DOI: https://doi.org/10.46354/i3m.2019.emss.017

ARTIFICIAL INTELLIGENCE: CONCEPT OVERVIEW, METHODOLOGICAL APPROACHES AND CHOICE METRICS

Michael Bogner[a], Martin Steiger[b], Franz Wiesinger[c]

[a],[b],[c] University of Applied Sciences Upper Austria – Embedded Systems Design
Softwarepark 11, A-4232 Hagenberg, Austria

[a]michael.bogner@fh-hagenberg.at, [b]Martin.Steiger@fh-hagenberg.at, [c]franz.wiesinger.@fh-hagenberg.at

ABSTRACT
During the last couple of years there has been a renaissance in the field of artificial intelligence, also called AI. A wide diversity of possible concepts to this topic leads to the compulsion to be properly informed about a variety of approaches. This paper focuses on explaining the primary and most relevant theoretical concepts in regard to artificial intelligence and to rate them based on derived criteria. To achieve this, the most significant manifestations of these learning concepts are analyzed to identify their core characteristics. Choice metrics are derived based on this knowledge and selected with regard to an industrial environment. Additionally, a methodical approach is developed to ease the user's choice of an appropriate concept according to the given criteria. The final result of this paper is a set of diagrams that illustrate the different artificial intelligence concepts based on the found criteria.

Keywords: artificial intelligence, deep learning, independent metrics, decision support system

1. INTRODUCTION

The term artificial intelligence, or AI, can be interpreted in many different ways. Due to this, the original meaning describing the process of teaching a system to learn new things, steps into the background slowly but steadily. The overall goal of AI is to create a system which is able to develop solving strategies towards a specific problem based on given examples. In general, a real artificial intelligence has to meet a given set of criteria. On the one hand, it has to be able to generalize problems to operate outside of the given examples. This property is also known as generalization. On the other hand, it is also capable of drawing conclusions from different situations to find appropriate solutions, which is called reasoning. Additionally, a real AI has to perform perception. This describes the ability to perform an analysis of environments and of the relationships regarding objects within those environments. An artificial intelligence is used when a systems purpose is to react to its environment using complex problem-solving strategies.

The rapid developments during the last couple of years and the predominantly positive prognoses for further use cases resulted in a strong public and scientific focus. This has the consequence that a variety of different approaches to the realization of AI applications are pursued, which leads to the question what methodology is the most suitable for a given problem. The motivation of this paper is therefore to support an AI user towards this decision.

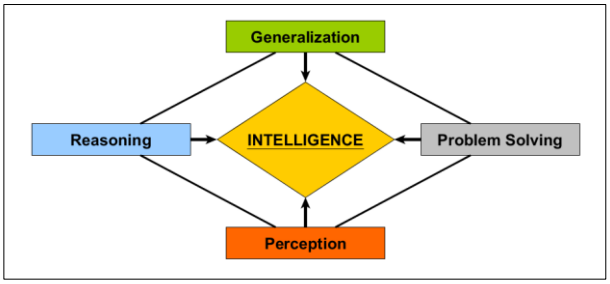

Figure 1: Criteria for a Real AI

To realize this goal, the most commonly used concepts and approaches are analyzed theoretically. To supplement these insights, the concepts are also reviewed with regard to their practical relevance, respectively how they can be utilized for industrial usage. This includes improving processes in companies, emend provided services and advance automatization especially in relation to industry 4.0. The analysis starts at basic machine learning approaches and concludes to artificial neural networks and finishes with the latest large concept called deep learning.

2. RELATED WORK

The artificial intelligence characteristics mentioned in the previous section have different fields of application and limitations. However, all approaches are based on the concept of machine learning. They are therefore only refinements or modifications of the original approach and some models in turn derive from these adaptions. Following this principle, artificial neural networks, for example, are based on the basic approaches of machine learning (Beck 2010) and deep learning is a refinement of these networks. This also leads us to the current state of the art, because the most powerful concepts are currently based on deep learning (Hinton 2007). Convolutional (LeCun 1998) and Long Short-Term Memory Networks (Hochreiter 1997), for example, will be discussed in more detail in the course of this work.

