**KUNSKAPSKONTROLL 2**

(DEEP LEARNING)

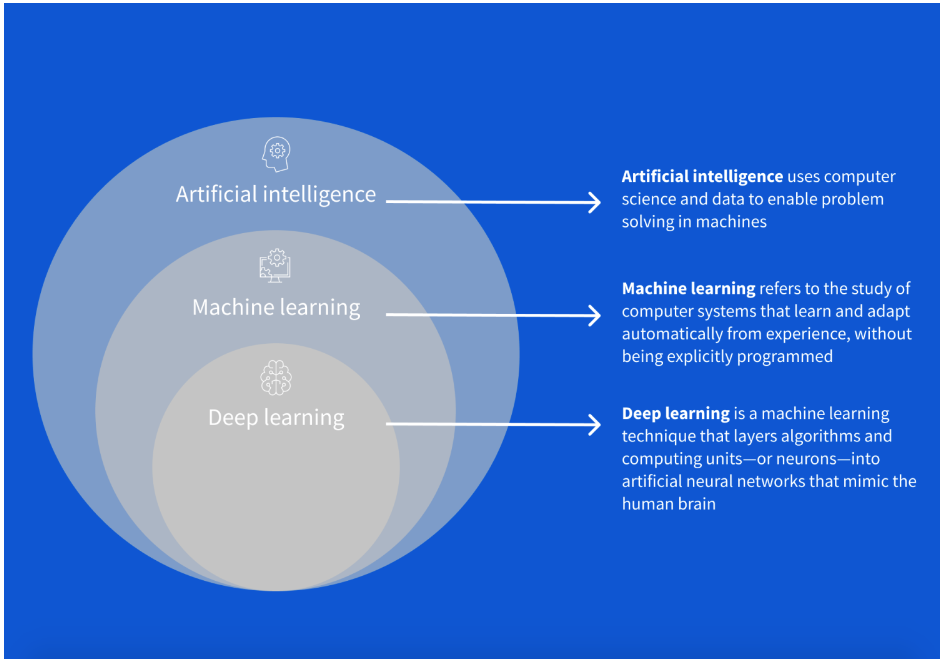
**TEORETISKA FRÅGOR**

EC UTBILDNING

AIKATERINI ANTONIOU

05-2024

**1**.Artificial Intelligence (AI) is the broad field that involves creating systems capable of performing tasks that typically require human intelligence, such as problem-solving, understanding natural language and recognizing patterns. Machine Learning (ML) is a subset of AI that focuses on developing algorithms that allow computers to learn from and make decisions based on data. Within ML, Deep Learning (DL) is a specialized area that uses multi-layered neural networks to analyze various factors of data, allowing for more complex and nuanced pattern recognition. DL has led to significant advances in fields like image and speech recognition due to its ability to handle large amounts of structured and unstructured data. So it can be concluded that AI is the broadest concept, encompassing any technique that enables machines to mimic human intelligence, while ML is a subset of AI, focused on algorithms that allow computers to learn from data and DL is a subset of ML, using neural networks with many layers to model complex patterns in data. The figure below better illustrates this relationship between them. (Coursera Staff,2024;GeeksforGeeks,2023)



**2**.TensorFlow and Keras are both essential tools in the deep learning ecosystem and they are closely related. TensorFlow is an open-source deep learning framework developed by Google Brain, designed for high-performance numerical computation and the creation of machine learning models. Keras, originally an independent project, is an open-source neural network library written in Python that offers a user-friendly, high-level interface for building and training deep learning models. Initially, Keras could work with multiple backend engines, including TensorFlow, Theano, and CNTK. However, since 2017, Keras has become tightly integrated with TensorFlow and is now the official high-level API of TensorFlow. This integration simplifies the process of building complex models by providing an intuitive and easy-to-use interface while still leveraging TensorFlow's powerful computational capabilities for performance and scalability. This combination allows developers to quickly prototype and deploy deep learning models with minimal code, making it accessible to both beginners and experts in the field.(Terra,2024)

**3**.A parameter is a variable that is learned from the data during the training process of a model. Examples of parameters are the weights and biases in a neural network. These values are adjusted through the training process as the model learns from the training data, aiming to minimize the error or loss function.

