Deep learning applications in biomedical engineering - summary

Konrad Duraj

1 Introduction

1.1 History of artificial intelligence

Despite being studied for many years, artificial intelligence (AI) is still one of the most enigmatic topics in computer science. AI includes everything from extremely intelligent machines to search algorithms used in board games. Following the Dartmouth Conference, the field of artificial intelligence experienced substantial success for nearly two decades. An early example is the famous ELIZA computer program, created between 1964 and 1966 by Joseph Weizenbaum at the Massachusetts Institute of Technology [HK19]. ELIZA was a natural language processing tool capable of simulating a conversation with a human. In 1970, Marvin Minsky gave an interview to Life Magazine in which he stated that a machine with the general intelligence of an average human being could be developed within three to eight years [HK19].

1.2 Expert systems

An expert system seeks to capture the knowledge of a human expert - an individual whose knowledge within a highly specialized area is recognized to be superior. There are a few key components to any expert system:

1. Knowledge base - it is a construct that represents the in-domain knowledge about the phenomenon studied. It can take any of the following forms:
   - Hierarchical tree structure,
   - Relational database,
   - Graphs,
   - Fuzzy base.
2. Knowledge engineering - it is a process of capturing knowledge from human experts and representing it in a computer in a manner that allows to draw conclusions.

One of the most popular methods of creating an expert system is through the use of fuzzy logic.

1.3 Fuzzy logic

Fuzzy logic is a mathematical framework for dealing with uncertainty and imprecision in decision making. It was first introduced by Lotfi A. Zadeh, a mathematician and computer scientist, in 1965 [Zad65]. Zadeh came to the conclusion that many real-world issues are too intricate to be explained by conventional binary logic, which only accepts true or false values. Instead, he suggested 'fuzzy logic', a more open method that accepts varying degrees of truth and partial membership in categories. Depending on how well a specific item or notion fits into several categories, fuzzy logic assigns varying degrees of membership to those categories. Functions that fuzzify an input feature are called membership functions. The most popular membership functions used nowadays are listed below:

- Singleton,
- Triangular,
- Trapezoidal,

- Gaussian,
- Generalized Bell.

1.4 The downfall of expert systems

Expert systems were once hailed as breakthrough technology that would revolutionize the way people solve complex problems. However, their popularity declined in the early 2000s, and today they are being used less and less. One of the main reasons for the downfall of expert systems is their inability to learn from data; they rely on the predetermined set of rules and expertise provided by the human experts. Thus, these systems at their peak will be as good as the software engineers that created them and are incapable of providing new ideas in the scope of scientific knowledge. They are incapable of processing unstructured data such as:

- images,
- text,
- time-series,
- graphs,
- videos.

which decreased their utilization even more. In general, the death of expert systems can be largely linked to their limited capacity to learn and adapt, their reliance on human subject matter experts, and the development of more sophisticated and adaptable technology.

1.5 Machine Learning

Expert systems were the first attempt to make computers solve problems in our daily life. The biggest flaw in their development pipeline was the fact that they required humans at every step, from data collection, through feature extraction, to decision-making at the very end. Researchers working in the field of AI saw that flaw and developed a series of solutions that would automate at least one of these steps, decision making. Those algorithms expect a series of features describing a particular problem and try to map it onto the decision space by adjusting the decision boundary, which is a hyperplane that separates the input space into different regions based on a given classification rule. When given enough examples, those statistical models can actually become a good estimator for future data points. There are several modes of learning, such as:

- Supervised learning - uses a training set to teach models to yield the desired output. This training data set includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.
- Unsupervised learning - uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention,
- Reinforcement learning - uses training method based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions, and learn through trial and error.

There exist several algorithms for each mode of learning. Most popular of them are listed below.

1. Supervised learning
    - Linear regression,
    - Logistic regression,
    - Decision trees,

    - Random forest,
    - Gradient-boosting methods,
    - Support vector machine.