3. MACHINE LEARNING (ML)

To understand what defines machine learning, the model concept has to be clarified first. A so-called model describes the very basic procedure for solving a given problem using AI. What distinguishes a machine learning model from classic solution methods is the used walkthrough. A ML model uses well-known example data of possible solutions, while other methods derive an algorithm based on a given challenge and validate the theory by comparing it with expected values. Consequently, ML approaches are used when the given problem can not be described with a unified algorithm, but with proven examples or when the classical approach is to complicated. The process of learning solutions based on examples is called model training.

Machine learning models can be separated into two different classes. Descriptive models are used to identify or evaluate specific objects and predictive models are used generate forecast based on past knowledge. It is assumed that every problem is deterministic and no random influences occur. Overall there are five main goals of machine learning. To perform classification, a model has to determine, whether a given object can be assigned to a known class based on its attributes. A class describes a set of objects which share common and significant properties that are easy to identify. Regression models try to interpolate a characteristic based on numerical data. It has to be emphasized that such a model does not provide logical statements, but numerical results. Unlike the previous two types, clustering models try to separate an unknown set of data into classes with common attributes. If a model realizes classification or regression, knowledge is gained through examining existing examples. The goal of dimensionality reduction is to compress a large set of data, preserving the characteristic properties. To clarify the issue, an attribute of an object is called dimension in the context of AI. Consequently, dimensionality reduction models try to discard unnecessary attributes and return a cleaned dataset. At last location models are used to find specific data within a large database by trying to estimate the basic structure of the database.

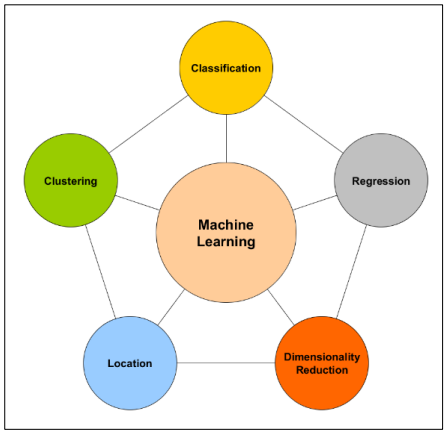

Figure 2: Goals of Machine Learning

To determine the quality of a given model, there are different approaches which are based on four indications. The true positives (TP) describe the number of positive results which are detected as such and the true negatives (TN) determine the number of negative results which have been recognized as such. Additionally, the false positives (FP) display the amount of positive results which have been labeled as negative and the false negatives (FN) act vice versa. Different quality metrics can be derived from these attributes:

- accuracy: $a = \frac{TP + TN}{TP + TN + FP + FN}$
- precision: $p = \frac{TP}{TP + FP}$
- recall: $r = \frac{TP}{TP + FN}$

As already mentioned in the previous chapter machine learning, models can be separated by their goal. But it is also possible to partition them by the used learning paradigm. When supervised learning is used, the models receive labeled training data. The labels abstract the expected output. In contrast to this, unsupervised learning uses training data without any labels, appropriate to the clustering models. This training method results in a black-box where the functionality of the output model is unknown. Mixed forms like semi-supervised learning are also possible. After the definition of the general model concept a few well known approaches will be analyzed.

3.1. Regression Algorithms

As already mentioned within 2.1, regression models, respective regression algorithms have the purpose to interpolate a characteristic based on numeric data. To determine which regression model is appropriate for a given task, one has to consider three aspects. The shape of the regression line is one of those key aspects. Linear regressions can be realized with low effort, but suffer from correlation issues, if the considered dimensions are not statistically independent and are very vulnerable regarding to heteroscedasticity, which describes outliners within the training dataset. To compensate these effects polynomial regression is used as an alternative, which is capable to overcome the heteroscedasticity. This shape has the huge disadvantage of possible over- or underfitting, hence when the chosen polynomial shape fits the training data too exactly, which restricts generalization or too inaccurate to fit the given problem (Sunil 2015). The second key aspect of regression models is the variable type, which can be numeric, discrete or categorical. Lastly the number of independent variables is also crucial for the regression model choice. The more independent variables have to be considered within the model, the more complex it becomes. The independent variables describe the number of data attributes used for a regression. To handle that kind of problems, there are specific approaches available, e.g. stepwise regression models.

The advantages of this approach are a simple implementation for linear and polynomial models and the intuitive selection of the appropriate model type based on training data visualization. In contrast the dimensionality of the model depends on the chosen algorithm, which may result in very large or unhandy models and interferences within the training data can reduce the model accuracy significantly.

