda nod och **'sigmoidaktivering'**.

Efter dessa arkitektoniska steg **kompileras** modellen med **'Adam-optimizer'** för att underlätta träningsoptimering. Valet av **'binary_crossentropy'** som **'loss'** funktion indikerar att modellen är inriktad på **klassificeringsuppgifter**. Slutligen anges **'accuracy'** som utvärderingsmåttet för att mäta modellens prediktiva **'accuracy'**.

Efter kompileringen implementeras en **'early stopping monitor'** för att stoppa träningen när **'patience'** når 5, och upphöra den om **'loss'** inte minskar under de senaste 5 epokerna. Slutligen

är modellen anpassad (**fit**) till '**x_train**' och '**y_train**', med en 20% **valideringsdelning** och 100 **epoker** specificerade för träning.

6. Vad är syftet med att regularisera en modell?

Svar: **Regularisering** är en metod för att minska overfitting i maskininlärningsmodeller. Normalt byter regularisering ut en marginell minskning av tränings accuracy mot en ökning av generaliserbarheten. Regularisering omfattar en rad tekniker för att korrigera för överanpassning i maskininlärningsmodeller. Till exempel: Ridge Regression, Lasso Regression, Dropout, osv.

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

Svar: Inom maskininläring hänvisar "**Dropout**" till praxis att bortse från vissa noder i ett lager på slump under träning. Ett 'Dropout' är en regulariseringsteknik som förhindrar overfitting genom att säkerställa att inga enheter är samberoende med varandra. 'Dropout' förhindrar sammanpassning av neuroner och uppmuntrar robusthet genom att simulera en ensemble av mindre nätverk under träning.

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

Svar: '**Early Stopping**' är en regulariseringsteknik för djupa neurala nätverk som stoppar träningen när parameteruppdateringar inte längre ger bättre resultat på valideringsdatan. Istället för att träna modellen för ett fast antal epoker övervakar 'Early Stopping' modellens prestanda på en valideringsset under träning och stoppar träningsprocessen när prestandan på valideringssetet börjar försämrats. Detta förhindrar att modellen fortsätter att lära sig bruset i träningsdata och hjälper den att generalisera bättre till osynliga data.

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

Svar: Jag skulle säga att för bildanalys **Convolutional Neural Network (CNN)** är den mest populära typen av neurala nätverk. Den är speciellt utformad för att fungera bra med visuell data. CNN:er är strukturerade för att automatiskt och adaptivt lära sig rumsliga hierarkier av funktioner från bilder, vilket gör dem väl lämpade för uppgifter som objektigenkänning och bildklassificering. CNNs arkitektur inkluderar lager som **convolutional layer**, **pooling layer** och **fully connected layer**, vilket gör att de effektivt kan fånga och bearbeta visuella funktioner.

Andra algoritmer kanske inte är lika effektiva för bildklassificering eftersom de kanske inte har samma inbyggda förmåga att lära sig hierarkiska representationer av visuella data. Populära CNN-arkitekturer inkluderar LeNet, AlexNet, VGG, GoogLeNet (Inception), ResNet och på senare tid EfficientNet.

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

Svar: En **CNN** består vanligtvis av flera lager, som brett kan kategoriseras i tre grupper: convolutional layers, pooling layers och fully connected Layers. **Convolutional layer** är den grundläggande byggstenen i ett CNN och det är där majoriteten av beräkningarna sker. Det här

lagret använder ett filter eller en kernel- en liten matris av vikter - för att flytta över det mottagliga fältet för en ingångsbild för att upptäcka närvaron av specifika egenskaper. **Pooling layer** syftar till att minska dimensionaliteten hos input data samtidigt som kritisk information behåll, vilket förbättrar nätverkets totala effektivitet. **Fully Connected layer** spelar en avgörande roll i slutskedet av ett CNN, där det ansvarar för att klassificera bilder baserat på funktionerna som extraherats i de tidigare lagren. Termen 'fully connected' betyder att varje neuron i ett lager är ansluten till varje neuron i det efterföljande lagret.

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

Svar: Jag skulle föreslå min vän att antingen använda **Data Augmentation**, där hon på konstgjord väg kan öka storleken på datasetet. Detta inkluderar operationer som att rotera, vända, skala och beskära bilderna. Ett annat förslag jag skulle ge henne är att hon skulle kunna göra **transferinlärning** och använda en förtränad modell genom att finjustera den efter sina egna behov. Hon kunde använda en pre-trained modell, som VGG eller ResNet. Dessa modeller är tränade på stora datamängder som ImageNet och kan känna igen en mängd olika objekt. Så även om hon har en liten data set kan hon använda dessa båda tekniker tillsammans för att bygga en bra modell.

12. Vad gör nedanstående kod?

Svar: Den första raden med kod är att spara den tränade modellen i en **'h5'-fil** med metoden **'save'** med namnet 'model_file.h5'. Den andra raden med kod laddar den sparade modellen med **'load_model'**-funktionen, genom att ge den en variabel 'my_model', för att använda den för ytterligare ändamål.

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.

Svar: **CPU (Central Processing Unit)** är en processor eller dators hjärna som bearbetar grundläggande instruktioner som aritmetiska eller logiska funktioner. En CPU bearbetar uppgifter sekventiellt, där uppgifterna delas upp mellan dess flera kärnor för att uppnå multitasking.

GPU (Graphics Processing Unit) är också en dators processor, som ursprungligen är designad för att rendera 2D- och 3D-bilder, videor och animationer. GPU:er används i applikationer långt bortom grafikbearbetning, inklusive stora analyser och maskininlärning. GPU:er använder parallell bearbetning, och delar upp uppgifter i mindre deluppgifter som är fördelade på ett stort antal processorkärnor i GPU:n. Detta resulterar i snabbare bearbetning av specialiserade datoruppgifter.