2. Unsupervised
    - K-Nearest Neighbours,
    - Gaussian Mixture Models.

3. Reinforcement
    - Q-learning,
    - Actor-Critic,
    - SARSA.

Although the immense success that methods of machine learning have brought in the Internet and big data era, they still were not enough to solve many of computer science problems. Machine learning algorithms still suffer from the inability to deal with the data in its raw form (audio, video, images, text). Moreover, these solutions still rely on the features provided by humans, thus assuming that software engineers and computer scientists are capable of creating a set of instructions that will extract meaningful information for a given problem.

1.6 Deep learning

Conventional machine learning techniques were limited in their ability to process natural data in their raw form. For decades, the construction of a pattern recognition or machine learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input [LBH15]. In recent years, deep learning has emerged as a leading approach to AI. Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With increasing amounts of data, these algorithms have surpassed the conventional machine learning models. Modern architectures have yet to reach their limits on some tasks.

Deep learning went one step further than traditional machine learning methods. Instead of actively trying to design features for classification task, it automates the feature extraction process and decision making in a single architecture. The machine generates an output in the form of a vector of scores, one for each category, after being presented with examples of data during training. Prior to training, it is improbable that the targeted category would have the best score of all categories. The error (or distance) between the output scores and the desired pattern is calculated using an objective function. To reduce this inaccuracy, the machine then alters its internal adjustable parameters. These programmable variables, frequently referred to as weights, are values that control machine input-output functionality. There may be hundreds of millions of these configurable weights and hundreds of millions of tagged samples in a typical deep learning system. To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector. Nowadays one of the most popular optimization procedures is called stochastic gradient descent, which was first formalized in [Rob51]. After computing an error value for a subset of the whole data set, we compute the gradient of the objective function with respect to all the weights of the neural network. Computation of the output is often referred to as ”forward propagation” and propagating the error signal through the network is often referred to as ”backward propagation” or ”backpropagation” for short.

Deep learning is making substantial progress in addressing problems that have long defied the best efforts of the artificial intelligence community. Because of its success in identifying complex structures in high-dimensional data, it can be used in a wide range of scientific and commercial applications. Examples of applications are listed below.

- language modelling [BMR+20],
- machine translation [DCLT18],
- text generation [BMR+20],
- image classification [BZK22, YWV+22, CND+22],
- object detection [WDC+22, ZSL22, YLDG22],
- semantic segmentation [CDW+22, YZW22],
- time series forecasting [ZCZX23],
- speech recognition [OEB+19],
- graph classification [ZBE+19],
- drug discovery [LLH+20]
- and many others.

Modern deep learning provides a very powerful framework for supervised learning. By adding more layers and more units within a layer, a deep network can represent functions of increasing complexity. Given a sufficiently large model and data set of labeled training samples, deep learning can be used to complete most tasks that involve quickly translating an input data to an output vector. For input of varying data type (text, image) there are several architectures that can be used separately or together to solve a particular task.

- Multilayer Perceptron - it is a feedforward neural network made up of numerous layers of connected nodes, where each layer is in charge of handling a distinct feature of processing input data. The input layer is the first layer that receives the input data. After passing through a number of hidden layers, the input data finally reaches the output layer. MLP is an effective tool for a variety of applications because it can simulate intricate nonlinear interactions between inputs and outputs,
- Convolutional Neural Networks (*CNNs*) - were designed for processing data with a specified, grid-like structure, are a particular subtype of neural network. Time-series data, which may be depicted as a 1D grid capturing samples at predetermined intervals, and image data, which can be represented as a 2D grid of pixels, or a grid of voxels for 3D medical images (MRI, CT). The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers,
- Reccurent Neural Networks - (*RNNs*) - this type of a neural network was specifically designed for sequence processing - where the prediction of the future depends on an arbitrary number of steps into the past. RNNs process an input sequence one element at a time, maintaining in their hidden units a state vector that contains information about all the past elements of the sequence.
- Transformers - one of the most recent revolution in the deep learning field was the transformer architecutre. is a deep neural network that, in its core replaced the recurrent connections with the self-attention mechanism. Self-attention allows the model to capture longer dependencies between tokens. It was introduced as the seq2seq model, which task was to translate between languages. That is why it was first designed as an encoder-decoder type model. The overall architecture is using stacked self-attention and point-wise, fully-connected layers for both encoder and decoder.