3.2. Instance-Based Algorithms

Such models create and maintain databases in which the most significant records from the training data are stored. Due to the fact that instance-based algorithms compare the input data with the generated database, such approaches are commonly used for classification tasks. The input data receives the class which is most similar to its attributes compared with the samples from the database. Particularly noteworthy in such models is the fact that they can still be changed after training by adding or removing records

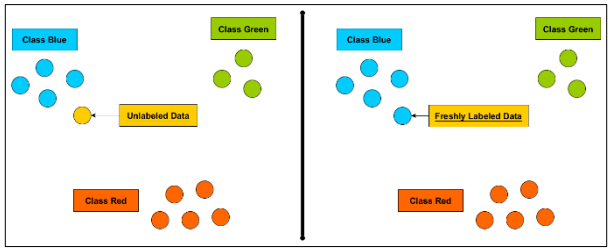

Figure 3: Instance-Based Approach

The advantages of this approach are the simple implementation and the flexible adaption to biased or noisy training data. Major disadvantages of this approach are the high memory consumption for storing the database and the high computation effort, because all records have to be compared with the input sample.

3.3. Regularization Algorithms

Regularization models are approaches which are capable of supporting other models by compensating a possible overfitting during the training process. These algorithms extend existing models to penalize complex developments during the training process, favoring simpler models. Simpler models lead to better generalization. Consequently, regularization methods are often combined with regression algorithms to compensate an over-adaption to the training data.

The training of a model is executed by evaluating the loss function, which describes the deviation of the result of a model from the actual labeled result. Ideally, this value converges to zero. If it becomes too high then the internal parameters of the model have to be changed to compensate this development. By adding certain offsets, so-called regularizations, an attempt is made to influence the loss function and thus also the parameter distribution within the model. The most common regularizations are L1 and L2, which have several advantages and disadvantages.

L1 results in sparse parameters which can be handled efficiently through appropriate algorithms, but is very vulnerable towards outliners within the model which can lead to instability during the training process. L2 loses the sparse parameter property of L1, but is less likely to become unstable.

3.4. Dimensionality Reduction Algorithms

The previously discussed approaches share the common problem that the model size scales with the complexity of the used dataset. Dimensionality reduction algorithms are used to determine whether all dimensions of a given data are necessary, depending on a specific model. Those algorithms can be separated into feature extraction or feature selection methods. Feature extraction tries to generate the best possible subset from a set of given attributes using three different approaches. The filter method calculates the subset which correlates most with the desired output, the wrapper method is capable of a dynamic adaption of the subset during the actual model training and the embedded method combines the other two approaches, also considering the current performance of the model (Kaushik 2016).

Feature extraction uses a different procedure. It does not try to generate a new subset from a given set of attributes, but to map an entire dataset to a new one with a reduced number of dimensions using different mapping rules. The variance, which describes how well the data are scattered, is maximized during this process.

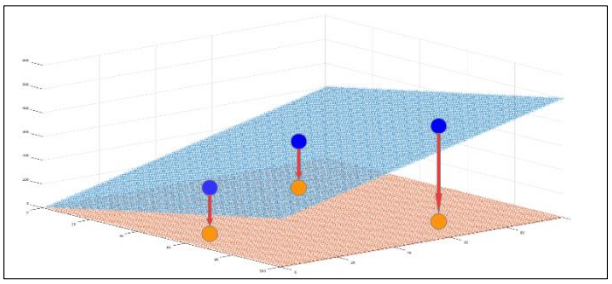

Figure 4: Feature Extraction with PCA

3.5. Summary

Taking the mentioned points from the previous chapters into account, simple machine learning models are easy to implement, but also limited in their performance. In particular, this relationship holds true with regard to the complexity of the considered datasets. If the complexity becomes too high, the models can either be trained poorly or not at all and the consumption of resources also rises sharply in some cases. Additionally, most models are only useful if the input data and the training data are uncorrelated, so each sample can provide a clear information for itself. Nevertheless, those models are recommendable for simple problems because they are very user friendly. Possible fields of industrial application are correcting invalid or noisy data, performing predictive maintenance for production lines through sensor networks and a regression approach or organizing simple databases.

4. ARTIFICIAL NEURAL NETWORKS (ANN)

As already mentioned in section 2, the problems and methodologies with regard to machine learning are generally designed and the proposed solutions can also be applied without comprehensive understanding. But those models are limited by the dimensionality of the datasets, because the training duration or the resource usage is too high. From this point on, artificial neural networks are acting. This approach is capable of handling a high dimensionality.

