Received September 2018; | Revised March 2019; | Accepted July 2019
DOI: xxx/xxxx

ARTICLE TYPE

Parameterizing Neural Networks for Disease Classification

Guryash Bahra[1] | **Lena Wiese**[2]

[1]Institute of Computer Science, University of Göttingen, Göttingen, Germany
[2]L3S Research Center / Knowledge Based Systems Group, Leibniz University Hannover, Hannover, Germany

Correspondence
Lena Wiese, L3S Research Center / Knowledge Based Systems Group, Leibniz University Hannover, Appelstraße 4, 30167 Hannover, Germany. Email: wiese@l3s.de

Summary

Neural networks are one option to implement decision support systems for healthcare applications. In this paper we identify optimal settings of neural networks for medical diagnoses: the study involves the application of supervised machine learning using an Artificial Neural Network (ANN) to distinguish between gout and leukemia patients. With the objective to improve the base accuracy (calculated from the initial set-up of the neural network model) several enhancements are analyzed – such as the use of hyperbolic tangent activation function instead of the sigmoid function, the use of two hidden layers instead of one as well as transforming the measurements with linear regression to obtain a smoothened data set. Another setting we study is the impact on the accuracy when using a dataset of reduced size but with higher data quality. We also discuss the tradeoff between accuracy and runtime efficiency.

KEYWORDS:
Artifical Neural Network, Disease Classification, Supervised Machine Learning, MIMIC-III

1 | INTRODUCTION

Future precision medicine can vastly profit from data analytics and machine learning to obtain a patient-specific personalized diagnosis based on various individual health indicators. Based on a body of prior experience documented in electronic medical records of other patients, machine learning techniques can form the underpinning of medical decision support systems – ideally relying on an integration of both the capabilities of storage and data analytics as for example in (Tashkandi, Wiese, & Wiese 2018). An Artificial Neural Network (ANN) is an established method to perform classification (one of the supervised learning techniques) on clinical data. In medical data analysis, machine learning is a common procedure used for classification of patients suffering from different diseases. In this paper, an ANN is used to predict and classify the diagnoses of either gout or leukemia based on uric acid measurements. However we address the topic from a more technical perspective to find out which enhanced settings (in terms of data preprocessing and neural network configuration) provide the most benefit.

1.1 | Medical Background

We concentrate on the differentiation of two diseases that both can be identified by measuring the concentration of uric acid. In particular, one of the common factors in leukemia and gout diseases is the abnormal uric acid signature in blood. Uric acid concentration in a healthy human in developed countries ranges from 3.5 mg/dl in infants to about 6 mg/dl in adults (Alvarez-Lario & Macarron-Vicente 2011; Lasko, Denny, & Levy 2013; Wilcox 1996). In patients suffering from gout or leukemia, the uric acid concentration increases more than the normal range and is therefore regularly monitored and treated. In case of gout, a combination of genetic mutations and environmental factors causes uric acid concentration to increase. This further results in formation of uric acid crystals which precipitates into joints and causes painful arthritis (Lasko et al. 2013). As for leukemia, turnover of white blood cells increases the uric acid concentration. The two diseases have different pathophysiology, which in turn combined with their treatment procedures, results in different signatures of uric acid concentrations. Therefore, in this article, supervised learning

is performed on uric acid measurements to classify the two diseases leukemia and gout. In this way, we can derive an automated recommendation for medical staff regarding the likeliness of one disease or the other.

1.2 | Machine Learning Techniques

Machine Learning is one of the top growing fields of recent times and is being utilized in various areas for many applications – such as financial trading, healthcare, recommendations, online search and many more. Machine learning involves building of models from the given dataset which can be utilized to make future predictions. This process is executed in two phases: (i) from the input dataset, calculate unknown dependencies and (ii) from those dependencies predict outcomes on so far unseen datasets.

The two common types of machine learning are supervised and unsupervised learning. Supervised learning involves use of a labelled set of input data to predict the output. In contrast, unsupervised learning involves unlabelled data. Because of this, there is no designated output such that the learning model has to identify patterns in the input data. The work in this article addresses the supervised learning problem of classification which involves categorizing the data into finite classes. More precisely, uric acid concentrations are classified into the two classes leukemia and gout.

1.3 | Objectives

The main objective of this article is to perform supervised machine learning with neural networks and assess the accuracy of the system to distinguish between the patients with either gout or leukemia. We verify and extend our previous preliminary analysis in (Bahra & Wiese 2018) by considering several advanced settings of the neural network; we also analyze the tradeoffs between these different settings of the neural network model. To achieve this objective, several tasks have to be performed and these are summarized as follows:

- identify files to be used from the Medical Information Mart for Intensive Care (MIMIC) dataset provided by Goldberger et al. (2000); Johnson et al. (2016).
- identify data (patients with either gout or leukemia and their uric acid signatures) required for the study.
- perform pre-processing on the identified data (uric acid concentrations) with linear regression.
- perform supervised learning with 3-folds cross validation on original and linear regression transformed uric acid measurements in different settings.
- improve the data quality by reducing the data set to patients with at least 3 uric acid measurements and observing the impact on the accuracy.
- assess the impact on the runtime efficiency of these enhancements.

In more detail, our study starts with identifying the tables from the MIMIC-III database. From those tables, patients with gout and leukemia diseases and their corresponding uric acid measurements are identified. The data is then cleansed, and is further used for supervised learning. The neural network for the supervised learning is designed using 1 hidden layer and sigmoid activation function. Accuracy is then calculated to measure the effectiveness of the model. Furthermore, to improve the initial accuracy of the model, a couple of enhancements are tried. These enhancements include the use of the hyperbolic tangent (tanh) activation function, 2 hidden layers and linear regression for transformation of the original uric acid measurements. Lastly, the size of the dataset is reduced in order to avoid a bias that results from singular uric acid values in the measurement sequence of a patient. It is achieved by removing the sequences with less than 3 measurements.

1.4 | Outline of the Article

After discussing the background and objectives of the article in Section 1, Section 2 surveys related work in the area. Section 3 covers the detailed description of the MIMIC database and includes tables identification as well as dataset selection and creation. Section 4 covers the supervised learning performed on our use case, introducing the basics of neural networks along with their implementation. Results and observations of supervised learning are then discussed in Section 5; this section also covers various enhanced techniques which include the use of normalization, hyperbolic tangent function as activation function, 2 hidden layers instead of 1, and linear regression for data preprocessing or transformation. Results with a reduced dataset size are presented in Section 6. The runtime behaviour of our settings is discussed in Section 7. The final Section 8 summarizes our work.

2 | RELATED WORK

Several studies in health care are based on machine learning. In this section we survey some applications of machine learning in health care in the last two decades.

Lee, Liao, and Embrechts (2000) analyzed heart disease databases with techniques like data visualization and correlation analysis to identify important features in heart disease and to build a multivariate relationship model to visualize the relationship between any two features. They describe two different approaches, discriminant analysis and neural networks classification, to identify high risk patients.

García-Gómez et al. (2004) study classification of soft tissue tumors into either benign or malignant classes based on a pattern-recognition approach. They use MR images as input data and perform machine learning with several approaches – ANN, support vector machine (SVM) and k-nearest neighbor – for the classification task. They conclude that neural networks give relatively good results compared to the other two techniques.