2 Deep learning applications in biomedical engineering

The main goal of this thesis is to evaluate different applications in which deep learning can be helpful to the field of biomedical engineering. It also shows that while deep learning is an incredible piece of technology, it has its shortcomings, which are crucial, especially considering possible applications in the medical domain.

2.1 Predicting Molecule Toxicity Using Deep Learning

We are exposed to a large number of chemicals every day - through our environment, food, medicine, etc. Knowledge about properties of certain substances is crucial in order to protect our bodies from exposure to dangerous factors. This paper describes an approach to the classification of molecule properties, namely its toxicity. The degree to which a substance can injure a live organism is indicated by its toxicity. The experimental assessment of the toxicity of a molecule is time-consuming, expensive and requires specialized personnel and tools. Deep learning can be applied to automatically evaluate this property. Dataset called ”SMILES Toxicity” was used for training the neural network, which is available on the popular data science website - ”kaggle.com” [Fan19]. It consisted of 7962 different molecules in total, from which 6998 were non-toxic and 964 were toxic. The molecules are encoded using the ”SMILES” format, which stands for the Simplified Molecular-Input Line-Entry System. It is an ASCII-based format, which defines a molecule’s structure - its atoms, bonds and connectivity using strings of characters. The data set was split into training and testing sets in the following way:

1. **Train**
   - Number of non-toxic molecules: 6760,
   - Number of toxic molecules: 937.
2. **Test**
   - Number of non-toxic molecules: 238,
   - Number of toxic molecules: 27.

Different combinations of model architecture and hyperparameters were tested. At the end of the experiments, the model which was a hybrid of convolutional and recurrent neural network achieved the best results. Because the class imbalance for this particular problem is significant, additional techniques were introduced to prevent overfitting and minimize the number of false negatives, such as:

- class weighting,
- cyclical learning rate.

These methods have improved the model performance, which initially was overfitting drastically on a non-toxic class. The final model achieved 77% accuracy on the test set while maintaining relatively high recall to other trained models and available solutions - 53%.

2.2 Recognition of Drivers’ Activity Based on 1D Convolutional Neural Network

Driving a car is a complex activity that involves movements of the whole body. Many studies on driver behavior are conducted to improve road traffic safety. This paper attempts to develop a classifier of scenarios related to learning to drive based on the data obtained in real road traffic conditions through smart glasses. The study was carried out under real road conditions according to Chapter 4 of the Act on Vehicle Drivers of the Republic of Poland in two groups of volunteers: ten experienced drivers (age 40 to 68) with a minimum of ten years of driving experience and ten learner drivers who attended driving lessons at a local driving school (age 18 to 46). The final data set consisted of 520 labeled electro-oculogram (EOG) recordings, acceleration and gyration signals, but only EOG signals were considered for the classification task. The recordings were divided into four classes in the following way:

- 120 recordings describing process of parking a car,
- 120 recordings describing process of driving through roundabout,
- 160 recordings describing process of driving through city traffic,
- 120 recordings describing process of driving through intersection.

The developed model was a three-layer, one-dimensional convolutional neural network optimized with a cross-entropy loss function. The results achieved for each of the categories are described in Table 2.2.

Category	Precision	Recall	F1-score
parking	0.98	0.91	0.94
roundabout	0.97	0.98	0.97
city traffic	0.95	0.98	0.96
intersection	0.95	0.98	0.96

To stop the increase in traffic fatalities and accidents caused by the growing number of vehicles on the road, the paradigm of the driver training process must be changed. The chance to assess drivers' perception can reveal important information about their attentiveness. Driver assistance technologies that are precise and affordable could help promote safe driving. However, real-time behavior and monitoring of driving conditions come with technical difficulties and the need to keep an eye on the driver's health, particularly dizziness caused by extended travel, abrupt changes in lighting, reflections from glasses, or adverse road conditions.