A neural network consists of a large number of subunits, so-called neurons, which fulfill a similar functionality than their biological counterpart. Figure 5 displays the structure of the neuron subunit. A neuron forwards all input values from external neurons or other information sources individually weighted to the nucleus, where the inputs are summed up. The result of this calculation is the input for the so-called activation function, which represents the activation threshold for the neuron. This output gets forwarded to the next neuron and so on.

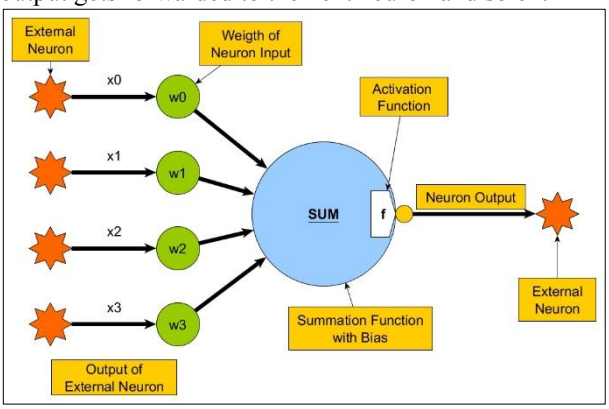

Figure 5: Neuron Structure

To achieve a desired functionality, the neurons need to be organized appropriately, respectively in layers with different tasks, e.g. input layers, hidden layers and output layers. The general structure of layers is displayed in Figure 6, but the functionality of the different layers is strongly determined by the ANN model.

It has to be mentioned that ANNs are a part of machine learning. Consequently, a more detailed comparison between the methodologies is required. Neural networks outperform simple machine learning models with regard to highly nonlinear problems like data with human influence, but suffer a reduced accuracy compared to the ML models if the problem can be described with a unified algorithm. Another notable information is the fact that artificial neural networks may use the local optimum of a problem and not the globally best solution, depending on the training process.

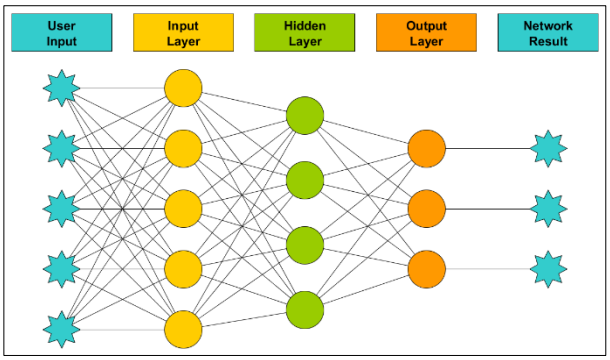

Figure 6: ANN Layer Structure

4.1. Pattern Associator

These neural networks are used for typical classification tasks of uncorrelated data without the need of hidden layers. The lack of layers reduces the model size and training duration significantly. Furthermore, pattern associators provide a solid generalization and tolerance with regard to neuron fading during training. This problem occurs when the input weights of a neuron converge to zero and the subunit is not able to react to input changes any longer. This is even more serious due to the fact that once a weight reached zero during the training process it is likely that it remains zero, which leads to a permanent deactivation of neurons, reducing the overall computation capabilities. Additionally, this model proves very robust against noisy or incomplete input data, but lacks of accuracy. Consequently, it can not be used without any adaptions, to return other then a rough tendency of the input data (Rey 2018).

4.2. Recurrent Neural Networks (RNN)

This type of neural networks differs from the basic pattern associator network by the bidirectional information flow. In general, the information provided to the network gets forwarded straight from the input layer to the output layer. But in this case, there are feedback loops between the layers, which are designed to recognize correlated input data. This functionality appears first within this paper, because every other model mentioned previously is not able to detect and handle correlated data. The type of feedback loop is essential for the functionality. A direct feedback loop describes a coupling between the output and the input of the same neuron, whereas indirect feedback loops act as a connection between the output of a neuron and the input of a neuron in the previous layer (Figure 7). Lateral feedback loops within the same layer and full feedback loops between every neuron are also possible. The purpose of this connections is to provide an internal state or memory within the neural network, which changes based on the last processed sample. Due to this fact, recurrent neural networks are capable of recognizing long-term relationships within the input data (Rey 2018).

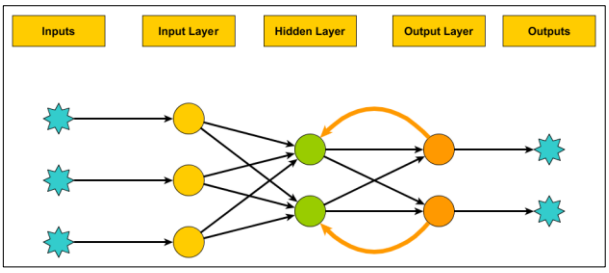

Figure 7: RNN Layer Structure

4.3. Competitive Networks

This type of artificial neural network is similar to the pattern associator model, but uses competitive learning during the training process. This learning paradigm support certain neurons within the ANN during the training that perform very well and optimizes the neighbors towards this neuron by modifying their weights. This circumstance is often used for data prefiltering in combination with other models, because competitive networks perform well with noisy data and can therefore provide clean input data for other models. It is also notable that competitive networks use unsupervised learning during the training process, consequently the most striking features prevail. Due to this fact it is also used for clustering tasks.

4.4. Self-Organizing Maps (SOM)

Self-organizing maps follow an entirely different principle than the other network types. In general, the result of the neurons within the output layer determines also the result of the entire network. But in SOMs the crucial network information follows from the activation states of all neurons. A typical task for self-organizing maps is to map inputs with high dimensionality into a space with low dimensionality. Very helpful in this context is the image of a landscape that is displayed on a map.

4.5. Summary

The decisive advantage of artificial neural networks is the fact that they are able to handle correlated data and to implement memory to recognize long-term dependencies. Also notable is the increased ability to cope with complex datasets. In contrast, the implementation of ANNs is more complicated than the simple machine learning models. To detect errors or possible biases during the training process, special toolsets are necessary. Otherwise the learning remains a black-box. Possible industrial applications are analyzing short-term dependencies in customer data, organizing big data, associative speech detection, creating intelligent inverse kinematics for robotic application or even optimizing delivery routes and other logistic issues.

5. DEEP LEARNING (DL)

As already mentioned in the last section, ANNs are already capable of solving complex problems, but they have got one decisive disadvantage. To perform multiple tasks within one model, a combination of different artificial neural network has to be used, e.g. one network for feature extraction and one for feature detection. Additionally, the mentioned vanishing gradient problem has got a higher impact on networks using a large number of layers, which ensure a better generalization. This problem results in a slow training process. To handle these disadvantages deep learning has been introduced to capsule an automatic feature discovery within the network. In conclusion, a deep learning model uses more hidden layers than a common artificial neural network.

5.1. Convolutional Neural Networks (CNN)

This type of deep learning model is commonly used for image recognition or for processing data with high dimensionality. It consists of three different layer types. The convolutional layer performs the feature extraction using so-called kernels that are generated during the training process. These kernels are often referred as filters, which return a filtered data sample with a possibly reduced dimensionality. After every convolutional layer follows a pooling layer which has the only purpose to reduce the dimensionality of a data sample even further by picking the most significant attribute from the filtered sample and passing it to the next layer. After some pooling and convolutional layers, the dimensionality of the sample has been reduced and it can be processed by classical ANN approaches. The general structure is displayed in Figure 8.

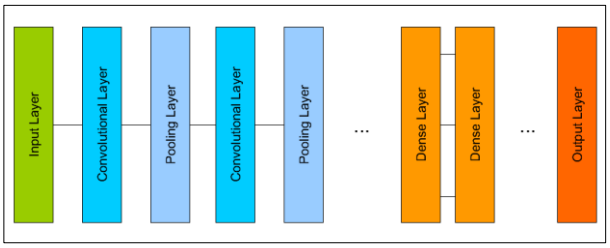

Figure 8: Structure of a CNN

5.2. Long-Short Term Memory Network (LSTM)

LSTM networks are a variation of recurrent neural networks, which leads to the conclusion that it contains a memory due to internal feedback loops. Large RNNs may cause a vast vanishing gradient problem because of their kind of training, called loop unrolling. The more feedback loops exist, the more training layers are resulting. The decisive difference to common RNNs is the kind of neurons used to implement the state capability. So-called LSTM units are operating similar to a computer memory. Data can be extracted, fed or removed from a LSTM unit using gates and the special structure of those units leads to a decreased vanishing gradient problem. Consequently, LSTM networks are used to handle very long-term dependencies within the input data which would lead to a vanishing gradient problem within common RNNs.

5.3. Summary

The reason why common ANNs are mostly used for problems with a medium complexity is the vanishing

gradient problem which leads to an excessive training duration. Deep learning approaches are able to compensate this problem, but suffer from a very high implementation complexity. Additionally, there are numerous model variants, which have to be carefully evaluated for a given problem. There exists a large number of possible industrial applications due to the outstanding generalization capabilities of DL approaches. For example, context recognition within video/audio sequences or texts, analyzing long-term dependencies in stock prices, weather forecasting, automated translations in different languages and live image recognition in autonomous driving cars.

6. CHOICE METRICS FOR AI APPROACHES

Taking the previous sections into consideration, a high number of different AI approaches are available to deal with a given problem. To be able to compare models even if the original purpose is different, the further proceeding is problem-oriented. Consequently, the first metric is resource usage, not only considering the physical storage but also training duration and hardware requirements. One has to be aware that a high score in this metric indicates a vast resource usage and should be considered carefully. The second metric is the correlation of the available and expected data, which assesses the ability to recognize and process correlated data. To conclude, data complexity and implementation friendliness also have to be taken into consideration. In Figure 9 an assessment for different types of machine learning approaches is displayed. The same approach has been performed for the artificial neural network types (Figure 10) and some deep learning models in Figure 11.

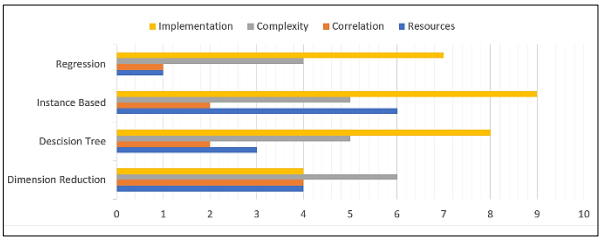

Figure 9: Choice Metrics for Machine Learning

As already mentioned within Section 2, machine learning models provide a high implementation friendliness, but suffer with regard to the correlation and complexity metric. An exception is the dimension reduction approach, but it has to be mentioned that most practical applications are using this approach to pre-filter the dataset for other models. A standalone application of a dimensionality reduction model is very rare.

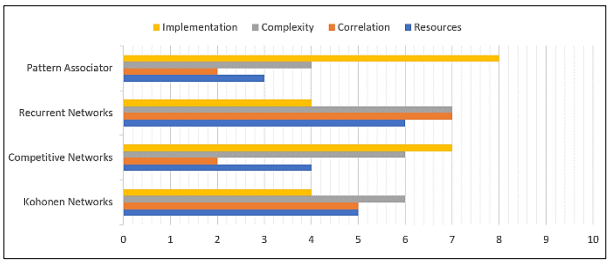

Figure 10: Choice Metrics for ANNs

Artificial neural networks outperform the simple machine learning models with regard to complexity and correlation capabilities, but have got a decisive disadvantage. The resource usage increases significantly and the simplicity of implementation decreases slowly, as described in Section 3.

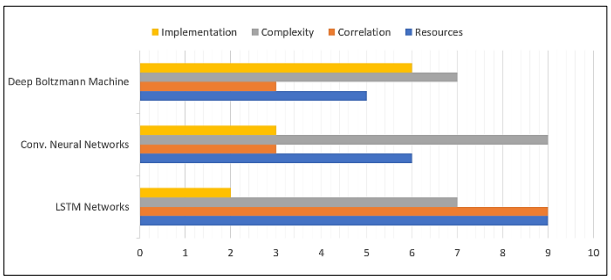

Figure 11: Choice Metrics for DL Examples

Although only a few specific DL model examples are considered, the tendencies are clear. In general, such models are difficult to be realized and therefore need careful consideration. As a balance, they are able to deal with complex data and additional long-term dependencies as displayed through the score with regard to the correlation metric.