Joshi, Pakhomov, Pedersen, and Chute (2006) perform machine learning on electronic medical records to unambiguously expand acronyms in clinical reports. The learning is carried out with three different algorithms, Naïve Bayes, decision tree and SVM. They observe that accuracy of the model is better with all the features combined as compared to when features are used individually and in pairs, independent of the algorithm used.

Huang, McCullagh, Black, and Harper (2007) perform supervised learning on diabetes data to classify patients into diabetic and non-diabetic classes. The study also involves identification of features which give a good accuracy for the model. They used several learning algorithms – Naïve Bayes, IB1 (an instance based learning algorithm) and C4.5 (a decision tree algorithm) – to achieve the objective.

A. Nguyen, Moore, McCowan, and Courage (2007) use SVM to classify lung cancer patients into multiple classes of T (the tumor stage which describes size and position of the tumor) and N (the node stage which describes presence of spread into the lymph nodes) stages of lung cancer.

Juhola (2008) classifies otoneurological data into six classes: vestibular schwannoma, benign positional vertigo, Menière's disease, sudden deafness, traumatic vertigo and vestibular neuritis. Seven techniques – k-nearest neighbour searching, discriminant analysis, Naïve Bayesian decision rule, k-means clustering, decision trees, multilayer perceptron neural networks and Kohonen networks – are employed for this task. It is observed that linear discriminant analysis performed better than the other six techniques.

Maes, Twisk, and Johnson (2012) apply supervised learning methods, such as linear discriminant analysis, pattern recognition and factor analysis, to differentiate between myalgic encephalomyelitis, chronic fatigue syndrome and chronic fatigue.

Lasko et al. (2013) use an unsupervised learning method to identify features in gout and leukemia diseases with the use of the uric acid signatures. The work described in their paper is implemented on data extracted from Electronic Medical Records Roden et al. (2008). In their paper it is mentioned that the data is noisy, sparse and irregular, therefore, it is smoothened with the use of Gaussian process regression. On this data, deep learning is performed with the use of Sparse Autoencoders. Furthermore, the features learned from first and second layers of Sparse Autoencoders, are then utilized for a supervised learning classification task (to classify between gout and leukemia) using Logistic Regression.

Shouval et al. (2014) survey different supervised learning algorithms – namely, ANN, decision tree and SVM – on a hematopoietic stem cell transplantation database for the classification task.

Kourou, Exarchos, Exarchos, Karamouzis, and Fotiadis (2015) review supervised learning methods, namely, ANN, Bayesian Networks, SVM and decision trees for prognosis of cancer and its prediction. Their paper even highlights the case studies used for machine learning tools to predict cancer susceptibility, cancer recurrence and cancer survival.

Beaulieu-Jones and Greene (2016) develop a semi-supervised learning method and use it to improve survival predictions of patients suffering from ALS. Semi-supervised learning results from a combination of supervised and unsupervised learning. The method for semi-supervised learning is developed using Denoising Autoencoders (an unsupervised learning approach) for phenotype stratification, in combination with Random Forests (as supervised learning) for classification. In their paper, unsupervised learning is performed to reduce the amount of features of the input dataset, which is then followed by performing supervised learning on this reduced dataset.

Weng, Reps, Kai, Garibaldi, and Qureshi (2017) use supervised learning for cardiovascular risk prediction. To achieve this, the following algorithms are used: random forest, logistic regression, gradient boosting machines and neural networks. It is concluded in their paper that neural networks perform better than the rest.

It can be seen from this survey that classification of diseases is a popular use case and ANN is a widely used technique for this classification task. In contrast to the surveyed approaches, in this paper we apply supervised learning on a dataset different from the ones used before. In addition we study the effect of several enhancements (in particular of linear regression for data smoothening). Moreover, we assess the runtime behaviour of the different settings.

	# unique admissions	# admissions Leukemia	# admissions Gout
Initial admissions	2, 837	618	2, 219
Admissions after eliminating patients with both diagnoses	2, 773	584	2, 189
Admissions after eliminating patients with no uric acid measurements	1, 076	398	678
Final admissions after eliminating admissions with no uric acid measurements	640	306	334

TABLE 1 Selection of hospital admissions for analysis

3 | DATA SET

MIMIC-III is the third iteration of a large clinical database called Medical Information Mart for Intensive Care as reported byGoldberger et al. (2000); Johnson et al. (2016). It comprises of the medical data of patients admitted to critical care units, Coronary Care Unit (CCU), Cardiac Surgery Recovery Unit (CSRU), Medical Intensive Care Unit (MICU), Neonatal Intensive Care Unit (NICU), Surgical Intensive Care Unit (SICU) and Trauma Surgical Intensive Care Unit (TSICU), at the hospital Beth Israel Deaconess Medical Center in Boston. According to Goldberger et al. (2000), the current version of the database is 1.4 as of September 4, 2016, and consists of health-related records of de-identified 46,520 subjects out of which 38,645 are adults and 7,875 are neonates. The patients are de-identified according to Health Insurance Portability and Accountability Act (HIPAA) standards, which involved removal of 18 fields, such as patients' name, telephone numbers, addresses, etc., as listed in HIPAA. It also involved shifting of the dates, including date of birth, by a random offset, as directed by the HIPAA. The database not only includes the information of the vital sign measurements, medicines administered, laboratory measurements, fluid balance, imaging reports, out-of-hospital mortality but also patients' demographics, nurses' and physicians' notes, procedure and diagnostic codes, and more. Note that the MIMIC database was prepared by compiling data from two data sources, CareVue and Metavision Intensive Care Unit (ICU) databases, used at the hospital.

3.1 | Dataset Identification

There are 26 Comma Separated Values (CSV) files in the MIMIC-III data set. Out of those 26 files, the following 4 files are considered for our case study:

- `D_ICD_DIAGNOSES`: gives the disease codes (ICD-9 codes) for gout and leukemia diagnoses
- `DIAGNOSES_ICD`: identifies the hospital admissions and patients suffering from gout and leukemia
- `D_LABITEMS`: gives the ID for uric acid signatures
- `LABEVENTS`: gives the data about uric acid measurements of the patients identified with gout and leukemia

There are 78 ICD-9 codes for leukemia and 11 for gout identified from the `D_ICD_DIAGNOSES` table. Then, from the `DIAGNOSES_ICD` table, a total of 2,837 hospital admissions are identified with the above diagnoses – out of which 618 hospital admissions are for leukemia and 2,219 are for gout. These many admissions correspond to 2,259 patients or unique `SUBJECT_ID`s; in R, the *duplicated* function with logical negation operator (!) was used to find these unique `SUBJECT_ID`s. Among these subject IDs, 454 patients suffered from leukemia and 1,805 suffered from gout. Furthermore, 22 patients are common for both the diagnoses. After removing IDs of those patients, a total of 2,773 hospital admissions were identified for the patients suffering either leukemia (584 `HADM_ID`s) or gout (2,189 `HADM_ID`s) but not both. These 2,773 admissions correspond to 2,215 patients, out of which 1,783 patients suffered from gout and 432 from leukemia. Moreover, the hospital admissions are reduced from 2,773 to 1,076 by removing records with no uric acid measurements for those patients. The number of admissions further decreased to 640 as there are no uric acid observations corresponding to those admissions. And finally, these 640 unique admissions correspond to 567 unique patients, out of which 311 suffered from gout and the remaining 256 patients suffered from leukemia. Tables 1 and 2 summarize the above information.