2.3 Heartbeat Detection in Seismocardiograms with Semantic Segmentation

Because it is acknowledged as a representative measure of cardiac function, heartbeat detection is a crucial component of cardiac signal analysis. Location of the QRS complex on an electrocardiogram is the gold standard for heartbeat detection. Seismocardiography (SCG), which uses vibrations to measure heart rate, is replacing electrocardiography as a reliable method of doing so due to the advancement of sensors and information and communication technologies (ICT). Thus, there is a natural need for new algorithms capable of extracting relevant parts of the caridac cycle from such data. This paper addresses these needs by developing a segmentation model for extracting the arguably most essential part, the heartbeat. The data set used in this paper was named "Combined Measurement of ECG, Breathing and Seismocardiogram", publicly available at PhysioNet.org. It contained 60 simultaneous recordings of ECG signals (Leads I and II), breathing signals, and seismocardiograms (on the z axis) acquired from 20 healthy volunteers of Caucasian race who were awake and remained in a supine position on a bed. Segmentation masks were generated by extracting R peaks from the ECG signal, using the Pan-Tompkins algorithm. The generated data set was split into train, validation, and test data. The model developed as a U-Net-style architecture algorithm [PT85]. which replaced the standard two-dimensional convolutional operator with its one-dimensional counterpart. The model was evaluated using the Jaccard index and the F1 score. The results of the test data set are described in Table 2.3.

Averaging	Jaccard	F1-score
Micro-averaging	0.99	0.99
Macro-averaging	0.97	0.98
Weighted-averaging	0.99	0.99

2.3.1 Semantic Segmentation of 12-Lead ECG Using 1D Residual U-Net with Squeeze-Excitation Blocks

The evaluation of the ECG is perhaps the most widely used biomedical signal to perform diagnostic measurements. This signal has its reflection in the mechanical action of the heart and can inform us about the physiological condition of this organ. The analysis of an electrocardiogram is a complex process that requires specific knowledge. In this study, we used the data set provided by Lobachevsky

University, available on the Physionet website. In this data set, the following elements of the 12-lead ECG signals were annotated:

- QRS,
- T-Wave,
- P-Wave.

The initial data set was divided into training, validation and test as follows: first, we divided the data into a training and testing set with 80% and 20% based on the patient, then we extracted 20% of the training set into the validation set. The developed model as an encoder-decoder-type architecture. The model was trained in two different variants, one with and the other without squeeze-excitation blocks. To measure the performance of this model, we used a Jaccard Index metric. The results are presented in Table 2.3.1.

Averaging	Macro-Jaccard Index	Micro-Jaccard Index	Weighted-Jaccard Index
Without squeeze-exciting	0.8	0.86	0.86
With squeeze-exciting	0.87	0.91	0.91

The created model achieves a high set of performance parameters (accuracy, AUC, specificity, and sensitivity) independently on all signal leads. Thanks to this, our solution can be used in any type of electrode configuration as a basis for heart rhythm and heart rate variability (HRV) classification, and many other parameters resulting from specific fragments of ECG. The model, due to relying only on convolutional layers instead self-attention and transformer architecture (which may be better suited for sequence translation) can be efficiently deployed directly on hardware.

3 A new paradigm

Deep learning systems in healthcare must be transparent and interpretable in a way that encourages public confidence in the algorithms and models that will be used to make critical decisions. This is especially true for healthcare applications, where explainability is essential for clinical judgments. When a deep learning model produces a prediction or delivers a diagnosis, it is vital that physicians are able to understand the reasoning behind the decision making process in order to provide the best possible care to patients. The adoption of these systems in healthcare can suffer if a model comes to a conclusion that cannot be justified or understood. This might result in mistrust and skepticism toward the technology. In biomedical sciences, it is crucial that decisions made by neural networks can be traced back to specific features, their combinations, and data points in the training data. This is important for identifying possible biases or weaknesses in the models and for confirming their precision and applicability in various clinical scenarios. Without it can be difficult to ensure the safety of deep learning systems in healthcare. Interpretable models can bridge the gap between clinicians and data scientists, which in turn can provide better care for the patient.

3.1 Limits of deep learning

Although deep neural networks have achieved spectacular results, they suffer from many flaws. Garry Marcus, a professor of psychology and neural sciences at New York University, wrote a paper on problems that deep learning has yet to solve [Mar18]. He pointed out several limitations, such as:

- They are data hungry - neural networks perform well under three conditions:
  1. The dataset on which they were trained is large and annotated,
  2. The dataset on which they were trained is diverse,
  3. The architecture of the model is large enough.

  Human or animal intelligence needs a few examples to solve a particular task, which is why Marcus thinks that more work should be put into unsupervised deep learning.

- They do not create compositional representations - this can be understood better when considering image classification task. Although architectures such as convolutional neural networks do build up a higher-level feature representation to determine the object's category, they do not deconstruct such an object into its distinctive parts,
- They are not sufficiently transparent,
- The prior knowledge cannot be well integrated,
- They are unable to perform symbolic manipulation,
- They are unable to distinguish causation from correlation.

In recent years, there has been a development of so-called explainable artificial intelligence (XAI) methods. These methods were supposed to allow developers of deep learning algorithms to gain insight into the inner workings of a neural network. Some of the popular algorithms are as follows:

- SHAP (*SHapley Additive exPlanations*) - a game-theoretic approach to explain the output of any machine learning model. SHAP provides a unified approach to explain the output of any model by calculating the importance of each feature in the prediction.
- GradCAM - it works by using the gradients of a specific class with respect to the final convolutional layer to generate a heat map of the important regions in the image. This technique helps us understand which parts of the image the model is focusing on and which features of the image are most important in making the prediction.
- LIME - is a method for explaining the predictions of any black-box model by approximating it with a simpler model.

3.2 Neuro-Fuzzy Inference System

Both neural networks and fuzzy systems have certain disadvantages which almost completely disappear by combining both concepts. They can be used for solving a problem (e.g. pattern recognition, regression, or classification) if there does not exist any mathematical model of the given problem. Neural networks and deep learning techniques are especially useful when dealing with high-dimensional, complex, non-linear data, where feature extraction done by hand-crafted algorithms does not scale. That being said, they can only be applied when a sufficient amount of observed examples are provided. On the one hand, no expert knowledge about the problem needs to be given. On the other hand, however, it is not straightforward to extract comprehensible rules from the structure of the neural network. On the contrary, a fuzzy system demands linguistic rules rather than learning examples as prior knowledge. Furthermore, the input and output variables must be described linguistically. If the knowledge is incomplete, wrong, or contradictory, then the fuzzy system must be tuned. Since there is no formal approach to it, the tuning is performed in a heuristic way. This is usually very time consuming and error prone. Fuzzy inference systems represent an important part of fuzzy logic. In most practical applications (i.e., control), such systems perform crisp nonlinear mapping, which is specified in the form of fuzzy rules encoding expert or common-sense knowledge about the problem at hand. There are three main types differentiated in the available literature.

- Cooperative Fuzzy Inference Neural Networks,
- Concurent Fuzzy Inference Neural Networks,
- Hybrid Fuzzy Inference Neural Networks.

Neuro-Fuzzy Inference Systems were once one of the most powerful and exacting techniques in machine learning. But their limits were quickly uncovered, and they have fallen behind more efficient and scalable machine and deep learning methods. The first big problem is the fact that they still cannot process the unstructured data, such as text and images. Another reason lies in their architecture. By merging multilayer neural network with fuzzy logic system into a single object researchers have putted a constraint regarding the deepness of the neural network. From recent developments in the deep learning field, it has become known that those algorithms given enough data create levels of abstraction which create linear separation in the input data space. By constraining the number of layers of a neural network, we are basically making it unable to correctly separate data points, thus worsening their overall performance.

3.3 Fuzzy Representation Learning

The main idea behind this algorithm is that a deep neural network based on input data must produce a complete fuzzy logic system which is then used to make a prediction. There are no rules specified by humans; the system converts information from input to fuzzy system to prediction in the end. The steps of the algorithm are as follows:

1. Feature extraction,
2. Defining fuzzification of the features,
3. Fuzzy adjectivness scores for features,
4. Crisp adjectivness scores,
5. Defining fuzzification of the classes,
6. Fuzzy adjectivness scores for classes,
7. Summation over class scores,
8. Output computation.