7. METHODOLOGICAL SELECTION

After introducing the choice metrics for AI approaches, a methodological selection approach is also provided. At first an AI user has to identify the goal of the model based on Figure 2. Afterwards an inventory of the available data must be carried out to analyze the ground truth used for training and model validation. Above all, the structure and the amount of existing data record is important. For example, in this step, it is evaluated whether it is pixel data of pictures or a time series. Using the gathered information from the previous step, one can estimate the complexity of the dataset, which leads to the macrostructure of the model, respectively whether a simple machine learning approach is appropriate or a deep learning model has to be used. Taking the performed steps into account, one is able to choose a model using the choice metrics introduced in the last section. Afterwards a search for implementation strategies takes place, considering possible programming languages or frameworks, which might simplify the programming process. The next steps are implementation, model training, evaluation and comprehensive practical tests before the model can be considered as finished.

After the actual implementation and validation of the model, however, it must also be maintained and adapted to dynamic customer specifications for industrial applications. To ensure a consistent and reliable model management process Figure 12 introduces the AI lifecycle.

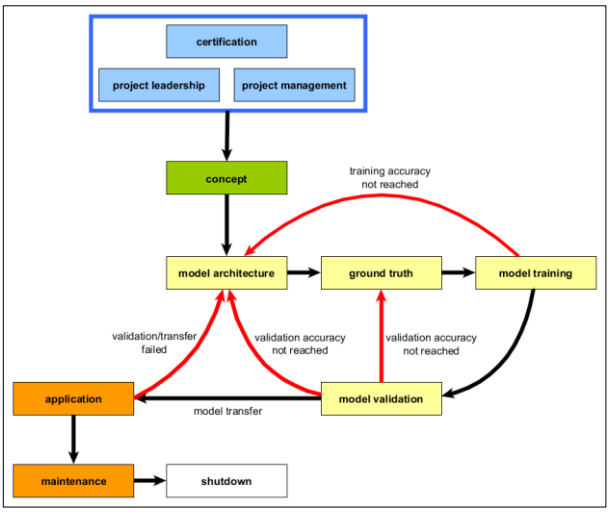

Figure 12: Model Management Lifecycle

8. APPLICATION EXAMPLE

To demonstrate the selection flow, lets assume the following problem. Images from the well-known MNIST dataset (LeCun 1998, MNIST), containing hand-written digits, have to be classified by those numbers. An example for a MNIST data sample is shown in Figure 13. The goal is a classification with regard to Figure 2 and there are 60.000 training examples with a dimensionality of 28x28 pixels available. Even though the resolution of one image if quite low, it will result in a model with high complexity. Therefore, simple machine learning approaches are no option. Remaining are ANNs or deep learning approaches. Most image classification problems are solved using CNNs, due to this it is also used in this example. Additionally, there is no correlation between the data samples. Consequently, the correlation choice metric can be low, the complexity metric has to be high and the resources and implementation are based on personal preferences. That confirms the usage of a CNN. The same choice flow can also be applied to industrial applications. But one has to be aware that it acts mostly as an orientation aid, which means it provides assistance for choosing the general model type and not a specific algorithm.

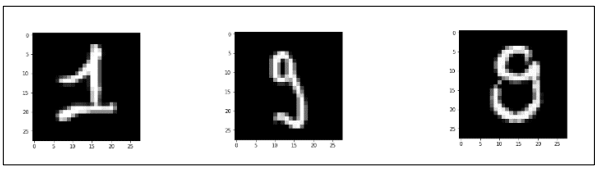

Figure 13: the MNIST dataset (LeCun 1998)

Although the actual implementation of a convolutional neural network is not the main intention of this paper, a possible solution is provided for the sake of this simple application example. As already mentioned, the dimensionality of a MNIST data sample is very small compared to more practical applications like real-image processing from a camera, e.g. in autonomous vehicles. Due to this, less convolutional layers than usual are required. To improve the network performance an additional dropout layer is added. This makes an overadaptation of the model to the training data unlikely. Figure 14 provides a brief overview of the proposed model structure.