A hyperparameter on the other hand is a configuration that is set before the learning process begins and cannot be learned from the data. Hyperparameters control the learning process and include settings like the learning rate, the number of hidden layers in a neural network, the number of units in each layer, the batch size and the number of training epochs.(Pramod,2023)

**4**. When performing model selection and evaluation the dataset is typically divided into three parts: training data, validation data, and test data. Each part serves a distinct purpose:

Training Data: This is the portion of the dataset used to train the model. The model learns from this data by adjusting its parameters to minimize the error or loss function. The training data should be representative of the overall dataset to ensure that the model can learn relevant patterns and relationships.

Validation Data: After the model is trained on the training data, it is evaluated on the validation data. This dataset is used to tune hyperparameters and make decisions about model architecture. The validation data helps in assessing how well the model generalizes to unseen data, allowing for adjustments to prevent overfitting.

Test Data: Once the model has been trained and tuned using the training and validation data, it is evaluated on the test data. This dataset is kept separate from the training and validation processes to provide an unbiased assessment of the model's performance. The test data simulates how the model will perform on real-world, unseen data and helps estimate the generalization error.

So generally can be said that training data is for learning, validation data is for tuning and test data is for final evaluation. This division helps in building a model that generalizes well to new data and avoids overfitting. (Saha,2023;Shah,2017)

**5**. The code builds and trains a neural network for binary classification, using early stopping to prevent overfitting and improve generalization. The analysis of the code is as follows:

At first a Sequential model is created which means layers will be added one after another. So the first layer is a Dense layer with 100 neurons and ReLU activation function. The input\_shape parameter indicates the input data shape, which is (n\_cols,). A Dropout layer is added to prevent overfitting by randomly setting 20% of the input units to 0 at each update during training. The second Dense layer has 50 neurons and also uses the ReLU activation function. The final Dense layer has 1 neuron with a sigmoid activation function, suitable for binary classification.

Then the model is compiled with the Adam optimizer. The loss function is set to binary cross-entropy, appropriate for binary classification tasks and the model will also track accuracy as a metric during training. After that an early stopping callback is created to monitor the validation loss during training. The code writes that if the validation loss does not improve for 5 consecutive epochs, training will be stopped early to prevent overfitting.

Finally the model is trained on the training data X\_train and y\_train. According the code 20% of the training data is used as validation data to evaluate the model’s performance during training and the model is set to train for up to 100 epochs but early stopping may halt training sooner if validation loss stops improving. According the code the early stopping monitor is used as a callback during training.

**6**. The purpose of regularizing a model is to prevent overfitting. Overfitting occurs when a model learns the training data too well, including its noise and outliers, resulting in poor generalization to new, unseen data. Regularization techniques add constraints or penalties to the model which help to simplify it and reduce its variance. This improves the model’s ability to generalize from the training data to the test data. Common regularization methods include L1 and L2 regularization (types of regularization with linear models) while common types of regularization in machine learning are data augmentation (dataset), early stopping (model training), dropout and weight decay (neural networks). These techniques help to ensure that the model performs well on new, unseen data by avoiding overly complex models that fit the training data too closely. (Murel & Kavlakoglu,2023; Simplilearn,2024)

**7**. "Dropout" is a regularization technique that involves randomly setting a fraction of neurons to zero during each training iteration. This helps the model avoid overfitting by preventing it from becoming too dependent on specific neurons. By doing this, dropout encourages the network to learn more generalized and robust features, enhancing its performance on new, unseen data.(Dremio,2024)

**8**. "Early stopping" is a regularization technique where the training process is stopped when the model’s performance on the validation data stops improving. This prevents the model from overfitting to the training data by stopping the training before overfitting occurs. (Simic,2024)

**9**. For image analysis, convolutional neural networks (CNN) are very popular. They are particularly effective at recognizing patterns and features in images through the use of convolutional layers, pooling layers and fully connected layers. (Zvornicanin,2024)

**10**. A Convolutional Neural Network (CNN) works by applying a series of convolution and pooling layers to an input image. Convolution layers use filters (kernels) that slide over the image, computing the dot product to extract features like edges and textures. Pooling layers then downsample the convolved outputs to reduce dimensionality and improve computational efficiency. A CNN can have many layers, each learning increasingly complex features. Initially, filters detect simple patterns, and as layers deepen, they capture more complex, unique features that define the object in the image. This hierarchical feature learning enables CNNs to effectively recognize and classify images. (GeeksforGeeks,2024)

**11**. In that case a pre-trained model can be used. Pre-trained model is a model that has already been trained on a large dataset and can recognize many different objects. Some examples of such models are VGG, ResNet or Inception. In that way, by using a pre-trained model, the features that the model has already learned can be leveraged and applied to my friend´s own dataset to classify the images with tennis balls and zebras.