As for the uric acid signatures, 3 IDs were identified from the `D_LABITEMS` table. Corresponding to those IDs, 19,906 observations representing uric acid measurements are identified from the `LABEVENTS` table. These 19,906 observations reduced to 7,076, as the removed observations did not correspond to the identified `SUBJECT_ID`s. Furthermore, 7,076 observations reduced to 5,665, as those observations did not correspond to the identified hospital admission IDs. The Table 3 summarizes the above information.

	# unique patients	# patients Leukemia	# patients Gout
Initial observations	2,259	454	1,805
Observations after eliminating patients with both diagnoses	2,215	432	1,783
Observations after eliminating patients with no uric acid measurements	761	273	488
Final patients after eliminating admissions with no uric acid measurements	567	256	311

TABLE 2 Selection of patients for analysis

	# uric acid measurements
Initial observations	19,906
Observations after eliminating patients with different diagnosis	7,076
Final observations corresponding to final admission IDs	5,665

TABLE 3 Selection of uric acid measurements for analysis

3.2 | Dataset Creation

The data, i.e, uric acid concentrations (from Table 3), is then arranged into 567 sequences, grouped according to the patient IDs (from Table 2). These sequences are then broken down to a size of 17 values per row, where the first two values are the label (1 for leukemia and 0 for gout) and patient ID, and the remaining 15 values are the uric acid concentrations. Note that the sizes of measurement sequences are unequal. In order to make use of all measurements in the data set, patients with more than 15 measurements occur multiple times: each sequence of 15 consecutive measurements is treated as a new sequence. On the other hand, for sequences with less than 15 measurements, value 0 is used for the remaining part of the sequence. This resulted in a total of 813 sequences. The sequences are then shuffled with the use of `sample` function provided by R Team (2014).

4 | METHOD

4.1 | Neural Networks

Neural networks are one class of several supervised learning classification techniques (as introduced in Section 1.2), and are based on the concept of perceptrons which was originally introduced by Rosenblatt (1958) and is now widely discussed in standard textbooks and surveys like Kotsiantis (2007); Nielsen (2015). In this work, a 3-layered neural network is used, where the first layer, L_1, is the input layer, the second layer, L_2, is the hidden layer, and the last layer, L_3, is the output layer. Note that the output of one layer is the input of the next layer, and there are s_l number of nodes in layer l. Figure 1 illustrates these settings.

4.1.1 | Defining the Layer Sizes

To begin, the number of nodes s_l in all the layers are defined. According to our dataset (from Section 3.2), the input size is 15, that is, the number of nodes (s_1) in layer 1 (L_1) is 15. This is because, from Section 3.2, out of 17 values per row, 15 values are the uric acid measurements. The number of output nodes is 1, as the label (leukemia or gout) per row is single-valued (from Section 3.2). The number of nodes in the hidden layer s_2 is varied in the upcoming tests in order to assess the changes (positive, negative, or no change) in the result. We chose to test $s_2 = 5$, $s_2 = 10$ and $s_2 = 25$.

4.1.2 | Forward Propagation

As the nodes are calculated starting from the layer L_1 up to layer L_3 in the network, this step is called forward propagation. The activation unit, $a_i^{(l)}$, is used to define the output of the ith unit in layer l. Therefore, for the L_1 layer, $a_i^{(1)} = x_i$ consists of one measurement value from an input sequence; as for the layers L_2 and L_3, the nodes are computational and therefore, activations are calculated as a function of an input vector $a^{(l)}$ (the activations of the previous layer), a *weights matrix* W and a *bias vector* b:

a. for layer $l = 2$: $a^{(2)} = f\left(W^{(1)} * a^{(1)}\right) + b^{(1)}$

Note that $a^{(1)}$ is an input data sequence and f is the sigmoid function (see below).

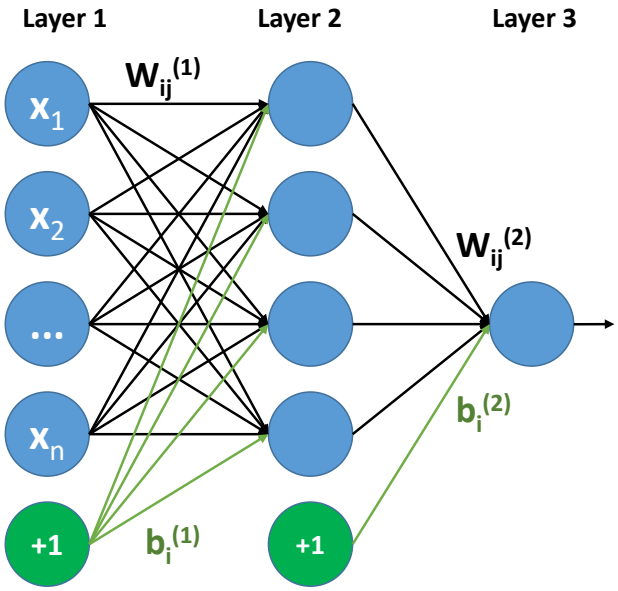

FIGURE 1 Our neural network settings: Weight matrix W and bias b are recalibrated in the learning step; weight decay parameter λ and the amount of nodes in layer 2 s_2 are varied in our tests.

b. for layer $l = 3$: $a^{(3)} = f(W^{(2)} * a^{(2)}) + b^{(2)}$; this is the final output of the neural network denoting the classification into either gout or leukemia.

Here, $W_{ij}^{(l-1)}$ is the weight or the parameter of the connection between the jth unit in layer $l-1$ and the ith unit in layer l, bias unit b_i^l corresponds to the ith unit in layer l.

4.1.3 | Initialization of Weights W and Biases b and Activation Function

The weights in matrix W are to be initialized to a value close to 0 and therefore are randomly initialized from the interval [-0.5, 0.5] (as in D. Nguyen and Widrow (1990)). The biases b are initialized to 0.

Function $f : R \to R$ is the activation function and is usually defined with sigmoid function as shown by the Equation 1; it produces an output ranging over $(0, 1)$.

$$f(z) = \frac{1}{1 + \exp(-z)} \tag{1}$$

An important identity to note is that the derivative $f'(z)$ of sigmoid function $f(z)$ (in Equation 1) is given by the Equation 2, and is used in the learning phase.

$$f'(z) = f(z)(1 - f(z)) \tag{2}$$

In order to adapt the bias vector b and the weight matrix W, we apply the squared-error cost function and minimize its value by backpropagation in the neural network.