The general architecture of the described alogrithm is presented in Figure 1

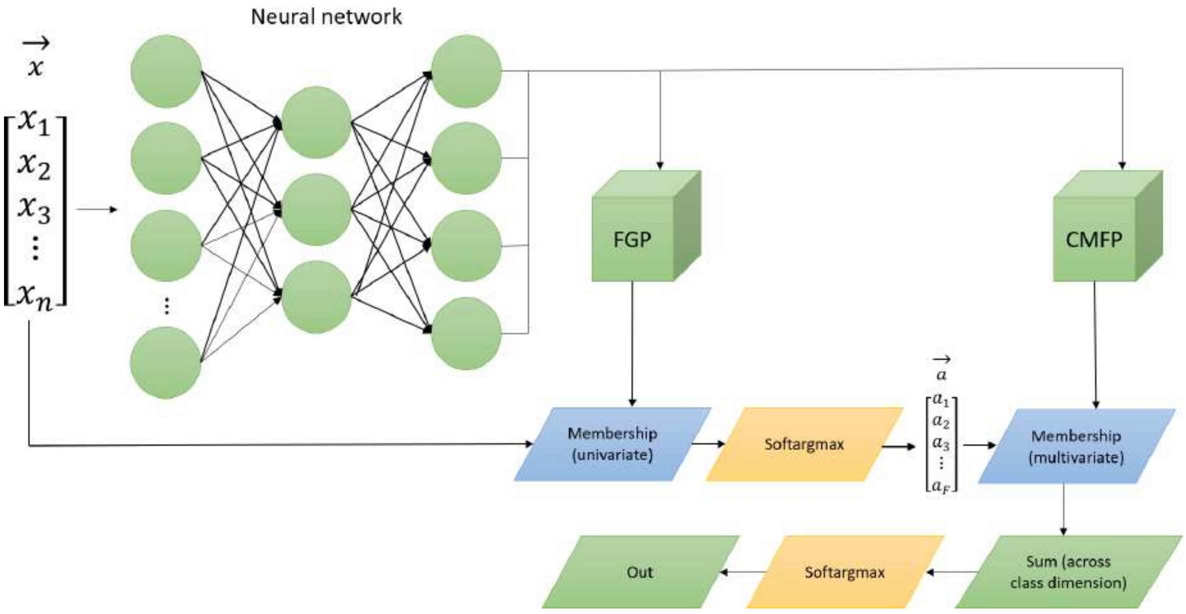

Figure 1: Overall system architecture.

3.4 An example

As a proof of concept, the XOR logic gate (exclusive OR) was adapted as a problem to be solved. The XOR is a logic gate that gives a true (high-state) output when the number of true inputs is odd; otherwise the state is false (low-state). It is also considered as the simplest nonlinear problem. To describe this problem in the setting of this algorithm, the granularity of feature membership functions (number of adjectives to describe a feature) was set to two (each feature can be either high or low). The granularity of the class memberships was also set to two because each combination of feature adjectives can be assigned to two possible states within one class (each class can be in a high or low state). The neural network was a standard multilayer neural network with two hidden layers, each consisting of 128 neurons followed by the ReLU activation function. The model was trained using the binary cross-entropy function, which is a standard for training deep neural networks for the classification task. The model was trained for 100 epochs, resulting in a final loss of 0.1231. The state of fuzzy representation at the last epoch is presented in Figure 2.