**12**. This code saves and loads a trained Keras model to and from a file. Specifically the trained model is saved to a file named model\_file.h5. The file contains the model’s architecture, weights, and training configuration. The model is also loaded from the file model\_file.h5 back into the variable my\_model and then can be used to make predictions or continue training.

**13**. According to the article, a CPU (Central Processing Unit) is a versatile processor designed to handle a broad range of tasks in a sequential manner, utilizing multiple cores for efficient multitasking. It is often referred to as the computer’s brain, as it interprets and executes most instructions from both hardware and software. Conversely, a GPU (Graphics Processing Unit) excels in parallel processing, making it particularly suited for tasks such as rendering high-resolution graphics and training deep learning models. With thousands of cores, GPUs can perform numerous computations simultaneously, greatly accelerating the training process for deep learning models compared to CPUs. This capability is crucial for handling the extensive and complex computations involved in machine learning and neural network training. In brief CPU is the main processor in a computer, responsible for executing most calculations and control logic and GPU is a specialized processor designed to quickly process and render graphics, often used in parallel computations within deep learning. (https://blog.purestorage.com/purely-informational/cpu-vs-gpu-for-machine-learning/)

**REFERENCES**

1.Coursera Staff. (2024). Deep learning vs. machine learning: A beginner’s guide. Coursera. <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide> . Uppdaterad den 1 apr 2024

2.Dremio. (2024). Dropout in neural networks. Dremio. <https://www.dremio.com/wiki/dropout-in-neural-networks/#:~:text=Dropout%20is%20a%20technique%20used,units%20during%20the%20training%20process>.

3.GeeksforGeeks. (2023). Difference between artificial intelligence vs machine learning vs deep learning. GeeksforGeeks. <https://www.geeksforgeeks.org/difference-between-artificial-intelligence-vs-machine-learning-vs-deep-learning/> . Last updated 10 March 2023

4.GeeksforGeeks. (2024). Introduction to convolution neural network. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>. Last upadated 14 March 2024

5.Murel, J., & Kavlakoglu, E. (2023). What is regularization? IBM. <https://www.ibm.com/topics/regularization#:~:text=Regularization%20is%20a%20set%20of,overfitting%20in%20machine%20learning%20models>. Published 16 November 2023

6.Pramod, O. (2023). Model parameters and hyperparameters in machine learning. Medium. <https://medium.com/@ompramod9921/model-parameters-and-hyperparameters-in-machine-learning-502799f982d7. Published 14 January 2023>

7.Saha, D. (2023). Machine learning: Training set, validation set, and test set. Medium. [https://medium.com/@dipan.saha/machine-learning-training-set-validation-set-and-test-set-48d10ca389db. Published 7 June 2023](https://medium.com/@dipan.saha/machine-learning-training-set-validation-set-and-test-set-48d10ca389db.  Published 7 June 2023)

8.Shah, T. (2017). About train, validation and test sets in machine learning. Towards Data Science. [https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. Published 6 December 2017](https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7.%20Published%206%20December%202017)

9.Simic, M. (2024). How to do early stopping? Baeldung. <https://www.baeldung.com/cs/early-stopping-regularization#:~:text=In%20this%20article%2C%20we%20talked,validation%20set%20doesn't%20improve>. Last updated 18 March 2024

10.Simplilearn. (2024). The best guide to regularization in machine learning. Simplilearn. <https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning#:~:text=Improving%20Model%20Generalization%3A%20Regularization%20helps,set's%20specific%20details%20and%20noise>. Published 15 May 2024

11. Terra, J. (2024). Keras vs TensorFlow vs PyTorch: Key differences among deep learning. Simplilearn. <https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article#:~:text=TensorFlow%20is%20an%20open%2Dsourced,runs%20on%20top%20of%20TensorFlow>. Last updated 7 March 2024

12.Zvornicanin, E. (2024). How does a neural network recognize images? Baeldung. <https://www.baeldung.com/cs/neural-networks-image-recognition#:~:text=Neural%20Networks%20and%20Images&text=They%20have%20become%20increasingly%20popular,network%20used%20in%20image%20processing>. Last updated 18 March 2024