A Computational Biology Framework

Creating a platform for biomedical engineers to efficiently do their research

T.P.A. Beishuizen (0791613)
Biomedical Engineering - Computational Biology
Data Engineering - Information Systems
Eindhoven, University of Technology
Email: t.p.a.beishuizen@student.tue.nl

January 29, 2018

Contents

1 Introduction

At the Computational Biology department (cBio) of Biomedical Engineering (BME), many requests are made to analyse gathered data. This data usually stems from research in hospitals, but can also be from other BME groups and publicly available. Currently a standard is missing to efficiently analyse those data sets. With the vast number of data sets that are available, such a standard in the form of a framework on data analysis would be valuable. It would speed up projects and give them a higher chance to succeed the goal, due to improved efficiency. Before a framework can be made however a research must be done on all aspects that influence data analysis.

First an extensive background on important topics for such a framework will be discussed. Four different parts are explained why they are important for the creation of such a framework. These parts are: biomedical data (data used for analysis), data analysis goal (how does a goal influence the choice of analysis), data analysis tools (which ones are usable) and at last biomedical knowledge (what engineers of BME and third parties already know about data analysis). After the extensive background research, a research question will be formulated with several sub-questions for each of the four parts and a hypothesis as an answer for each of the four questions.

2 Background

Biomedical engineering can be seen as a specific part of engineering with a wide variety of topics. These topics can be theoretical, non-experimental undertakings, but also state-of-the-art applications. Not only research and development can be used, but also implementation and operation. Combining all of these different parts in one definition is hard. [1] For this project, the focus is mainly on research and development, also known as knowledge discovery. [2]

When a biomedical engineer starts a project, at the start usually only a data set and the research goal are known. To achieve that certain goal from the data set, several different aspects influence the project's course and development. At first obviously the data itself is a big part of such an influencer as the research is restricted to limitations from it. Examples of such restrictions are multidimensionality, set size, data heterogeneity, missing feature values and population handling. The other obvious influencer is the main research goal. Since the biomedical engineer wants to achieve a certain goal, the approach outcome must match that goal for the research to be successful. Most goals are focused around either data mining, extracting relations from available data, or modelling, creating a model within data features. A third influencer is the availability of data analysis tools. The steps to take from data to goal do not only include an approach, but also a tool to execute it. The choice of a certain tool has a big impact on the project, as each one of them has its own advantages and disadvantages. The two most well known tools within BME are MATLAB and Python, however some engineers have used R, Java or C++ and there are still other possibilities. A last big influencer is the biomedical knowledge. What experience the scientist already has with similar projects can greatly influence the choice of approach and framework. Knowledge of the supervisor and publicly known information on the research subject from books and articles also influence the approach, as already known outcomes do not have to be researched again.

Previous research projects on data mining have called for a model how to retrieve patterns from data collections. Frameworks to effectively do that have been proposed, usually with a number of steps.[3] These suggested frameworks do not specifically fit the cBio group though for being to broad [4] or being to specific. [5] A customized framework is needed.

2.1 Biomedical Data

A big aspect of choosing how to set up the data analysis is the data itself. The amount of data in the biomedical world is growing at an enormous rate, faster than biomedical engineers can analyse. Due to this rapid growth being uncontrollable, several additional challenges arose, aside

being more than the biomedical world can handle. These challenges are mainly focused around data volume, dimensionality, complexity, heterogeneity and quality.[6, 7]

Collecting data because it is possible can make data sets bigger than needed. Both in number of instances and features, data sets can be harder to understand or analyse when more is available.[6] This volume problem usually is tackled by taking sub-populations of the complete set. These subsets can either be focused around a part of the population (gender, age, race) or taken at random to still represent all of it. Due to the efficiency of analysis techniques and the rise in computational speed of servers[8], volume on its own becomes less of an issue. Volume does however become an issue when combining with heterogeneity and quality. [9, 10]

Not all data sets have a high number of instances that cause a big data volume. Sometimes there are relatively few instances, while the number of features is proportionally high. [11] Usually many of those features are not relevant enough for the research, however are still used for testing. Trying to remove features that are not important, will greatly help finding relations between the others and create more knowledge about the research topic. Lowering the number of features also makes the data volume go down, so analysis should be easier. Mainly an optimal features set should be selected to obtain the best results. [12]

Biomedical data can also be very complex. Useful results may be present, however it can be very hard to obtain it. Examples of complex data are images, several biomedical signals and temporal data. Details of the useful results that are present in images can for example be very hard to detect, the temporal data can vary quite much over time and the biomedical signals can be hard to combine with static biomarkers. [13] This aspect can benefit from exchanging knowledge with other research areas that specialize in mining of those complex data sets. [9, 14]

The biggest challenge encompasses aligning different data sets. No standard for data sets is available and therefore data sets differ greatly from each other. Data is weakly structured or even unstructured [10] and variables are processed differently due to other protocols or the collectors' preference of representation.[15] Also the variety of data is hard to combine when sources are fundamentally different. When parts of the data are images, another part is a table from the laboratory and a third part is textual remarks of the doctor, standardizing merging those three is much harder than merging three lab sets. Those merges are also very prone to errors, as imprecisions can be vastly different between those data sets. No tool can work directly with these raw data sets and preprocessing must almost definitely occur beforehand.[3, 9]

A last challenge is about data quality. The data is usually gathered by doctors and laboratory workers. Since the data is manually gathered by humans, the data have a relatively high error rate. Therefore the data can be quite noisy, values can be inconsistent, wrongly entered or even missing. [3] Not only human errors cause the data quality to drop, but the heterogeneity, as well. Two hospitals might have different protocols for the same treatment and sample different biomarkers for that protocol. Due to that difference, biomarkers may be missing for some of the entries. The time of data gathering is also a big factor as some biomarkers change greatly over time. The databases are usually also built for financial purposes and not for research, which can hurt the quality. [13]

These challenges within the data are greatly discussed.[14] Many proposals to tackle them are made, however none is actually widely adopted, yet, as a global standard for databases. Also, with the uncontrolled growth in biomedical data, it will become hard to have such a standard recognised all over the world. [15, 16, 17, 18, 19, 20]

2.2 Data Mining

Not only the data itself is important, but also the goal of a project. The data is analysed with a certain target in mind. This target off course heavily influences the data analysis approach that is taken. Two major aspects can be done for data analysis, data mining and modelling. For this project, the focus will be on data mining.

Data Mining has gained multiple definitions over the year. The most adopted definition is the following: "Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful

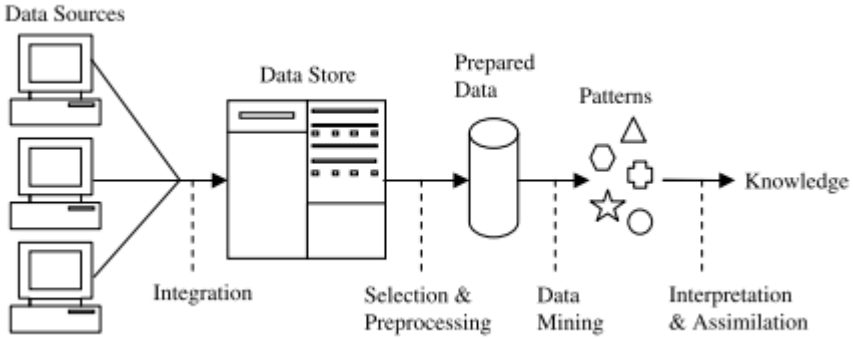

Figure 1: A schematic overview how a project involving data mining is done. Multiple data sets are integrated in one database. Feature selection and preprocessing takes place to prepare the data. Then the data is mind to find patterns. These patterns are then interpreted and assimilated to discover knowledge on the subject.[2]

to the data owner." [21] As can be read in this definition, the main goal is find new and useful insights and patterns through datasets, that can be used in further decisions or hypotheses.[6, 23] It is one of the links that brings data and knowledge together.[13] A data mining project follows a certain layout. (Figure 2.2), which in the end leads to further knowledge that aids the medical world.[2]

Since biomedical data is a wide scope, data mining has several specialisations in different direction. An example would be text data mining, used to find related articles with websites such as Pubmed and Google Scholar. These articles can mention similar genes, diseases or proteins and give additional information about them. Another examples would be microarray data mining. This type of data mining focuses on extracting entities and pathways that define a disease or other phenotype. Two other data mining types are proposed. One focuses on extracting useful information out of mass spectrometry data points, called proteomic data mining. A second focuses more on a molecular level and how these molecules would affect different cell types.[22]

Whichever splits are made in the data mining term for different areas in biomedical data analysis, the techniques are mostly based on statistical analysis and machine learning. A discussion of these two types has been made earlier (Appendix A)

2.2.1 Machine Learning

A machine learning technique gaining increasing recognition and popularity in recent years is the support vector machines (SVMs). SVM is based on statistical learning theory that tries to find a hyperplane to best separate two or multiple classes (Vapnik, 1998). This statistical learning model has been applied in different applications and the results have been encouraging. For example, it has been shown that SVM achieved the best performance among several learning methods in document classification (Joachims, 1998; Yang and Liu, 1999). SVM is also suitable for various biomedical classification problems, such as disease state classification based on genetic variables or medical diagnosis based on patient indicators. [6]

The accuracy of a learning system needs to be evaluated before it can become useful. Limited availability of data often makes estimating accuracy a difficult task (Kohavi, 1995). Choosing a good evaluation methodology is very important for machine learning systems development. [6]

Machine learning is a general-purpose method of artificial intelligence that can learn relationships from the data without the need to define them a priori [7]

2.3 Data Analysis Frameworks

One of the technologies that can help in carrying out the DMKD process is XML (eXtensible Markup Language) [6]. All formatted text documents consist of text and markup. Markup is the set of commands, or tags, placed within the text, that control spacing, pagination, linkages to other documents, font style, size, color, and foreign alphabets. On the Internet, the most popular markup language is the Hypertext Markup Language (HTML). In HTML, each start-tag begins with ¡and ends with¿; each end-tag begins with ¡/and ends with¿. Thus, for example, the sequence ¡B¿ ... TEXT ... ¡/B¿ causes the computer monitor or printer to display ... TEXT ... in boldface. [3]

The ability to share data effectively and efficiently is the starting point for successful analysis, and thus several attempts have been made to standardize formats for such data exchange: PSI-MI [35], BioPAX [42], KGML, SBML [40], GML, CML, and CellML [30]. [15]

Resources for Studying Statistical Analysis of Biomedical Data and R [24]

One aspect in comparing learning machines with each other deserves specific attention. If various machines are trained on training data, their performance can only be compared in a fair manner by applying all machines to the same test data. In this case, the above described procedures lead to valid estimates. [14]

2.4 Biomedical Knowledge

The traditional approach in biomedical science has been knowledge-driven and aimed at generating hypotheses from domain knowledge in a top-down fashion. Within biomedical data mining, one of the most interesting aspects is the exploitation of domain knowledge and the integration of different data sources in the data analysis process. As a matter of fact, data analysis is strongly empowered by the knowledge available in electronic format, which can be either already formalized, say through ontology and annotation repositories, or still informal but novel, as, for example, the one reported in Pubmed abstracts and papers[14]

Perhaps the most distinctive feature that accompanies medical data analysis is knowledge; data analysis in medicine strives to discover new and useful knowledge, while using available knowledge to guide the process and incorporate it into discovered models. In these terms, we perform data analysis to extract new findings that either refine or supplement existing knowledge on the problem domain, a population of patients, or a specific patient under consideration. [25]

In data analysis, knowledge is represented either implicitly or explicitly. By explicit knowledge we refer to knowledge already established, formalized and coded within some knowledge base. Such knowledge is used in some way in the process of data analysis. Researchers in machine learning often refer to such knowledge as “background knowledge,” and use it in learning either in data preprocessing, feature selection or modeling. While this approach seems very promising in medical applications, the number of methods, tools and applications of such an approach are few. Most applications of data mining in medicine, for instance, have focused on building models directly from data and do not consider any explicitly represented knowledge in the process. These applications leave the interpretation of the models and the placement of any new information found in this way within the context of the available knowledge to analysts and domain experts. [25]

•The process of feature selection, or determining the set of variables that are thought best, a priori, to contribute to an analysis depends on an intimate familiarity with the problem domain. Numerous methods are available for reducing the feature space. Some are cognitive, such as expert panel verification, while others are computational. Results of either type of approach may be combined with existing knowledge to select most informative and interesting features that cover the problem from various angles. •The selection of preprocessing methods that may be required for certain data mining tasks depends on expertise in using not only the preprocessing methods but the data mining tools that may require the use of such methods. Discretization of continuous variables is an example of such a method. While the knowledge required for this task would appear to be primarily analytic, problem domain knowledge is needed as well, since some

methods (such as discretization) can cause the loss of important scientific or clinical information. •The selection of an appropriate modeling methodology would also appear to be primarily in the analytic domain. However, the contribution of clinical or scientific knowledge to the modeling process cannot be underestimated. Examples of this are knowing when and how to model feature interactions. The selection and creation of these models requires collaboration between domain and analytic experts, due to the domain-specific realities of complex interactions and analysis-specific realities (and constraints) of computational complexity. •Finally, the selection of an appropriate methodology for presenting the results of an analysis requires expertise in such methodologies, but also in the clinical and scientific domain experts to whom the presentation is targeted. The primary goal of these methods is to reveal discovered knowledge in the optimal way to facilitate its communication to all members of the research team.[25]

3 Research Question

3.1 Hypotheses

References

[1] J. D. Bronzino and D. R. Peterson, *Biomedical engineering fundamentals*. CRC press, 2014.

[2] M. Bramer, *Principles of data mining*, vol. 180. Springer, 2007.

[3] K. J. Cios and G. W. Moore, "Uniqueness of medical data mining," *Artificial Intelligence in Medicine*, vol. 26, no. 1, pp. 1 – 24, 2002. Medical Data Mining and Knowledge Discovery.

[4] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, *et al.*, "Knowledge discovery and data mining: Towards a unifying framework.," in *KDD*, vol. 96, pp. 82–88, 1996.

[5] K. J. Cios, A. Teresinska, S. Konieczna, J. Potocka, and S. Sharma, "A knowledge discovery approach to diagnosing myocardial perfusion," *IEEE Engineering in Medicine and Biology Magazine*, vol. 19, no. 4, pp. 17–25, 2000.

[6] H. Chen, S. S. Fuller, C. Friedman, and W. Hersh, *Medical informatics: knowledge management and data mining in biomedicine*, vol. 8. Springer Science & Business Media, 2006.

[7] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, p. bbx044, 2017.

[8] D. Blythe, "Rise of the graphics processor," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 761–778, 2008.

[9] C. Turkay, F. Jeanquartier, A. Holzinger, and H. Hauser, *On Computationally-Enhanced Visual Analysis of Heterogeneous Data and Its Application in Biomedical Informatics*, pp. 117–140. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[10] A. Holzinger and I. Jurisica, *Knowledge Discovery and Data Mining in Biomedical Informatics: The Future Is in Integrative, Interactive Machine Learning Solutions*, pp. 1–18. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[11] W. Dubitzky, M. Granzow, and D. P. Berrar, *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media, 2007.

[12] Y. Peng, Z. Wu, and J. Jiang, "A novel feature selection approach for biomedical data classification," *Journal of Biomedical Informatics*, vol. 43, no. 1, pp. 15 – 23, 2010.

[13] I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.-F. Chang, and L. Hua, "Data mining in healthcare and biomedicine: A survey of the literature," *Journal of Medical Systems*, vol. 36, pp. 2431–2448, Aug 2012.

[14] R. Bellazzi, M. Diomidous, I. N. Sarkar, K. Takabayashi, A. Ziegler, A. T. McCray, *et al.*, "Data analysis and data mining: current issues in biomedical informatics," *Methods of information in medicine*, vol. 50, no. 6, p. 536, 2011.

[15] D. Otasek, C. Pastrello, A. Holzinger, and I. Jurisica, *Visual Data Mining: Effective Exploration of the Biological Universe*, pp. 19–33. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[16] L. Marenco, T.-Y. Wang, G. Shepherd, P. L. Miller, and P. Nadkarni, "Qis: A framework for biomedical database federation," *Journal of the American Medical Informatics Association*, vol. 11, no. 6, pp. 523–534, 2004.

[17] V. Y. Bichutskiy, R. Colman, R. K. Brachmann, and R. H. Lathrop, "Heterogeneous biomedical database integration using a hybrid strategy: a p53 cantcer research database," *Cancer informatics*, vol. 2, p. 277, 2006.

[18] W. Sperzel, R. Abarbanel, S. Nelson, M. Erlbaum, D. Sherertz, M. Tuttle, N. Olson, and L. Fuller, "Biomedical database inter-connectivity: an experiment linking mim, genbank, and meta-1 via medline.," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, p. 190, American Medical Informatics Association, 1991.

[19] F. Aubry, S. Badaoui, H. Kaplan, and R. D. Paola, "Design and implementation of a biomedical image database (bdim)," *Medical Informatics*, vol. 13, no. 4, pp. 241–248, 1988.

[20] D. Windridge and M. Bober, *A Kernel-Based Framework for Medical Big-Data Analytics*, pp. 197–208. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[21] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining*. MIT press, 2001.

[22] Y. Yang, S. J. Adelstein, and A. I. Kassis, "Target discovery from data mining approaches," *Drug Discovery Today*, vol. 17, no. Supplement, pp. S16 – S23, 2012. Strategic Approach to Target Identification and Validation: A Supplement to Drug Discovery Today.

[23] J. E. Vogt, "Unsupervised structure detection in biomedical data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, pp. 753–760, July 2015.

[24] M. Kobayashi, *Resources for Studying Statistical Analysis of Biomedical Data and R*, pp. 183–195. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[25] B. Zupan, J. H. Holmes, and R. Bellazzi, "Knowledge-based data analysis and interpretation," *Artificial Intelligence in Medicine*, vol. 37, no. 3, pp. 163–165, 2006.

A Biomedical Data Analysis

Biomedical Data Analysis

Discovering and reviewing several analysis techniques

T.P.A. Beishuizen (0791613)
Biomedical Engineering - Computational Biology
Data Engineering - Information Systems
Eindhoven, University of Technology
Email: t.p.a.beishuizen@student.tue.nl

January 29, 2018

Contents

1 Introduction

At the Computational Biology department (cBio) of Biomedical Engineering (BME), many requests are made to analyse gathered data. This data usually stems from research in hospitals, but also data from other research groups and publicly known data. With the vast number of data sets that are available, a framework on data analysis would be valuable. This framework would improve efficiency of this data analysis, as it would help choosing the best analysis approaches.

Before this framework is made however, first extensive research must be done on the already available analysis techniques. Many different approaches are used by researchers. These should be searched for and compared with each other, first using literature and second by actual use. This document focuses mainly on the literature part and explores the different techniques by analysing the articles.

At first the focus will mainly be on statistical methods. Different techniques will be explained as well as the programs that can be used for these statistical techniques. The second topic will be about machine learning, and more specifically deep learning.

2 Statistical Analysis

Statistical analysis consists of statements derived from large data sets that are often visualized in tables, graphs and charts. These statements are made on the basis of hypotheses and the analyses that either back up or reject those. The available data sets need to have a reasonably sized population so the statistical hypotheses can be tested. [1]

The statistical approach is mainly used for drawing conclusions on the data. When focusing on summarizing the data, inductive conclusions can be made which then can be used in the biomedical world. Techniques that are specifically useful for biomedical data are all combined in the term biostatistics. [1]

To be able to use statistical analysis, the data should be quantitative. The data should be a population of data points and usually the size influences the accuracy of the outcome. Since most biomedical data stems from experiments on patients, the data usually is quantitative and can be analysed by statistical techniques. [2]

In this statistical analysis chapter, first some concepts are discussed. Since for every data set the to be used computations are different, only the basic ones are used to keep the report from irrelevant explanations. Afterwards, five different programs or frameworks are discussed for their approach to statistical analyses. Afterwards these five are compared by literature and by personal testing.

2.1 Statistical Research Topics

Statistical research on its own has many applications. A selection of those that are relevant for Biomedical data are discussed in here. Important in statistical analysis is the presence of hypotheses. When doing this analysis, the researcher has an expectation of the outcome. An example of this would be having a data set of people of age and number of grandchildren. The hypothesis could be: H_0*: There is a correlation between people's age and the number of grandchildren.* Then at the end this or an alternate hypothesis H_1 can be either validated or rejected.[3]

2.1.1 Descriptive Statistics

Descriptive statistics is a collection of methods for data analysis based around its variables. Data set variables are either nominal (categorical or grouped), ordinal (Ranked relatively) or continuous (Ranked with equal intervals). Descriptive statistics takes these variables and computes details of their distribution (Table 1). [4] These then summarize the data for the scientist analysing it. The newly created summary can be used for further statistical research, for example by comparing different populations or variables. [1]

Table 1: Descriptive Statistics examples[4]

Variable distribution	**Analysis terms Center oriented**	**Analysis terms Spreading oriented**
Nominal	Mode	Frequency distribution
Ordinal	Median Mode	Frequency distribution Percentiles Minimum/Maximum Range
Continuous	Mean Median Mode	Frequency distribution Percentiles Quartiles Minimum/Maximum Range Standard Deviation

Aside from the aforementioned summary of the variables, descriptive statistics also consists of methods to properly visualize those. For nominal and ordinal data, histograms (Figure 1) and bar charts are often use to show in a easily understandable way how they are distributed. For continuous data box plots (Figure 2) can be used as well to show specific details of the distribution. [5]

2.1.2 Probability Concepts

When using biomedical data, it usually contains some phenomena that seem random and affect data values. Modelling and analysis of those data phenomena is hard, due to his unpredictability. A probabilistic approach can be used to tackle this. [6]

Values that seems to have a random part can be measured multiple times. When looking at multiple measurements an average can be found within their distribution in the sample space. The probability of a value to be within a certain interval is always a number between 0 and 1. The probability of a value to be in the sample space is 1, as it the complete interval. [6]

Another aspect of probability is about the several distributions. These show the total probability for a value to be on a certain interval. There are two different kinds of probability distributions: discrete and continuous. The difference between these two is the spacing between the possible values. Values from continuous distributions can take every value, whereas values from discrete distributions are limited and the possible values have spacings between them.[3]

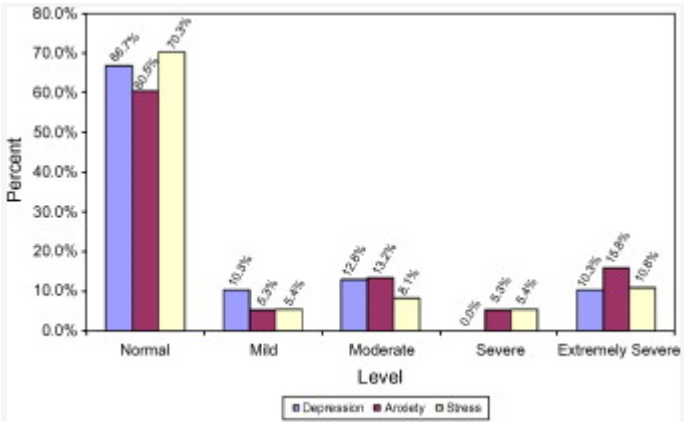

Figure 1: An example of a histogram of ordinal data. The depression, anxiety and stress scale of women with an acute coronary syndrome is shown. the data shows that over sixty percent rates themselves as normal. [4]

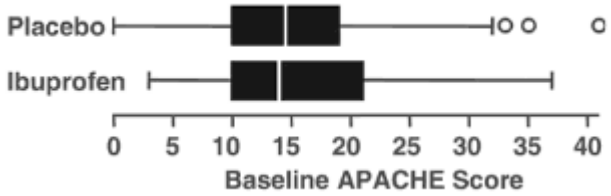

Figure 2: An example of a box plot of continuous data. The distribution of the APACHE score (Acute Physiology And Chronic Health Evaluation) was given for users of ibuprofen and placebo. The box plot doe snot only show the minimum, maximum and mean, but also quartiles and possible outliers. [5]

Binomial distribution is the most common example of a discrete distribution (Figure 3). This distribution can be seen as a number of n independent trials that have a success rate of p, with p being a number on the interval $(0, 1)$. This distribution has a mean of $\mu = np$ and standard deviation of $\sigma = \sqrt{np(1-p)}$. [3]

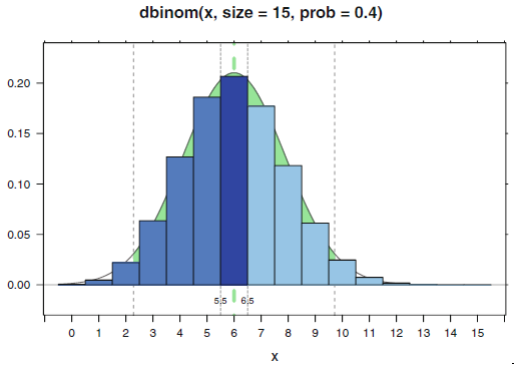

Figure 3: An example of a binomial distribution, a discrete probability distribution. In this example there were $n = 15$ trials with a success rate of $p = 0.4$. [3]

The most common continuous probability distribution is called a normal distribution (Figure

4). This distribution is bell shaped and symmetric around the mean μ, with a standard deviation σ. The standard normal distribution Z has a $\mu = 0$ and $\sigma = 1$ and can be made with a modification on a regular normal distribution X: $Z = \frac{X-\mu}{\sigma}$. [3]

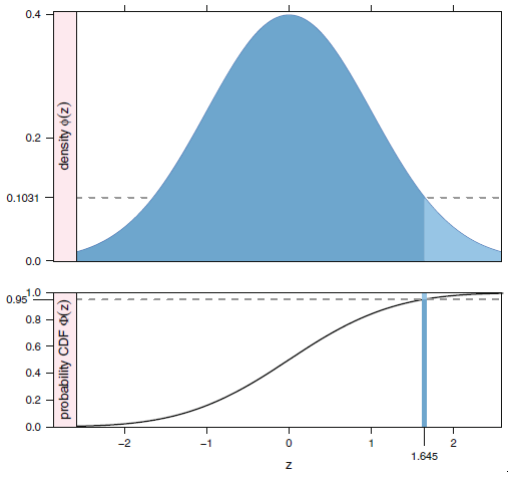

Figure 4: An example of a normal distribution, a continuous probability distribution. This example is the standard normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$. [3]

The third distribution that will be discussed is the student's t-distribution, another continuous distribution (Figure 5). This one is very similar to the normal distribution discussed earlier. The student's t distribution has a sample size of n and a sample standard deviation of s. The standard distribution can be found with the formula $Z = \frac{X-\mu}{s/\sqrt{n}}$ The larger the sample size n, the closer s will get to σ of a normal distribution, so an infinite sample size n is a normal distribution.[3]

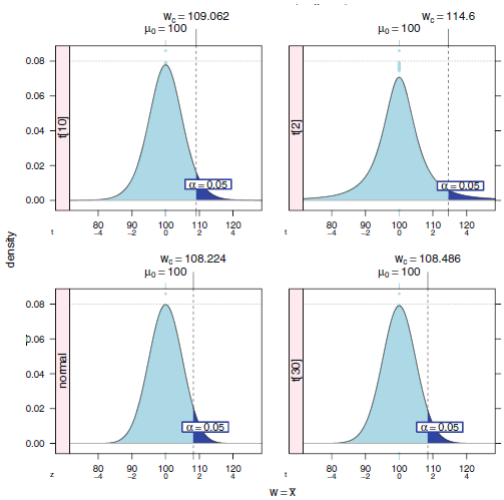

Figure 5: An example of a 3 student's t distributions and a normal distribution. With an sample size n becoming higher, the standard deviation s comes closer to normal distribution standard deviation σ. [3]

Several tests are used to find out if a data set can be fits a certain distribution. The "Goodness of fit" is then tested. The Chi-Square and Kolmogorov Smirnov are such tests. There may be more powerful and specialized ones though, for example the Shapiro Wilk for normality testing. [3]

2.1.3 Interval testing

On the previously discussed probability distributions specific intervals can be computed. A confidence interval can show the interval where a certain value should be in. This can be done for an unknown mean of a distribution or an unknown population proportion in a t distribution. These intervals are made similar to this one, the confidence interval of the mean, with mean $\hat{y}$, confidence number z, percentage cut-off α, standard deviation σ and population sample n (Equation 1).[3]

$$(\hat{y} - z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}, \hat{y} + z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}) \tag{1}$$

2.1.4 Group comparisons

The previously discussed confidence intervals can be used to find out whether two data sets are similar in terms of mean and variance. If the confidence intervals of the means of both data sets then the two data sets could be describing the same distribution. Its independence can then be proven by comparing their confidence intervals. [3]

The independence has two different parts. Two data sets can be independent comparing their means and they can be independent comparing their variances. The more important one of the two is when means are compared, as then they can be substantially different. To test whether two data sets are independent on their mean, a series of tests can be done (2). For these tests it makes a difference whether the unknown variance is common, different or the data is paired.[3]

Table 2: Formulas to compute whether two samples are independent by means. The parameters $s_{\Delta\bar{y}}$ and t_{calc} can be three different values (Table 3). [3]

		Tests		**Confidence Interval**	
H0	**H1**	**Rejection region**	p-value	**Lower**	**Upper**
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$	$t_{calc} > t_\alpha$	$P(t > t_{calc})$	$((\bar{y}_1 - \bar{y}_2) - t_\alpha s_{\Delta\bar{y}},$	$\infty)$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$	$t_{calc} < -t_\alpha$	$P(t < t_{calc})$	$(-\infty,$	$(\bar{y}_1 - \bar{y}_2) + t_\alpha s_{\Delta\bar{y}})$
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$\lvert t_{calc}\rvert > t_{\frac{\alpha}{2}}$	$P(t > \lvert t_{calc}\rvert)$	$((\bar{y}_1 - \bar{y}_2) - t_{\frac{\alpha}{2}} s_{\Delta\bar{y}}$,	$(\bar{y}_1 - \bar{y}_2) + t_{\frac{\alpha}{2}} s_{\Delta\bar{y}}$

Table 3: The values of $s_{\Delta\bar{y}}$ and t_{calc} for data sets with common unknown variance, uncommon unknown variance and paired data. [3]

Data set	$s_{\Delta\bar{y}}$	t_{calc}
Common variance	$s_{\Delta\bar{y}} = s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$	$t_{calc} = \frac{\bar{y}_1 - \bar{y}_2}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$
Different variance	$s_{\Delta\bar{y}} = s_{\bar{y}_1 - \bar{y}_2}$	$s_{(\bar{y}_1 - \bar{y}_2)} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$, and $t_{calc} = \frac{\bar{y}_1 - \bar{y}_2}{s_{(\bar{y}_1 - \bar{y}_2)}}$
Paired data	$s_{\Delta\bar{y}} = \bar{s}_d$	$s_{\bar{d}} = s_d/\sqrt{n}$, and $t_{calc} = \frac{\bar{d}}{s_{\bar{d}}}$

To find out whether variances of two data sets could be common, a new value F is introduced with the formula $F = \frac{s_1^2}{s_2^2}$. Since this $F = 1$ is the most ideal situation (both variances being equal), an equation can be made in which the boundaries can be specified (Equation 2). In this equation $F_{low} = F_{1-\frac{\alpha}{2},n_1-1,n_2-1}$ and $F_{high} = F_{1-\frac{\alpha}{2},n_2-1,n_1-1}$, the upper percentage points with the $n_1 - 1$ and $n_2 - 1$ or $n_2 - 1$ and $n_1 - 1$ degrees of freedom respectively. [3]

$$(\frac{s_1^2}{s_2^2}\frac{1}{F_{low}}, \frac{s_1^2}{s_2^2}F_{high}) \tag{2}$$

If data is paired (Table 2), to test if their means are paired is usually done with a t-test. A similar version, the two sample t-test, is the way to go when the distribution is known of the data set. A specific way to tests multiple groups at the same time is the one-way analysis of variance

(ANOVA). This t-test is preferred over other test due to his power compared to the other tests. [3]

Data can also not follow any specific distribution. In those cases Non-parametric statistical procedures will be used. An example would be the Sign test, when a certain median is taken and found out if there are enough data points above and under that median. This Sign test can also be used with paired populations and find out whether there is a significant number of data points from one of the sets lower than the data points from the other set. An upgraded version of this Sign test is called Wilcoxon Signed-Ranks Test, which also takes the magnitude of the data into account.[3]

If the data is not paired, the Mann-Whitney Test may be a good choice. This test is related to the two sample t-test, however uses medians instead of means. The more than two sample testing version of the Mann-Whitney Test is the Kruskal-Wallis test. This test should be used if some of the populations do not seem to be uniformly distributed enough for an ANOVA test.[3]

2.2 Statistical Programs

2.2.1 SAS

Statistical Analysis System (SAS) is a system used for statistical analysis. It can be used to perform a big variety of those with ease. Scientist can learn SAS quickly to perform small statistical computations for quick results. Within SAS a variety of algorithms are available for analysis. This simplicity of calling those algorithms makes SAS very time efficient. [7]

2.2.2 SPSS

SPSS is the statistical program with the most diverse users. SPSS is very user-friendly and easy to use when not familiar in either programming or when the there is only need for basic statistical analysis. Due to these advantages in educational research SPSS is widely used, especially by the social studies in need of these analysis techniques. [8]

2.2.3 R

R is one of the more well known statistical computing language for people who have experience in programming. It provides a language as well as statistical procedures to use for their analysis. Both SAS and SPSS created extensions for R, so it can be used in theirs as well. This is less intuitive, due those two being very similar. [9]

2.2.4 Python

Python is widely known for being an open source programming language with a relatively low bar to learn it. It was not often used for scientific analyses. Due to the maturity of several useful library as Numpy and Scipy however, python's market share is growing. Its open source element increases the usability over time as well. At last, since python is also used for other projects not involving statistics, scientists may already know the language. [10]

2.2.5 MATLAB

Matlab is not necessarily known for its statistical analysis, but more for its good way of handling biomedical data. Biomedical scientists and engineers are often familiar with MATLAB and its clear way of writing commands to analyse data. For statistical analysis extensions are available, an example would be the computational statistics toolbox. Also on the internet, many freely available user-created functions can be found, which can be used to the scientist's advantage. [11]

2.2.6 Comparisons programs

Many comparisons are made for scientists which statistical program is best to use. A good scheme was made by Jeroen Kromme, a data scientist of Cmotions that is involved with bringing data analysis to other scientists (Figure 6). He compares the fairly accessible programs SAS and SPSS with the more potential programs R and Python.[12]

Another article has been written by Kunal Jain that compares the commercially used SAS with the academic R and the high potential Python. He rates the programs on different aspects with what he shows that Python has the best potential of the three (Figure 7).[13]

Combinations of the statistical programs is also possible. SAS and SPSS are very similar to each other and switching between those is not hard to do. Both of these use extensions that allows the user to work with R functions. This is useful as R has more variety in their functions compared with SAS and SPSS. [9]

These explained surveys and more [14] [15] show that these programs all have a different approach. SPSS is used by the scientists and companies that don't need very advanced techniques. SAS extends the the usability of SPSS by a little and is provided by support, therefore companies that need a bit more advanced statistical analyses prefer to use SAS. R is a widely known useful

	SAS	SPSS	R	Python
Advantages	1. High adoption rate in major industries 2. Flow based interface with drag and drop 3. Official support 4. Handling large datasets 5. 'PROC SQL'	1. Used a lot in universities 2. Good user interface with extensive documentation 3. Click & Play functionality 4. Writing code made easy using the ´paste´ button. 5. Official support	1. Big community who creates libraries 2. Free 3. Early adopter in explanatory and predictive modeling. 4. Easy to connect to data sources, including NoSQL and webscraping.	1. Scalability 2. General purpose language 3. Easy to learn 4. Good in machine learning 5. Big community 6. Free
Disadvantages	1. Relatively high cost 2. For not-standard options not in interface, you'll need to write the code 3. Slow adapting to new techniques 4. Different programs for visualization or Data Mining	1. Relatively high cost 2. different licenses for different functionalities. 3. Syntax limited 4. Slow adapting to new techniques 5. Slow in handling large datasets	1. Can be slow with big datasets 2. Steep learning curve 3. No official support 4. No user interface	1. Not as strong in explanatory modeling 2. Choice of version: 2.7 or 3.5? 3. No user interface 4. No official support

Figure 6: Advantages and Disadvantages of the statistical programs SAS, SPSS, R and Python.[12]

Parameter	SAS	R	Python
Availability/Cost	3	5	5
Ease of Learning	4.5	2.5	3.5
Data Handling Capabilities	4	4	4
Graphical Capabilities	3	4.5	4.5
Advancement in Tools	4	4.5	4.5
Job Scenario	4	4.5	4.5
Customer Service Support and Community	4	3.5	3.5
Deep Learning Support	2	3	4.5
Total	**28.5**	**31.5**	**34**

Figure 7: Grades for SAS, R and Python on different aspects of using a program for statistical analysis.[13]

program that is mainly used in the academic world by scientists that know a bit of programming and need more in-depth analyses. Python is not widely used, yet but is on the rise due to its popularity for other programming sciences, as well.

2.2.7 Personal comparisons

All of these statistical programs have been tested personally, too. This was done in past projects for courses at the TU/e The final comparisons of the articles match the outcome of those findings.

SPSS and SAS can do numerous things statistically, however it is still very basic compared to R and Python. Since biomedical engineers are known to process their data in MATLAB and python, it is not worth it to pre-process data to use in SAS and SPSS when MATLAB and python can do about the same things. If the only need is fairly simple statistical computations, MATLAB will be sufficient for that. If the there is need for more complex computations, SAS and SPSS most likely would not be able to do those easily anyways.

If more complex statistical analyses are needed, it would be smarter to look at R and python for use. They both work on a similar level and both have an advantage over the other. If a scientist is not familiar with complex statistical analyses or with either language, R might be better for him. R is more widely used and therefore more material is available for help. Scientists that

do know python or are already working with it, are better of doing the statistical analyses with python. There is no need learning a new language when it would only take up more time.

3 Deep Learning

As computers are becoming smarter and smarter, the idea of computers learning how to tackle various problems is bigger than before. Also the size of data sets is growing two, which makes current ways of extracting information harder to use. The idea of giving computers a way to learn from experience and teach them how to cope with examples for treating real data the right way is called deep learning. [16] Deep learning is structurally used more in the biomedical environment to explain several concepts, especially considering images and the usage of neural networks.[17]

The whole idea of a computer gaining experience from finding patterns in the data is called machine learning. Deep learning is a sub section of machine learning as deep learning tries to solve problems by combining multiple small solutions. These small solutions are then connected in a network, that is often called a artificial neural network (ANN), resembling the neural network in a human brain.[16]

3.1 Basic Machine Learning

As deep learning is a specific kind of machine learning, knowing the important basics of machine learning is necessary. This is divided in three different parts, the task, performance measurement and the experience gained.[16]

3.1.1 Tasks

The task of a machine learning program is to solve problems that can not be solved by humans only. This way of solving requires a different mindset in programming. The program must be made so it knows how to solve a problem instead of manually telling the program how to solve it. Tackling a problem in the case of machine learning usually is done by having it process examples that are a collection of features. The example is a data point which details are described by those features. What needs to be done with those examples can be different.[16]

One task the program could do is specify a specific label from a number of categories. The examples all should be divided between several groups and machine learning tries to find the correct group for every example. This can be done multiple ways. Examples are in a discrete way (every example has one group) or in a probabilistic way (every example has a certain chance to be part of some groups).[16]

Another task the program could do is predict a numerical value for the examples. This is similar to classification only the answer is not limited to a number of categories. Instead it can be any value on a numerical interval. [16]

Classification and regression are the most commonly used tasks within machine learning. Other less used tasks could be to translate data to a structured representation, to detect errors in the examples or remove noise from the data. All off these tasks could be tackled using machine learning.[16]

3.1.2 Performance Measurement

When using machine learning, the performance of the algorithm is important. The program should not only use the data to create output, but this output must be correct as well. To measure this correctness, the accuracy is often measured. This accuracy is the percentage of examples that give the correct output. An error rate can be used as a performance measurement too, the number of examples that give an incorrect output.[16]

Data is used when training a machine learning algorithm. The algorithm is made to be as effective as possible for this training data, therefore the accuracy is supposed to be high for that data. This training data is usually just a subset of all possible data though and therefore the actual accuracy may be lower. To find this out, a separate test set can be made alongside the training set. This test set is then only used to find this accuracy.[16]

The performance measurements also require further thoughts than the accuracy and test sets. It may very well be that some categories in classifications are more important to be right (e.g. high risk patients) or the penalty for big deviations in regression may be bigger than needed. These things all need to be considered before measuring the accuracy. [16]

3.1.3 Experience

There are superficially spoken two kinds of machine learning, unsupervised or supervised. The difference between those is the experience they are allowed to have during its training. This training is done with a specific idea in mind that differs for these two categories.[16]

Unsupervised machine learning algorithms have the experience of a dataset with many features. With these features it can create structures within them. An example for this unsupervised learning would be clustering, to group data in different clusters according to their features.[16]

When using supervised data an additional label or target is given in addition to it regular features. With these, the algorithm knows what outcome every data point should get and learns to achieve that output with the available features. [16]

3.1.4 Challenges

Several challenges arise when using machine learning. Obvious challenges are using the right algorithm, having a data set with enough data that is also sufficiently detailed and and retrieving a good enough accuracy. Eventually the goal is to have a high accuracy on the created test set. The performance quality on the test set is called generalization.[16]

There are three type of errors when working with an algorithm. At first there is the training error, the error rate for training data. This training error is always better than the test error, the error rate for the test data. The difference between those is called the generalization error. When an algorithm is made, the training error should be kept as low as possible as well as the generalization error.[16]

The challenge to have the training error as low as possible is to not under fit, which means that the algorithm does not achieve a low enough training error. The challenge for generalization error is to no over fit, which means that the test error is not close enough to the training error.[16]

3.2 Neural Networks

The name neural networks stems from the connections between neurons in the brain, a big research area in modern day biology. In the mathematical world, artificial neural networks can be made for modelling data analysis. These models that are created with the neural networks use a specific equation (Equation 3). The output (computation) is a combination of already stored information (storage), new information (transmission) and process it with available methods in the model (processing).[18]

$$\text{computation} = \text{storage} + \text{transmission} + \text{processing}. \tag{3}$$

Zooming closer into neural networks, the idea of a single neuron can be shown (Figure 8). These neurons have three different important parts. The first part are the input values x_i, which are first modified a bit with weights w_i. The third part is the actual function f that creates output for the weighted input values. These neurons then put together create a network with eventually an output (Figure 9).[18]

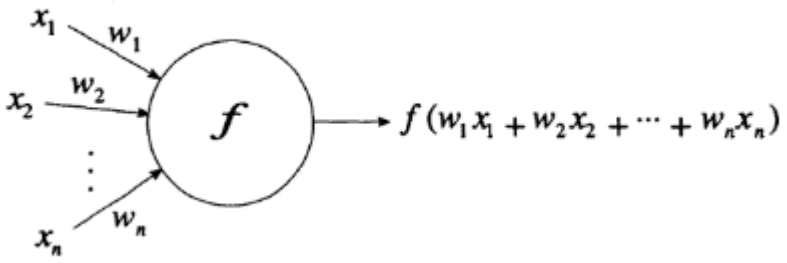

Figure 8: An example of a basic neuron in a neural network. Input $x_1...x_n$ together with a weight $w_1...w_n$ are put into function f to create output. [18]

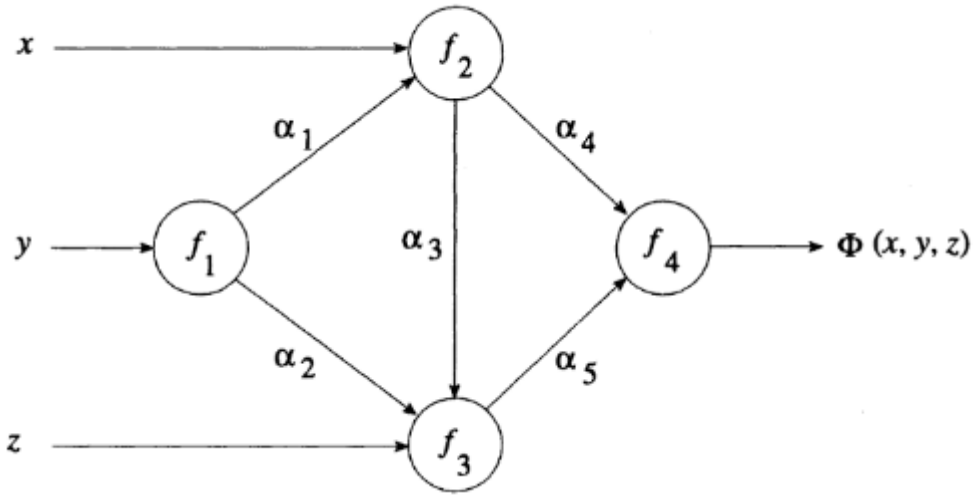

Figure 9: An example of a neural network consisting of neurons (Figure 8). [18]

These created networks are divided in layers. The first layer is the collection of all neurons that have an external input value. The last layer is the collection of all neurons that generate output values. All layers in between are called hidden layers, as a user cannot see those neurons from the outside.[19]

As explained in the section about machine learning, the neural network must be trained (Section 3.1). This can be done after the architecture is defined. Again, with the data, a training and a test set must be made, as well. With those, the neural network can be properly trained to create the right output when giving values as an input. [19]

3.2.1 Convolutional Neural Networks

A more specific variant of neural networks is a convolutional neural network (CNN). CNNs combine neural networks and convolutions to reduce noise in the signal and approximate the actual input better. Convolutions use kernels to use the values close to the input it wants to approximate. It also uses pooling, which makes the input smaller by approximating a location and its nearby inputs into one. [16]

3.3 Deep Learning Frameworks

A wide selection of deep learning frameworks are available. Most of these frameworks are implemented in Python, with some exceptions. A selection of the five most popular are explained, together with their advantages and disadvantages.

3.3.1 Skikit-learn Python

Python itself has a package that can be used for machine learning, named Skikit-learn. It is high-level interactive and is maturing in a quick rate, due to its usage in both the academic world as in the industry. It also has some C++ libraries incorporated, but makes most use of the three packages Numpy, Scipy and Cython. Due to his high-level and ease of use, a drawback is its computational efficiency and therefore is not the best choice for testing large data sets.[20]

3.3.2 Theano

Theano is one of the open source machine learning frameworks written in Python.[21] Its programming is declarative. [22] This one uses the famous NumPy syntax for higher computation speed in its language. It was developed mainly to simplify the implementation of the the algorithms used for well performing machine learning. Two tasks that are intensive for the processor, either training a multi-layer neural or a convolutional network. Theano works through a pipeline on compilation (Figure 10). At first it puts the graph in standard form (Canonicalization). Next it improves the stability of the computation (stabilization). Thirdly it replaces expressions with faster ones (Specialization). Fourth it moves the computation to the GPU, if compiled for GPU (GPU transfer). In the last step it loads Python modules with specialized implementations (Code Generation). [21]

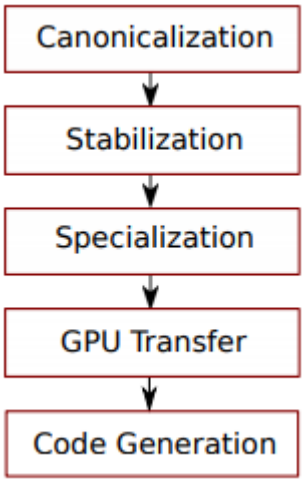

Figure 10: The steps Theano takes in compilation for functions for GPU. [21]

Theano has been seen as a base for many other neural network creation frameworks. Examples are Keras, Lasagne, GroundHog, Blocks and Pylearn2. It utilizes from the benefits of being written in Python, such as creating a friendly enviroment for fast and easy interaction with data. However, since threading is done by Python, it is not possible to have multiple interpreters going on at the same time. The graph optimization time, code compilation and memory usage time all can be improved. [23]

3.3.3 TensorFlow

TensorFlow, also open sourced, is developed for experimentation on new models, training those models with the help of large datasets and at last move those models into production.[24] Tensorflow is declarative and made in C++, while also having interfaces in Python. [22] Google makes regular use of TensorFlow, as an example. However many more programmers use TensorFlow for their applications. It is a follow-up of the earlier system DistBelief and simplified and generalized that. TensorFlow is mainly supporting training and inference on a larger scale. It uses a dataflow graph for representing the computation within the algorithm and the operation state of the algorithm. [24]

While executing a TensorFlow application, two different phases can be distinguished. The first phase of the two makes a to be trained neural network and the update rules in the form of a dataflow graph. The places for input are reserved by place holders for state representation. The second phase is the earlier neural network after optimization. Because of these two phases, the execution phase can be optimized with the information of the computation. There will not be any intermediate results slowing down the process. [24]

Not only Theano, but TensorFlow as well has some wrappers, too. Keras (also for Theano) and Pretty Tensor both use TensorFlow as a base for creating neural networks. [22]

3.3.4 Caffe

Caffe is a framework based on the C++ library together with Python and MATLAB. Caffe, too, is open sourced and its programming is imperative. [22] It is very useful for researches as its code is modular and its network definitions are well separated. The test coverage of Caffe is useful as well, as every new module has a test and is not accepted without a test. The bindings with python and MATLAB makes constructing networks and classifying inputs easier. At last does Caffe provide pre-trained reference models. This makes it easier to reproduce research. [25]

3.3.5 Torch

Torch is an object-oriented framework for machine learning and is implemented in Lua, but also has interfaces in C. Torch is also imperative.[22] A modular strategy was used to simplify any modifications of existing algorithms. There are four classes chosen. The first class is about data handling (DataSet). The second one is the black box that gives an output (Machine). The third is the class that is used to find the optimal set of parameters (Trainer). The last one is about printing the measures of interest (Measurer). These classes give a very clear general idea. Several examples of usable machines are the gradient machines, support vector machines and distributions. [26]

3.3.6 Comparing Frameworks

Kovalev compared the likings of Theano (with Keras extension), Torch, Caffe, TensorFlow and the not discussed Deeplearning4J based on convergence and prediction time, classification accuracy and framework complexity (lines of code). Both the depth (number of internal layers) and the width (number of neurons within fixed number of layers). When regarding training, prediction and testing times, all but DeepLearning4J were performing quite well. When watching Classification accuracy, Torch and Tensorflow performed visibly instable and Caffe seemed to be a bit worse overall. The last comparison, framework complexity was best for Theano, followed by TensorFlow and then Caffe. From best to worst the article rated the following order: Theano (with Keras), TensorFlow, Caffe, Torch, Deeplearning4J. [27]

The creator of Caffe compared its own framework with other known frameworks. The focus was mainly on basic parts of the programs, namely BSD licence, core and binding languages CPU and GPU coverage, Open source, training and the availability of pretrained models. Caffe has the advantage in comparison with the earlier described Theano and Torch regarding binding languages and the availability of pretrained models.[25]

Framework	License	Core language	Binding(s)	CPU	GPU	Open source	Training	Pretrained models	Development
Caffe	BSD	C++	Python, MATLAB	✓	✓	✓	✓	✓	distributed
cuda-convnet [7]	unspecified	C++	Python		✓	✓	✓		discontinued
Decaf [2]	BSD	Python		✓		✓	✓	✓	discontinued
OverFeat [9]	unspecified	Lua	C++,Python	✓				✓	centralized
Theano/Pylearn2 [4]	BSD	Python		✓	✓	✓	✓		distributed
Torch7 [1]	BSD	Lua		✓	✓	✓	✓		distributed

Figure 11: The comparisons between different machine learning frameworks from Caffe creator point of view. [25]

Bahrampour tested Caffe, Theano, Torch and the not discussed Neon on their extensibility, hardware utilisation and speed for both CPU and GPU usage. He stated that Caffe, Theano and Torch were at that moment the top three well developed and used frameworks. Neon was added because of its potential. The speed was measured by a forward time (check time it takes for a pre-selected batch) and a gradient computation time (time for each measurable parameter). The conclusion of the research had six parts: Theano and then Torch are the most extensible. For both CPU- and GPU-based training and deployment Theano and Torch are fighting for the number one spot. Torch would benefit from expansion in both documentation and debugging, though. [28]

Another research that compared deep learning framework was Shi. The frameworks Caffe, TensorFlow and Torch were tested by him, as well as the not discussed CNTK from Microsoft and MXNet. The tests that were done about speed were on CPUs, one GPU and multiple GPUs on both synthetic and real data. The overal outcome was that Caffe, CNTK and MXNet performed better than TensorFlow and Torch, although TensorFlow's production ramped up with more threads. [29]

Similar to Bahrampour[28], Fox compared TensorFlow, Caffe, Theano, Torch and the not discussed CNTK, Deeplearning4j, MXNet and H2O, for them being open-source, relatively mature and adopted by the community. He mainly discussed the different frameworks and afterwards made a table with all those findings (Figure 12). [30]

Parvat had a more listings approach in comparing the frameworks Theano, Caffe, Torch, Tensorflow and the not discussed Deeplearning4J and NVIDIA cuDNN. This time they were compared on their platform, interface, modelling capability, support for CUDA, OpenMP, OpenCL and support for pre-trained models. He concluded that TensorFlow was very flexible, due to being more than a deep learning framework. Theano is very helpful for creating models fast, especially with the libraries, and therefore TensorFlow and Theano are the most flexible. Torch is good because of the pre-trained models, however misses being written in a mainstream language. Caffe is useful when working with images, and also has a lot of pre-trained CNN models. Deeplearning4j has its advantage being the only one working with Java and Scala. At last TensorFlow has the advantage of being able to work with a distributed environment. [31]

A last article to discuss about comparing deep learning frameworks is written by Erickson. Erickson focused on language, environment, speed and maturity. Caffe, being the most mature one and very fast, has a disadvantage when tuning hyperparameters. TensorFlow is regarded as hard to use directly, however has good performance and tools for help. Theano is similar as TensorFlow, in that it has good performance, however is a bit harder to learn. Both of these frameworks can be made more user friendly with a library as Keras. Keras uses the performance of either TensorFlow or Theano and can be used easily to create models with efficient code. Also Torch is given a good maturity level and good documentation. [32]

As can be seen by the many articles written about these frameworks, there is not a clear distribution as to which framework should be used when. It seems that Theano and TensorFlow, especially with help from libraries such as Keras are regarded a bit better than the others. On

Platform	Tensorflow	CNTK	Deeplearning4j	MXNet	H2O	Caffe	Theano	Torch
Release Date	2016	2016	2015	2015	2014	2014	2010	2011 (deep learning)
Core Language	C++	C++	C++	C++	Java	C++	C++	C
API	C++, Python	NDL	Java, Scala	C++, Python, R, Scala, Matlab, Javascript, Go, Julia	Java, R, Python, Scala, Javascript, web-UI	Python, Matlab	Python	Lua
Synchronization Model	Sync or async	Sync	Sync	Sync or async	Async	Sync	Async	Sync
Communication Model	Parameter server	MPI	Iterative MapReduce	Parameter server	Distributed fork-join	N/A	N/A	N/A
Multi-GPU	✓	✓	✓	✓	✓	✓	✓	✓
Multi-node	✓	✓	✓	✓	✓	✗	✗	✗
Data Parallelism	✓	✓	✓	✓	✓	✓	✓	✓
Model Parallelism	✓	N/A	✗	✓	✗	✗	✓	✓
Deep Learning Models	DBN, CNN, RNN	DBN, CNN, RNN	DBN, CNN, RNN	DBN, CNN, RNN	DBN	DBN, CNN, RNN	DBN, CNN, RNN	DBN, CNN, RNN
Programming Paradigm	Imperative	Imperative	Declarative	Both	Declarative	Declarative	Imperative	Imperative
Fault Tolerance	Checkpoint-and-recovery	Checkpoint-and-resume	Checkpoint-and-resume	Checkpoint-and-resume	N/A	N/A	Checkpoint-and-resume	Checkpoint-and-resume
Visualization	Graph (interactive), training monitoring	Graph (static)	Training monitoring	None	None	Summary Statistics	Graph (static)	Plots

Figure 12: Comparisons of eight frameworks. The comparisons were implementation based.

the other hand Caffe has some advantages, too.

References

[1] R. F. Woolson and W. R. Clarke, *Statistical methods for the analysis of biomedical data*, vol. 371. John Wiley & Sons, 2011.

[2] R. Sapsford and V. Jupp, *Data collection and analysis*. Sage, 2006.

[3] R. M. Heiberger and B. Holland, *Statistical analysis and data display*. Springer, 2004.

[4] M. J. Fisher and A. P. Marshall, "Understanding descriptive statistics," *Australian Critical Care*, vol. 22, no. 2, pp. 93 – 97, 2009.

[5] W. D. Dupont and W. D. Dupont, *Statistical modeling for biomedical researchers: a simple introduction to the analysis of complex data*. Cambridge University Press, 2009.

[6] R. Shiavi, *Introduction to applied statistical signal analysis: Guide to biomedical and electrical engineering applications*. Academic Press, 2010.

[7] N. O'Rourke and L. Hatcher, *A step-by-step approach to using SAS for factor analysis and structural equation modeling*. Sas Institute, 2013.

[8] D. Muijs, *Doing quantitative research in education with SPSS*. Sage, 2010.

[9] R. A. Muenchen, *R for SAS and SPSS users*. Springer Science & Business Media, 2011.

[10] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, SciPy Austin, TX, 2010.

[11] W. L. Martinez and A. R. Martinez, *Computational statistics handbook with MATLAB*, vol. 22. CRC press, 2007.

[12] J. Kromme, "Python & r vs. spss & sas," March 2017. https://www.r-bloggers.com/python-r-vs-spss-sas/.

[13] K. Jain, "Python vs. r (vs. sas) – which tool should i learn?," September 2017. https://www.analyticsvidhya.com/blog/2017/09/sas-vs-vs-python-tool-learn/.

[14] K. Willems, "What is the best statistical programming language? infograph," June 2014. https://www.datacamp.com/community/tutorials/statistical-language-wars-the-infograph.

[15] Support, "R vs sas vs spss – top 3 data analytics tools comparison," March 2017. http://data-flair.training/blogs/r-sas-spss-data-analytics-tools-comparison/.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[18] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[19] S.-C. Wang, *Artificial Neural Network*, pp. 81–100. Boston, MA: Springer US, 2003.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proc. 9th Python in Science Conf*, pp. 1–7, 2010.

[22] L. Rampasek and A. Goldenberg, "Tensorflow: Biology's gateway to deep learning?," *Cell systems*, vol. 2, no. 1, pp. 12–14, 2016.

[23] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint*, 2016.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093, 2014.

[26] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning software library," tech. rep., Idiap, 2002.

[27] V. Kovalev, A. Kalinovsky, and S. Kovalev, "Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?," 2016.

[28] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative study of caffe, neon, theano, and torch for deep learning," 2016.