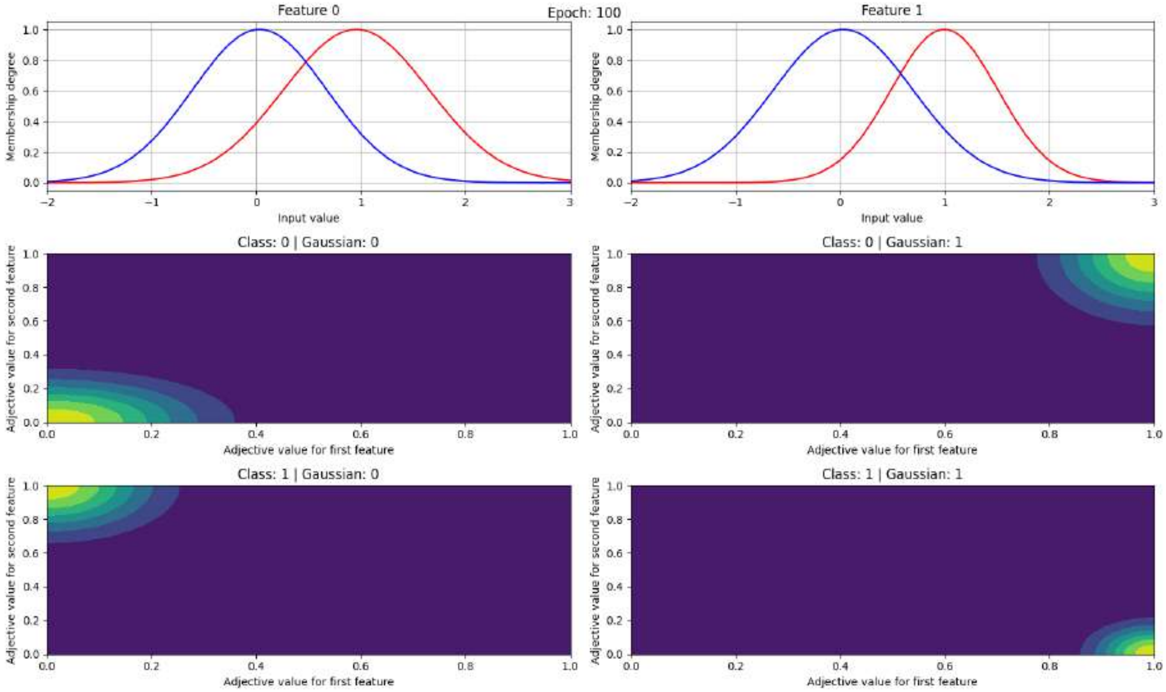

Figure 2: Fuzzy representation on the last epoch.

4 Conclusions

This thesis describes the potential for deep learning applications in the biomedical engineering domain. It starts by outlining the history of artificial intelligence and development of different approaches, such as expert systems, machine learning, and finally deep learning. Deep learning has great potential in the field of biomedical engineering. It can be used for tasks such as medical image analysis, medical signal processing, drug discovery, molecules classification, and disease diagnosis, potentially improving patient outcomes and accelerating the development of new treatments. But it also suffers from many flaws as well. This technology is often described as not being sufficiently transparent, not being well integrated with prior knowledge, or not generalizing well to out-of-distribution samples, to name a few. All of these reasons constitute a great debate about whether applying these methods in their current state to perform any kind of medical examination is ethical. On the other hand, fuzzy logic systems, an expert-system style algorithm, have proved to be resilient to out-of-distribution samples, representing knowledge in a transparent manner and being easily modifiable. Because of that a new approach has been proposed, called ”fuzzy representation learning” which tries to combine deep neural networks and fuzzy logic system that can be safely deployed in the healthcare applications.

References

[BMR$^+$20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020.

[BZK22] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Big vision. Available online at: https://github.com/google-research/big_vision, 2022. Accessed on:.

[CDW$^+$22] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[CND$^+$22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. 2022.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Fan19] Claudio Fanconi. Smiles toxicity. https://www.kaggle.com/fanconic/smiles-toxicity, 8 2019.

[HK19] Michael Haenlein and Andreas Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, 61:000812561986492, 07 2019.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[LLH$^+$20] Pengyong Li, Yuquan Li, Chang-Yu Hsieh, Shengyu Zhang, Xianggen Liu, Huanxiang Liu, Sen Song, and Xiaojun Yao. TrimNet: learning molecular representation from triplet messages for biomedicine. *Briefings in Bioinformatics*, 11 2020. bbaa266.

[Mar18] Gary Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018.

[OEB$^+$19] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[PT85] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985.

[Rob51] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[WDC$^+$22] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022.

[YLDG22] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks, 2022.

[YWV$^+$22] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022.

[YZW22] Haotian Yan, Chuang Zhang, and Ming Wu. Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention. *arXiv preprint arXiv:2201.01615*, 2022.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

[ZBE$^+$19] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. Hierarchical graph pooling with structure learning. *arXiv preprint arXiv:1911.05954*, 2019.

[ZCZX23] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? 2023.

[ZSL22] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training, 2022.