4.1.4 | Cost Function

For a single training example (x, y) – where x is one input sequnence of 15 uric acid measurements and y is the class label for either gout or leukemia – the squared-error cost function is given in Equation 3.

$$J(W, b; x, y) = \frac{1}{2} \parallel h_{W,b}(x) - y \parallel^2 \tag{3}$$

where, $h_{W,b}(x) = a_1^{(3)}$ is the final activation of the neural network; the size of $h_{W,b}(x)$ is 1×1, as there is one output node and one training example. More generally, for a given training set $\{(x^{(1)}, y^{(1)}), .., (x^{(r)}, y^{(r)}), .., (x^{(m)}, y^{(m)})\}$ of m training examples, the cost function is given as Equation 4,

$$J(W,b) = \left[\frac{1}{m}\sum_{r=1}^{m} J\left(W,b;x^{(r)},y^{(r)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_{l-1}}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ij}^{(l)}\right)^2 \tag{4}$$

where the first term is the sum-of-squares error term averaged over all input sequnences and the second term is the *regularisation* term (or the *weight decay* term) which decreases the value of the weights and avoids overfitting. Parameter λ in Equation 4 is the *weight decay parameter*.

To minimize the cost function $J(W, b)$ as a function of weights matrix W and bias vector b, every parameter $W_{ij}^{(l)}$ and $b_i^{(l)}$ is initialized to a random value near to 0. Then an optimization algorithm is applied for minimization. We chose the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method introduced by Liu and Nocedal (1989) for the minimization of the cost function because it usually faster than a basic gradient descent algorithm. The L-BFGS algorithm is employed with the use of function minFunc (see Schmidt (2005)).

4.1.5 | Define λ and update Parameters W and b

The weight decay parameter λ (in Equation 4) needs to be defined. In the upcoming tests, the value of weight decay parameter λ is varied in order to assess the changes (positive, negative, or no change) in the result. We chose to compare three different settings for λ – namely, $\lambda = 0.0001$, $\lambda = 0.00001$, and $\lambda = 0.000001$.

The weights matrix W and the bias vector b for layers L_1 and L_2, which minimize the cost function $J(W, b)$, are recalibrated after every iteration of the L-BFGS algorithm. In other words, W and b are equal to final values of the partial derivatives of the overall cost function, $\frac{\partial}{\partial W^{(l)}} J(W, b)$ and $\frac{\partial}{\partial b^{(l)}} J(W, b)$ respectively.

4.2 | Accuracy Calculation

To calculate the accuracy of the machine learning model on the training and the testing sets, forward propagation (as described in Section 4.1.2) is performed such that for layers $l = 2, 3$: $a^{(l)} = f(W^{(l-1)} * a^{(l-1)}) + b^{(l-1)}$. Note that the activation matrix of layer l=1, $a^{(1)} = \text{data}$ (either training or testing), f is the sigmoid activation function, and W and b are calculated using the L-BFGS algorithm.

Then the activation vector of the output layer, $a^{(3)}$, is used to assign labels to the data (either training set or testing set). If the value each element in $a^{(3)}$ is greater or equal to 0.5, then, label 1 (corresponding to leukemia) is assigned to the element, else 0 (corresponding to gout) is assigned. These assigned labels then form the prediction vector of size 542×1 in case of training set and of 271×1 in case of testing set.

Each element in prediction vector is then compared with the corresponding actual label of the data. If the labels are the same, 1 is assigned to a comparison vector; if labels are not the same, then 0 is assigned to the comparison vector. Furthermore, the average is calculated for the comparison vector, which is of size 542×1 in case of training set and of 271×1 in case of testing set. The average multiplied with 100 gives the accuracy of the model in percent.

4.3 | K-Fold Cross-Validation

To create training and testing sets used for the learning, the k-fold cross validation method is employed. In k-fold cross validation, the data is randomly divided into k subsets of equal size and a single subset is referred to as fold. Of the k folds, $k - 1$ folds are combined to form the training set and the remaining fold is used as the testing set, and the accuracy is calculated for the training and the testing sets which describes the stability of the model. This is then repeated for k iterations, and for every iteration, the testing set comprises of a fold used exactly once. Moreover, to use the model for new predictions and to estimate the overall accuracy of the model, consider the classifier for which the highest accuracy is achieved for the testing set.

We applied 3-fold cross validation as well as 10-fold cross validation. As described in the previous paragraph, first the data is divided into equal subsets. Therefore, 813 sequences (from Section 3.2) are divided into 3 equal subsets as well as 10 subsets, respectively. Then, supervised learning is performed on these subsets for three iterations – each iteration either executing 3-fold or 10-fold cross validation. For each iteration, the testing set is formed with a single subset used exactly once and the remaining subsets are used as the training set. The accuracy (calculated as in Section 4.2) is reported as the average of all iterations.

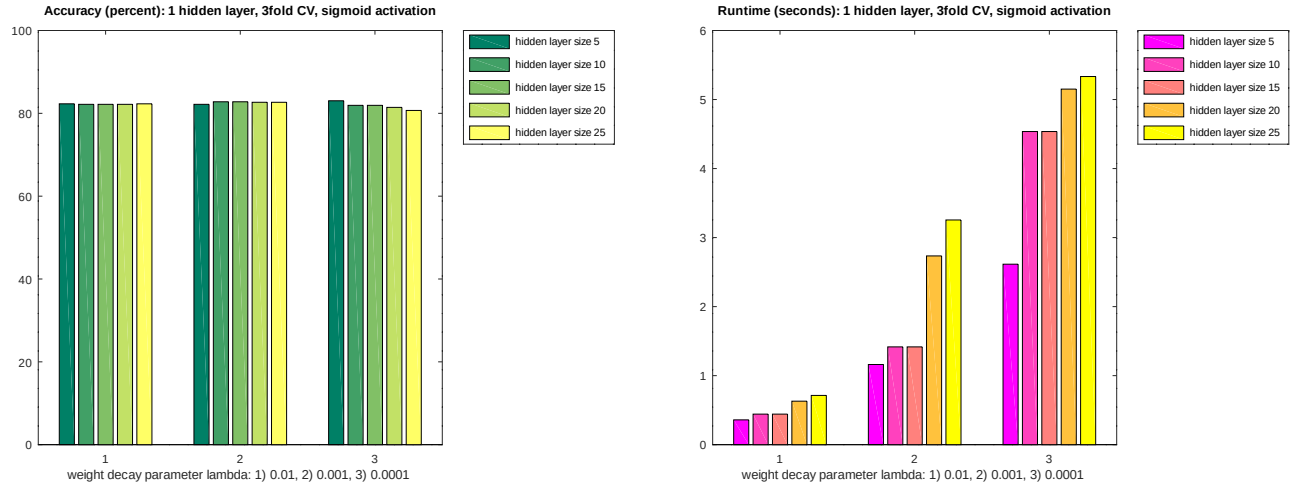

FIGURE 2 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on original dataset using neural network with 1 hidden layer.

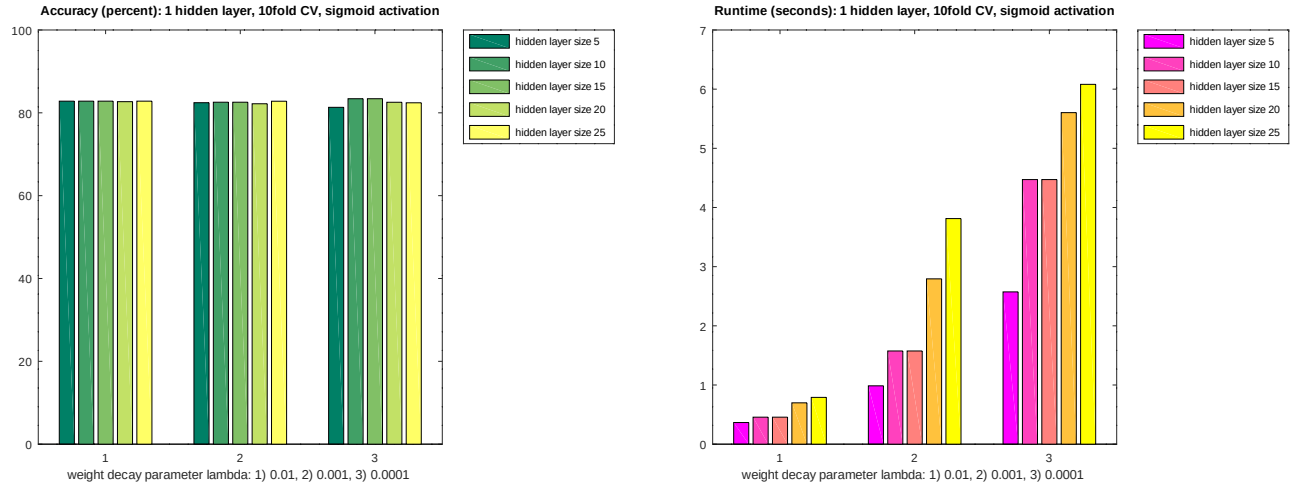

FIGURE 3 Accuracies (in %) and runtime (in seconds) of supervised learning with 10-fold cross validation on original dataset using neural network with 1 hidden layer.

5 | RESULTS ON COMPLETE DATASET

This section describes the results of supervised learning. The results represent the neural network model's ability to distinguish between patients suffering from either gout or leukemia based on abnormal uric acid measurements. The accuracies are determined for different cases, resulting from the change in the values of weight decay parameter λ and the number of hidden layer nodes s_2 (as in Section 4.1.1).

The accuracy is computed three times (iteration I1-I3) per case; this is because weights in W are randomly initialized (see Section 4.1.3) and therefore, give slightly different values for each iteration. For final accuracy of the case, these accuracies are averaged out. Accuracy is measured in percent.

5.1 | Results of Cross-Validation on Original Dataset

The results in Figure 2 are of 3-fold cross validation performed on the original dataset with one hidden layer. The following observations can be seen from Figure 2. All average test accuracies are very similar. The highest average test accuracy is 83% in case of 5 neurons in the hidden layer

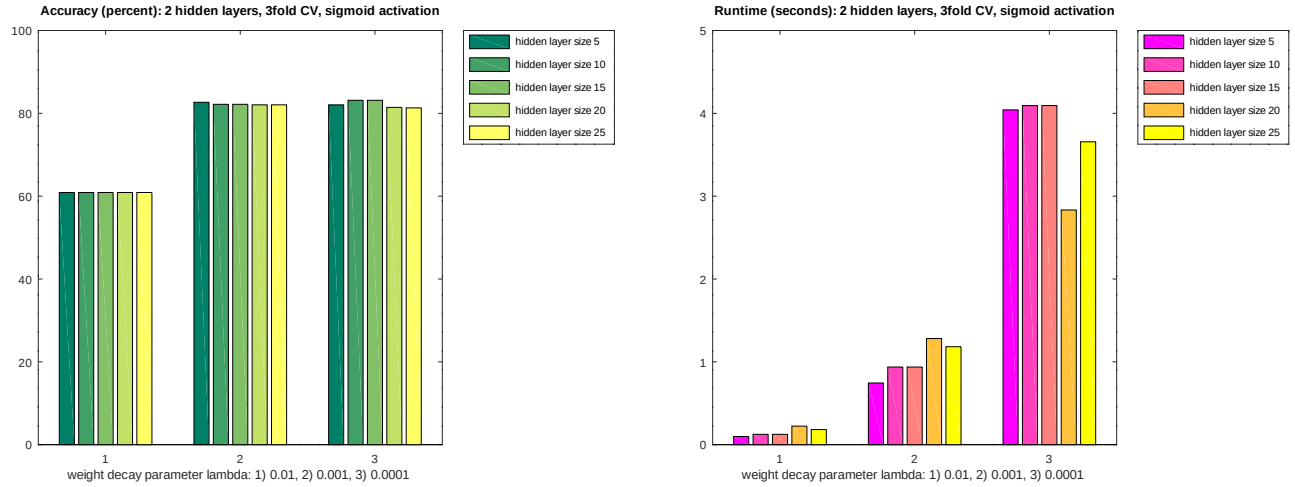

FIGURE 4 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on original dataset using neural network with 2 hidden layers and the first layer size fixed to 5.

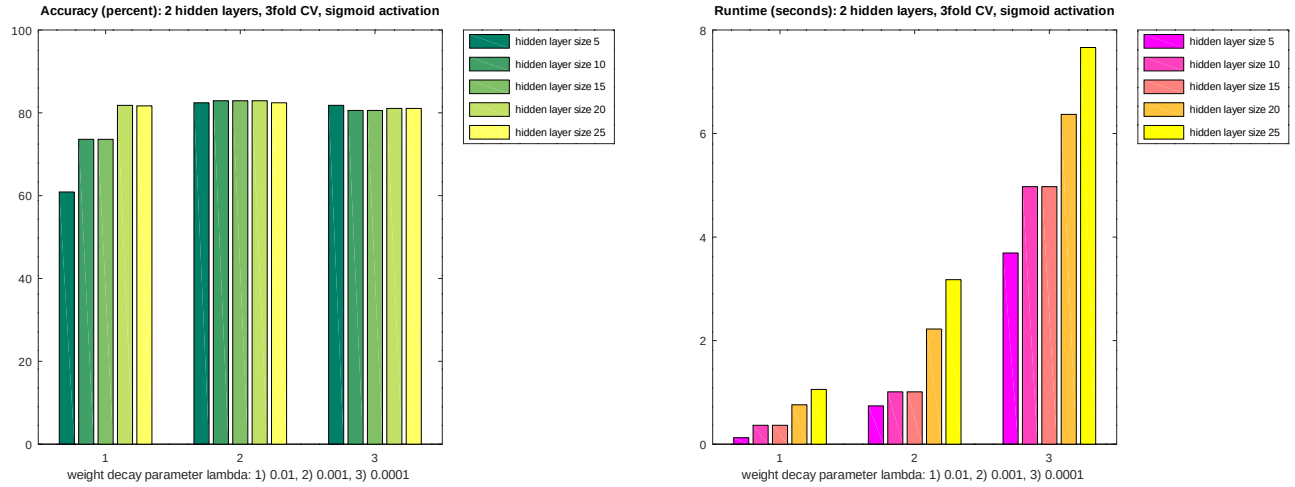

FIGURE 5 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on original dataset using neural network with 2 hidden layers and varying both layer sizes.

– that is, with the lowest number s_2 of nodes in the hidden layer – and the lowest setting for the weight decay parameter λ. However, the lower weight decay parameter incurs a much more increased runtime. The results of 10-fold cross validation in Figure 3 present a similar picture with the highest test accuracy of roughly 83% in case of 10 neurons in the hidden layer and the lowest setting for the weight decay parameter λ.

We tested different settings with 2 hidden layers. In the first case, the first hidden layer was fixed to size 5 and only the size of the second hidden layer is increased; in the second case both the first and second hidden layer sizes are increased. In the first case in Figure 4, adding the second layer reduced the accuracy for the largest weight decay parameter ($\lambda = 0.01$). For the other two weight decay parameters the accuracy remained in the same range as with one hidden layer. The second case in Figure 5, with both layer sizes increased, also shows a decrease in accuracy for the case of the largest weight decay parameter and low sizes of the first layer.

When trained with 10-fold cross validation, due to the larger training set size the decrease in accuracy did not occur (full results are not shown here due to space restrictions). In other words, the results of 10-fold cross validation with two hidden layers are comparable to Figure 3.

5.2 | Pre-processing of the Dataset using Linear Regression

The time-series data present in MIMIC-III are incomplete, inconsistent, sparse and noisy. In order to regularize the data, we applied linear regression to condition the data so as to handle irregularities in the data. We expected to smoothen the data by linear regression to obtain a general trend of each of the uric acid signatures.

Linear regression tries to model the relationship between input and output variables by fitting a linear equation to the input (or observed) data. To perform Linear regression to smoothen the data, the `lm`[1] function provided by R, is used. The `lm` function is called for a single patient ID at a time. The result of Linear regression for two different patient IDs (or sequences) can be seen in Figure 6.

FIGURE 6 Illustration of Linear regression transformation for two different sequences.

The results presented in the Figure 7 are of 3-fold cross validation performed on on the dataset transformed with linear regression before supervised learning with one hidden layer. Interestingly, for the high weight decay parameter ($\lambda = 0.01$) the linear regression of the data set

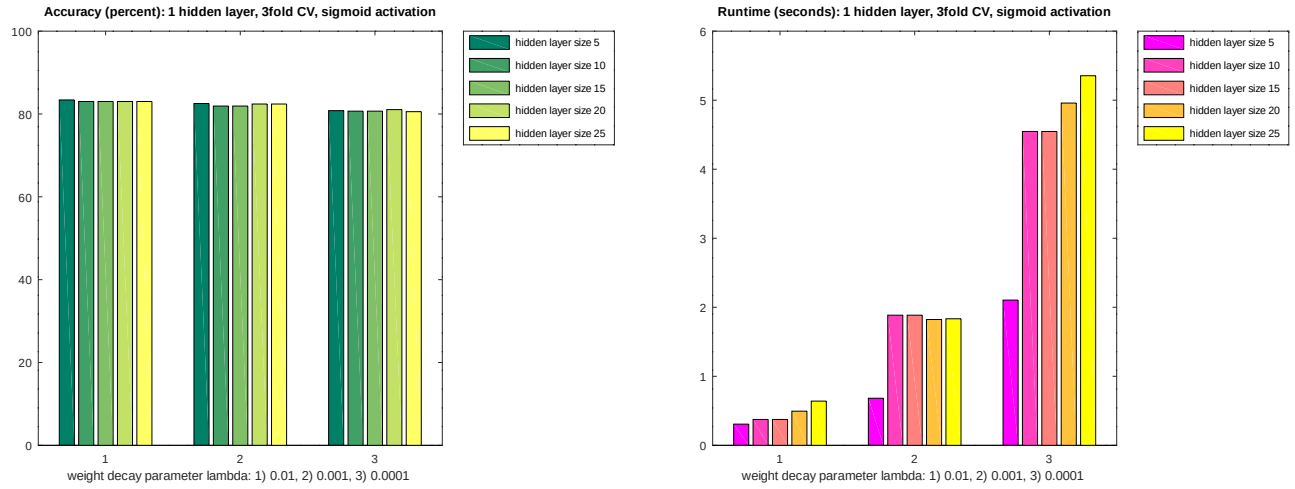

FIGURE 7 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on dataset with linear regression and using neural network with 1 hidden layer.

[1]Syntax: $\mathrm{lm}(\mathrm{x} \sim \mathrm{y})$. In this work, x is uric acid concentrations and y corresponds to time measurements.

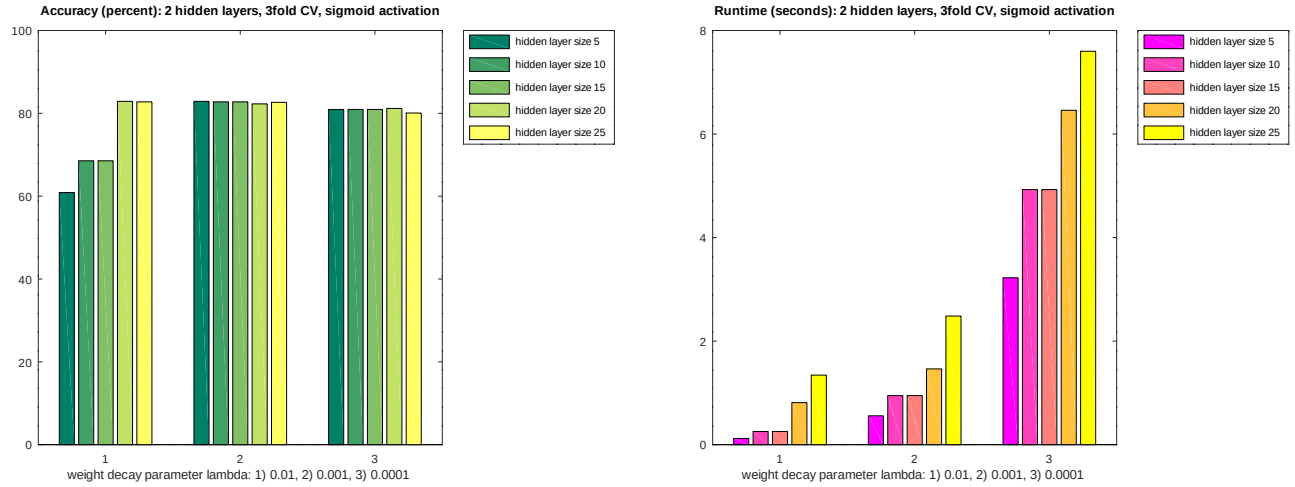

FIGURE 8 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on dataset with linear regression and using neural network with 2 hidden layers.

increased the accuracy to 83.3%). In the other cases the linear regression did not show any effect on the accuracy. The same observations were made with 10-fold cross validation.

With 2 hidden layers, the observations with linear regression resemble the observation without linear regression: for the largest weight decay parameter, the accuracy for low layer sizes is decreased when adding a second layer (shown in Figure 8). When fixing the first layer size to 5, the accuracy decreased even more in this case. In all other cases, that is, for different layer sizes as well as for all 10-fold cross validation cases, the linear regression with 2 hidden layers did not show any effect on the accuracy (results not shown here due to space restrictions).

6 | REDUCED DATASETS

In this section, sequences with less than 3 non-zero data points are removed in order to remove biases in the result due to too short uric acid measurement sequences. In other words, 3-fold cross validation is carried out on the data of the patients which have more than 2 non-zero data points (uric acid measurements) per sequence. Therefore, the number of sequences reduced to 462 (out of 813, from Section 3.2). Subsequently, the size of a single fold for cross validation was reduced accordingly, too. As in Section 5, the accuracies are determined for different settings; the accuracy is computed three times per setting and for the final accuracy, these accuracies are averaged out.

6.1 | Cross-Validation on Reduced Original Dataset

The results presented here are of cross validation performed on the reduced original dataset. The following observations can be seen from Figure 9 for one hidden layer with 3-fold cross validation. The accuracy increased when using the higher quality dataset to up to 88.4% in the case of medium weight decay parameter ($\lambda = 0.001$). A similar improvement was observed with 10-fold cross validation (not shown here due to space restrictions). In the case of 2 hidden layers these improvements also manifest for the case that both layer sizes are increased. Our main observation is that accuracies are overall better for the reduced data set (as compared to the original one).

6.2 | Cross-Validation on Reduced Transformed Dataset

We next tested the case of one hidden layer for the data set that is both reduced to retain only sequences of length at least 3 as well as transformed by linear regression. The results are shown in Figure 10. The accuracy increased to up to 89.1% for the case of $\lambda = 0.01$ with hidden layer size 20 as well as 25. This is the best setting that could be obtained; in all other cases (two hidden layers or 10-fold cross validation) this improvement could not be achieved.

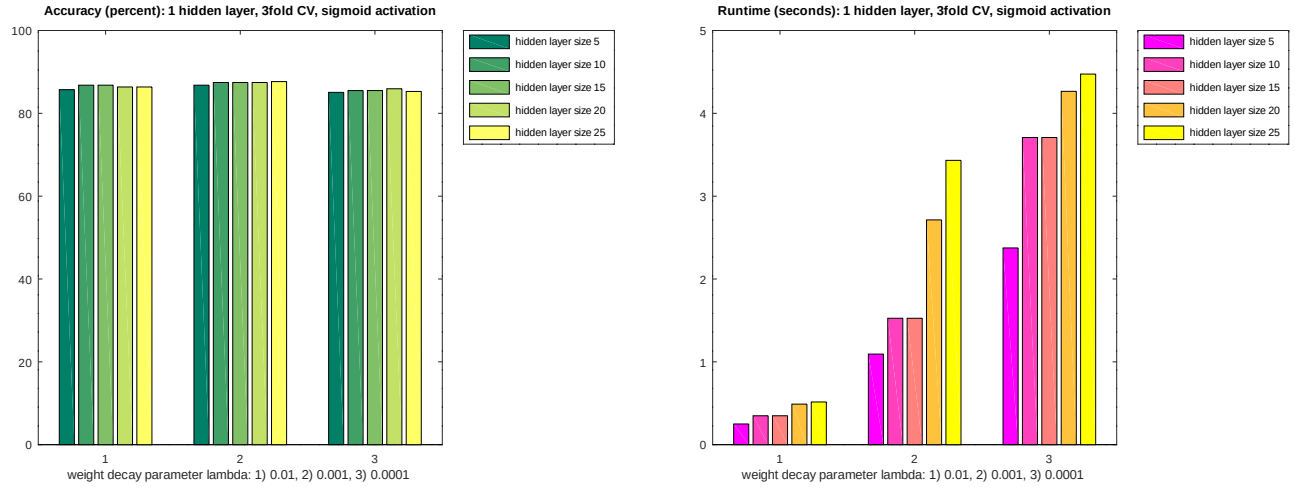

FIGURE 9 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on reduced dataset using neural network with 1 hidden layer.

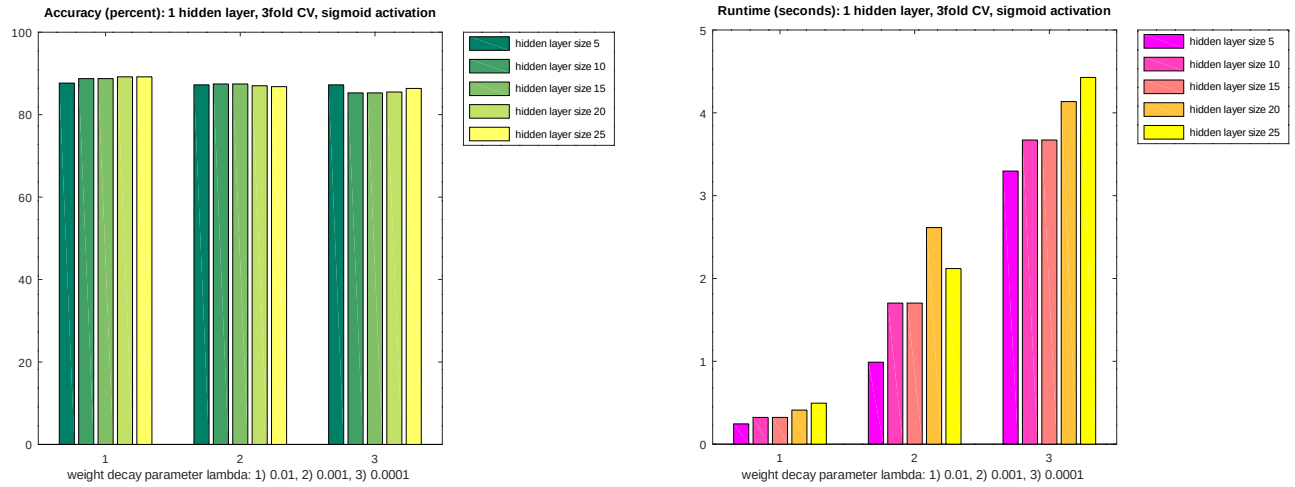

FIGURE 10 Accuracies (in %) and runtime (in seconds) of supervised learning with 3-fold cross validation on reduced dataset with linear regression and using neural network with 1 hidden layer.

The overall observation is hence that the best result can be achieved with the reduced and transformed data set lowest setting for the weight decay parameter λ. Transformation with linear regression in this case pays off with improved accuracies (as compared to the non-transformed case). In particular, again we can observe that accuracies are overall better for the reduced data set (as compared to the original one).

7 | RUNTIME COMPARISON

We implemented the steps to carry out the supervised learning presented in the previous sections in Octave Eaton (2019). Data preprocessing was done in R. We measured the runtime of the neural network model learning phases for all settings for which we reported the accuracies in the previous sections. Executions are run on a Ubuntu 16.04.2 LTS system with 8GB RAM and 1TB of hard disk. The right-hand sides of all the above figures shows the runtime in seconds.

In case of the original dataset, we can observe that the higher amount s_2 of nodes in the hidden layer(s) has a noticable impact on the runtime. In those cases where the higher amount of hidden layer nodes only gives a marginal improvement of accuracy (as in Section 5.2), the lower amount

of hidden layer nodes might be preferred in terms of runtime efficiency. Notably, the lowest value for the weight decay parameter ($\lambda = 0.001$) has a very strong impact on the runtime. With this λ value the backpropagation has difficulties finding the optimization quickly. Overall, this λ value does not pay off neither in terms of accuracy nor in terms of efficiency.

In case of the reduced dataset, we can observe that not only the accuracies are better but also the overall runtime decreased. Hence, having a smaller set of data but with a higher overall data quality can lead to better classification results.

8 | DISCUSSION AND CONCLUSION

In our experiments a neural network is designed to classify gout and leukemia patients based on their uric acid measurements. Tests showed that using more layers only improved the accuracy insignificantly. Yet, it can be observed for our use case that the lower value for the weight decay parameter leads to a runtime increase due to a more involved optimization steps and low values for this parameters should be avoided. In our settings using a high weight decay parameter and 20 hidden layers turned out to be the best setting for both accuracy and efficiency. Moreover, learning on the reduced dataset (with better data quality) performed better than on the complete dataset both in terms of accuracy as well as efficiency. Hence, regarding the tradeoff of having higher data quality with a reduced dataset size versus having overall lower data quality with a larger dataset size, we observed that a reduced dataset size provided the most benefit. It can also be observed that the using linear regression transformed data did improve the accuracy of the system best when used in combination with the reduced data set. Overall we can conclude that for our use case, this additional preprocessing step only provides a benefit on the reduced data set in terms of accuracy but not in terms of performance. To sum up, we conclude that several enhancements and settings of neural networks might not lead to optimal accuracy results. Hence, it should be carefully assessed which settings provide optimal results (both in terms of accuracy and efficiency) for the use case at hand.

In future work, our study can be extended by more features in addition to the uric acid signatures in order to improve the accuracy results. An extension to cover other diseases than gout and leukemia can also be a worthwhile topic of future work. Moreover, an in-depth comparison and combination with other related approaches (in particular, the feature learning approach in Lasko et al. (2013)) can be performed in order to assess the overall reliability of disease classification as well as quantify their runtime impact. Last but not least, we aim to investigate the impact of patient similarity Tashkandi et al. (2018) for identifying a cohort in combination with classification by neural networks.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

Alvarez-Lario, B., & Macarron-Vicente, J. (2011). Is there anything good in uric acid? *QJM: An International Journal of Medicine*, *104*, 1015-1024.

Bahra, G., & Wiese, L. (2018). Classifying leukemia and gout patients with neural networks. In *International conference on database and expert systems applications workshops* (pp. 150–160).

Beaulieu-Jones, B. K., & Greene, C. S. (2016). Semi-supervised learning of the electronic health record for phenotype stratification. *Journal of Biomedical Informatics*, *64*, 168 - 178.

Eaton, J. W. (2019). *GNU Octave version 5.1.0 manual: a high-level interactive language for numerical computations*. `https://octave.org/doc/interpreter/`.

García-Gómez, J. M., Vidal, C., Martí-Bonmatí, D. L., Galant, J., Sans, N., Robles, M., & Casacuberta, F. (2004, Mar 01). Benign /malignant classifier of soft tissue tumors using mr imaging. *Magnetic Resonance Materials in Physics, Biology and Medicine*, *16*(4), 194-201.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., … Stanley, H. E. (2000). *Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals* (Vol. 101). Circulation Electronic Pages.

Huang, Y., McCullagh, P., Black, N., & Harper, R. (2007). Feature selection and classification model construction on type 2 diabetic patients' data. *Artificial Intelligence in Medicine*, *41*(3), 251 - 262.

Johnson, A. E., Pollard, T. J., Shen, L., wei H. Lehman, L., Feng, M., Ghassemi, M., … Mark, R. G. (2016). *Mimic-iii, a freely accessible critical care database*. Scientific Data.

Joshi, M., Pakhomov, S., Pedersen, T., & Chute, C. G. (2006). A comparative study of supervised learning as applied to acronym expansion in clinical reports. *AMIA Annual Symposium Proceedings*, 399-403.

Juhola, M. (2008). On machine learning classification of otoneurological data. In *ehealth beyond the horizon - get it there* (p. 211-216). IOS Press.

Kotsiantis, S. B. (2007, October). Supervised machine learning: A review of classification techniques. In I. G. Maglogiannis, K. Karpouzis, & M. Wallace (Eds.), *Emerging artificial intelligence applications in computer engineering: Real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies* (p. 3-24). IOS Press.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, *13*, 8 - 17.

Lasko, T. A., Denny, J. C., & Levy, M. A. (2013). *Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data*. PLOS ONE 8(8).

Lee, I.-N., Liao, S.-C., & Embrechts, M. (2000). Data mining techniques applied to medical information. *Medical Informatics and the Internet in Medicine*, *25*(2), 81-102.

Liu, D. C., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, *45*(1-3), 503–528.

Maes, M., Twisk, F. N., & Johnson, C. (2012). Myalgic encephalomyelitis (me), chronic fatigue syndrome (cfs), and chronic fatigue (cf) are distinguished accurately: Results of supervised learning techniques applied on clinical and inflammatory data. *Psychiatry Research*, *200*(2), 754-760.

Nguyen, A., Moore, D., McCowan, I., & Courage, M. J. (2007). Multi-class classification of cancer stages from free-text histology reports using support vector machines. In *29th annual international conference of the ieee engineering in medicine and biology society* (p. 5140-5143).

Nguyen, D., & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Ijcnn international joint conference on neural networks* (Vol. 3).

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press. `http://neuralnetworksanddeeplearning.com/`.

Roden, D. M., Pulley, J. M., Basford, M. A., Bernard, G. R., Clayton, E. W., Balser, J. R., & Masys, D. R. (2008). *Development of a large-scale de-identified dna biobank to enable personalized medicine* (Vol. 84). Clinical Pharmacology and Therapeutics.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Schmidt, M. (2005). *minfunc: unconstrained differentiable multivariate optimization in matlab*. `https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`.

Shouval, R., Bondi, O., Mishan, H., Shimoni, A., Unger, R., & Nagler, A. (2014). Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in sct. *Bone Marrow Transplantation*, *49*, 332-337.

Tashkandi, A., Wiese, I., & Wiese, L. (2018). Efficient in-database patient similarity analysis for personalized medical decision support systems. *Big Data Research*.

Team, R. C. (2014). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. `https://www.r-project.org/`.

Weng, S. F., Reps, J., Kai, J., Garibaldi, J. M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PLOS ONE*, *12*(4), 1-14.

Wiese, L. (2015). *Advanced data management for sql, nosql, cloud and distributed databases*. DeGruyter.

Wilcox, W. (1996). *Abnormal serum uric acid levels in children* (Vol. 128). The Journal of Pediatrics.

AUTHOR BIOGRAPHY

Guryash Bahra received her B.Tech degree from Guru Tegh Bahadur Institute of Technology, India, and M.Sc. degree from University of Göttingen, Germany, in 2011 and 2018 respectively. Her research interests include topics like machine learning, data analysis, data mining and database systems.

Lena Wiese is a member of the L3S Research Center Hannover. She also leads the research group “Knowledge Engineering” (at the Institute of Computer Science, University of Goettingen). She holds a PhD and a Master degree from TU Dortmund. After her PhD she worked as a postdoctoral researcher at the National Institute of Informatics in Tokyo and as a visiting lecturer at the University of Hildesheim and the University of Salzburg. Dr. Wiese is author of the text book Wiese (2015) on Advanced Data

Management. Her research interests lie in the area of efficient and secure data management and analysis. She is an active member of the German Informatics Society (GI) and regularly acts as a reviewer for conferences and journals.

How to cite this article: Bahra G. and L. Wiese (2018), Parameterizing Neural Networks for Disease Classification, *???*, *2017;00:1–6*.