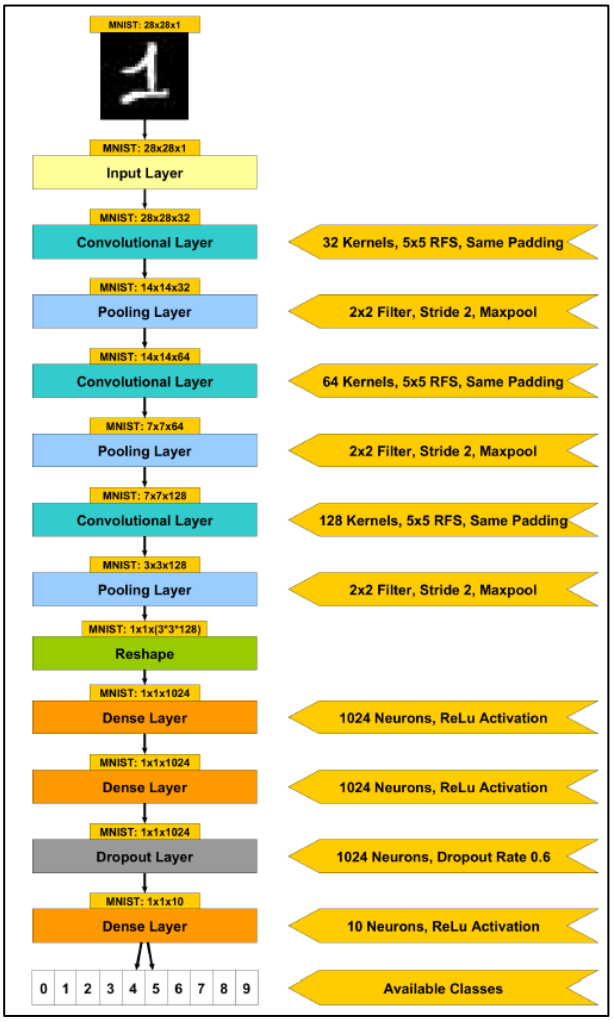

Figure 14: MNIST recognition CNN model structure

The final result of this model is a classification accuracy of approximately 97%. However, it must be taken into account that the example given here is often used as a learning example and that therefore detailed recommendations for the model design already exist. In general, accuracies at this level must be reviewed critically.

Another example for the choice flow would be for example the numerical evaluation of customer data. Due to the fact that information tends to be the most valuable

estate of large companies, it can be presumed that a maintained database already exists. It is also assumed that one data sample contains much information, for instance the purchasing behaviour of the customer for different products. The task is to forecast when and which number of independent products will be purchased. With regard to the made assumptions, a simple machine learning model is able to handle this problem. Due to the fact that the data are numeric a regression model seems to me the most appropriate solution. This assumption also confirms the statement that independent products are considered, which results in a low score perquisite with regard to the correlation metric.

9. CONCLUSION

Based on the results of the previous thought models, choosing an appropriate AI algorithm for a given task is still challenging, because is depends strongly on the problem type, the available ground truth and computation resources and also environmental perquisites. But is absolutely possible to take a rough direction based on the above AI choice metrics of resource usage, data complexity, data correlation and implementation friendliness. Also, a basic choice and maintenance flow has been provided, which is tailored to industrial applications.

Due to the fact that economic interaction is increasingly based on services and requires large amounts of information to be handled for this purpose, AI will be inevitable in most industries. Consequently one has to be prepared to be confronted with a tremendous amount of possible AI approaches, whereby the knowledge gained from this paper facilitate the orientation here.

REFERENCES

Sunil R., 2015. 7 Types of Regression Techniques you should know!, available at: https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/ [accessed: 16.05.2018]

Kaushik S., 2016. Introduction to Feature Selection methods with an example (or how to select the right variables?), available at: https://www.analyticsvidhya.com/blog/2016/12/introduct on-to-feature-selection-met hods-with-an-example-or how-to-select-the-right-variables/[accessed 16.05.2018].

Rey G. D. and Wender K. F., 2018. Neuronale Netze, Hofrege

Kohonen T., 2007. Self-Organizing Maps (3rd Edition), Berlin: Springer

Patterson D. W., 1996, Artificial neural networks: theory and applications, Singapore: Prentice Hall

Rojas R., 1996, Neural Networks. A systematic introduction, Berlin: Springer

Alpaydin E., 2010. Introduction to Machine Learning – Second Edition, Cambridge, Massachusetts, United States

LeCun Y. and Cortes C. and Burges C., 1998. The MNIST Database, available at: http://yann.lecun.com/exdb/mnist/ [accessed: 09.04.2019]

Hochreiter S. and Schmidhuber J., 1991. Long Short-Term Memory, Neural Computation, 9, 1735-1780

LeCun Y. and Bottou L. and Bengio Y and Haffner P., 1998, Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324

Hinton G., 2007, Learning multiple layers of representation, Trends in cognitive sciences, Vol. 11 No. 10