

Kunskapskontroll – Teoretiska Frågor

Besvara nedanstående frågor kort och koncist.

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

AI inkluderar maskininlärning och deep learning, och maskininlärning inkluderar deep learning (IBM, n.d). AI är den bredaste termen, och står för "maskiner" som härmar mänsklig intelligens och kognitiva funktioner. Deep learning kan vara mer automatiserat än maskininlärning, och efterliknar mer vår egen hjärnas sätt att lära sig med neuroner.

2. Hur är Tensorflow och Keras relaterat?

Keras är som ratten på en bil, om tensorflow är motorn (Lektion 1). Med hjälp av Keras så får man ett smidigt sätt att kontrollera Tensorflow.

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

En parameter är en inställning med ett värde. En hyperparameter är också ett värde, men ett värde vi har möjlighet att ändra för att påverka hur väl modellen tränas till vår data. Man kan t.ex. ändra antalet neuroner i de dolda lagren, för att se hur det påverkar modellen. Hur många dolda lger, drop-out-rate o.s.v. Hyperparametrar är parametrar vars värden kontrollerar inlärningsprocessen (Towards Datascience, n.d).

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.

Man delar upp sin data, så att all data blir representerad i tre delar. Träningsdata används för att träna modellen. Om man har flera modeller och/eller vill justera hyper-parametrar, så kan det vara bra att använda ett validerings-set för att utvärdera vilken modell som fungerar bäst. När man med validerings-setet valt ut den bästa modellen, så ställer man den mot test-setet så att den får ett slutgiltigt resultat. Test-setet används för att modellen ska ha något som den inte sett innan.

5. 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])
```

Först sparar man ner värdet av hur många kolumner tränings-setet har.

Sequential modell används, den tillåter en linjär sekvens av lager.

Base layer kräver input shape beroende på antalet kolumner i tränings-datan. Dense layer har sin egen weight matrix enligt Géron (2019, s.299)., denna med 100 noder. Relu är en activation function som ger output 0 om input-värdet är mindre än 0, vilket motverkar vanishing gradient, och gör att modellen tränas snabbare och bättre (Machine Learning Mastery, n.d.).

Drop out-lagret är en regulariseringsteknik som hjälper till att förhindra overfitting genom att stänga av neuroner under träningen, 20% i detta fallet.

Ännu ett dense-layer, med 50 noder.

Ett output layer med 1 neuron, vilket är "binary", t.ex. ja/nej, och då är sigmoid activation function bra att använda, så att slutresultatet är mellan 0 och 1.

Modellen kompileras. Eftersom det är binary classification problem så används loss function binary_crossentropy, för att mäta optimeringen. "Adam" (Adaptive Moment Estimation) är den valda optimerings-algoritmen för att optimera vikterna o.s.v. för att förbättra loss (Geeks for geeks, n.d.). Metrics accuracy är ett annat verktyg för att mäta hur väl modellen fungerar, hur ofta predictions = labels.

Early stopping är för att stoppa träningen om validation loss stiger antalet {patience} epoker i rad.

Tillsist tränas modellen med tränings-seten, en validation-split på 20% av tränings-seten, under 100 epoker (antalet gånger den tränas om igen), och early stopping inkluderas.

6. Vad är syftet med att regularisera en modell?

Det kan vara bra att börja med för mycket än för lite dolda lager och neuroner när man ska träna en modell, men det kan leda till "overfit", och det kan man regularisera med regulariseringstekniker.

Dropout är en av flera. Dropout med en egen satt sannolikhet, tvingar neuroner att lära sig själva och inte "samarbeta" lika mycket med andra neuroner (Lektion 1). L1, Lasso, är bra mot outliers (Medium, n.d.). L2, Ridge, fungerar bra till komplex data med är dålig till outliers.

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

Jag råkade svara på det i fråga 6 ovan.

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

När vi tränar en modell under ett visst antal epoker så blir modellen förhoppningsvis bättre efter var epok. Varje epok kan ge oss ett valideringsfel som visar hur träningen går, hur felet minskar efter varje epok. Valideringsfelet kan dock börja öka igen, och med "patience" kan vi bestämma när vi ska stoppa träningen, om valideringsfel fortsätter att öka förbi vår satta patience-nivå (Lektion 1).

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

CNN, Convolutional Neural Network. RNN är bättre för t.ex. tids-serier, vilket ord som kommer sist i en mening i text och tal, m.m.

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

CNN är en klass av ANN, och är inspirerad av hur hjärnan bearbetar visuell information (Lektion 4). Enkla former och färger identifieras först, och sedan kombineras dom till mer avancerade former, som öron, näsa o.s.v. Convolutional layers är filter som man kan ställa in på olika sätt för att läggas och flyttas (stride) över bilden, där den med hjälp av sina matris-värden kan "betona" värdena i bilden och på det sättet skapa fokus på olika former (Lektion 4). Modellen lär sig själv vilka filter som skall användas. "Convolution" är namnet på operationen som inkluderar filtrets och delmatrisens matematiska matris-uträkning.

11. 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?

Med hjälp av data augmentation (Tensorflow, n.d.), kan man generera "ny" träningsdata genom att ta den träningsdata man har, och förändra den på olika sätt (t.ex. rotering, spegling). ImageDataGenerator kan användas till detta. 100 bilder låter ganska lite att börja med dock.

12. Vad gör nedanstående kod?

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

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

Koden sparar / laddar modellen till / från hårddisken, vilket gör att man kan använda modellen i andra sammanhang än i dokumentet den skapades i.

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.

GPUs är bättre på parallella uträkningar, att bryta ner komplexa problem till mindre uträkningar. CPU gör detta sekventiellt. CPU, central processing unit, är datorns "hjärna", och bra på att hantera datorns program. GPU, graphics processing unit, var egentligen menad för bilder/grafik, men upptäcktes fungera väldigt bra till machine learning tack vare parallel processing. Detta passar bättre till de stora data set som matas in till machine learning-processen.

Geeks for geeks. Adam optimizers in tensorflow. Hämtad 15 maj, 2024, från Geeks for geeks sida <https://www.geeksforgeeks.org/adam-optimizer-in-tensorflow/>

Géron, Aurélien. (2019). Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Second edition. O'Reilly Media.

IBM. AI vs deep learning vs neural networks. Hämtad 15 maj, 2024, från IBM's sida <https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>

Machine learning mastery. Rectified linear activation function for deep learning neural networks. Hämtad 15 maj, 2024, från Machine learning mastery's sida <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>

Medium. Understanding L1 and L2 regularization for deep learning. Hämtad 15 maj, 2024, från Mediums sida <https://medium.com/analytics-vidhya/regularization-understanding-l1-and-l2-regularization-for-deep-learning-a7b9e4a409bf>

Tensorflow. Data augmentation. Hämtad 15 maj, 2024, från Tensorflow's sida https://www.tensorflow.org/tutorials/images/data_augmentation

Towards datascience. Parameters and hyperparameters. Hämtad 15 maj, 2024, från Towards datascience's sida <https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac>