Zhou *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:148
https://doi.org/10.1186/s13638-020-01765-7

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Diabetes prediction model based on an enhanced deep neural network

Huaping Zhou, Raushan Myrzashova* and Rui Zheng

* Correspondence: rose.myrzashova@gmail.com
College of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China

Abstract

Today, diabetes is one of the most common, chronic, and, due to some complications, deadliest diseases in the world. The early detection of diabetes is very important for its timely treatment since it can stop the progression of the disease. The proposed method can help not only to predict the occurrence of diabetes in the future but also to determine the type of the disease that a person experiences. Considering that type 1 diabetes and type 2 diabetes have many differences in their treatment methods, this method will help to provide the right treatment for the patient. By transforming the task into a classification problem, our model is mainly built using the hidden layers of a deep neural network and uses dropout regularization to prevent overfitting. We tuned a number of parameters and used the binary cross-entropy loss function, which obtained a deep neural network prediction model with high accuracy. The experimental results show the effectiveness and adequacy of the proposed DLPD (Deep Learning for Predicting Diabetes) model. The best training accuracy of the diabetes type data set is 94.02174%, and the training accuracy of the Pima Indians diabetes data set is 99.4112%. Extensive experiments have been conducted on the Pima Indians diabetes and diabetic type datasets. The experimental results show the improvements of our proposed model over the state-of-the-art methods.

Keywords: Deep learning, Deep neural networks, Diabetes type prediction, Type 1 diabetes, Type 2 diabetes

1 Introduction

According to a report by the International Diabetes Federation in 2017 [1], there were 425 million diabetics in the world at the time, and it was also concluded that the number will increase to 625 million by 2045 [2]. Diabetes mellitus is a group of endocrine diseases associated with impaired glucose uptake that develops as a result of the absolute or relative insufficiency of the hormone “Insulin.” The disease is characterized by a chronic course, as well as a violation of all types of metabolism. Generally, diabetes is classified into four categories [3]: type 1 diabetes, type 2 diabetes, gestational diabetes mellitus, and specific types of diabetes due to other causes. The most common types of the disease are the following two: type 1 diabetes (T1D) and type 2 diabetes (T2D). The former is caused by the destruction of the pancreatic beta cells, resulting in insulin deficiency, while the latter is due to the ineffective transportation of insulin into cells.

Both types of the disease can lead to life-threatening complications, such as strokes, heart attacks, chronic renal failure, diabetic foot syndrome, antipathy, neuropathy, encephalopathy, hyperthyroidism, adrenal gland tumors, cirrhosis of the liver, glucagonoma, transient hyperglycemia, and many other complications. Hence, the prediction [4] and early detection [5] of diabetes is essential for all people who are predisposed to diabetes. Currently, several diseases can be diagnosed using artificial intelligence (AI) techniques, and deep neural networks [6] have achieved the best performance in classification problems. In recent years, DNNs have been used for diagnosing various diseases. In the absence of diabetes, the pancreas works fine and produces enough insulin. As soon as insulin binds to receptors on the surface of the cell, the entry for the glucose molecule into the cell will open as well. With T1D, the pancreas gradually stops producing insulin, which accordingly disrupts the process of glucose delivery to cells. T2D is not caused by the pancreas not being able to produce insulin. There is enough insulin and glucose entering cells, but the insulin receptors that allow insulin to enter into cells have lost their ability to respond to insulin. The processes through which cells absorb glucose for people who are normal, have type 1 diabetes, or have type 2 diabetes are shown in Fig. 1.

Deep learning is a new research direction in the field of machine learning [7, 8], and in recent years, it has achieved breakthrough progress in speech recognition and computer vision applications The neural network was originally developed using machine perception. The difference between it and machine perception is that it joined the multiple hidden layers [9]. As the depth of the network deepens, the feature level increases, and this enhances the expression ability of the model. The output layer can have multiple outputs, and the model can be flexibly applied to classification, regression, downscaling, and clustering. The activation function of the perceptron is sign (z), which is simple, but sign (z) has a limited processing capacity. Deep neural networks generally use the Sigmoid, Softmax, tanx, ReLU, softplus, and other activation functions and add nonlinear factors to improve the expression ability of the model.

The deep neural network [10, 11] is an extension of machine perception, and sometimes it is called the multilayer perceptron (MLP). According to the location of the different layers, the layers of the DNN can be divided into three categories: the input

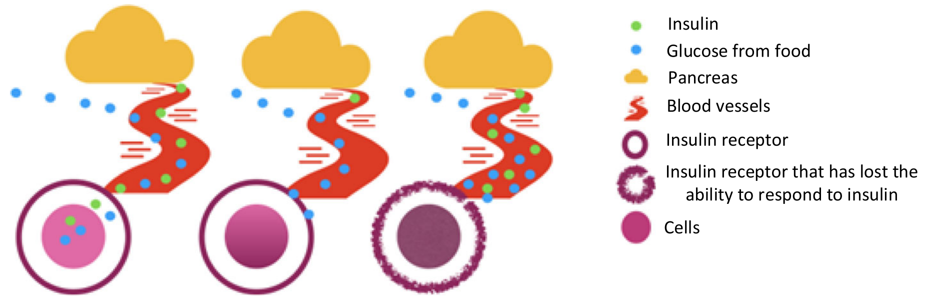

Fig. 1 Processes of glucose entering cells. This figure describes three different processes of glucose entering cells. The first process is a normal process, when a person was not diagnosed with diabetes and his pancreas produces enough insulin. The second process is when a patient has type 1 diabetes. With type 1 diabetes, the pancreas gradually stops producing insulin, which accordingly disrupts the process of glucose delivery to cells. The third process is showing the consequences of type 2 diabetes, which is not caused by the pancreas not being able to produce insulin. There is enough insulin and glucose entering the organism, but the insulin receptors that allow insulin to enter cells have lost the ability to respond to insulin. Both types are very dangerous and cause a long list of complications and illness

layer, hidden layers, and the output layer. Generally, the first layer is the input layer, the last layer is the output layer, and the middle layers are the hidden layers. According to the problems in different fields, people have developed different deep neural networks, such as the convolutional neural network (CNN) and the recurrent neural network (RNN). Due to the limitation of the parameters and the mining of the local structure, the CNN model is suitable for image recognition and speech recognition. The RNN can be viewed as a neural network that transmits over time, and its depth is the length of time. The RNN is applicable to natural language processing and handwriting recognition because the chronological order of samples' appearance is important for these areas.

The connection mode between layers is fully connected, that is to say, one neuron on layer i is connected to all neurons on layer i+1. The forward propagation algorithm in the DNN starts from the input layer. It uses the input vector x, a number of weight coefficient matrixes W and an offset vector b to make a series of linear operations and activating operations. An output layer is used to calculate the next output layer, and this proceeds all the way to the computing results for the output to the output layer. In the forward propagation process, the input information is processed by the hidden layers and then transmitted to the output layer through the hidden layers.

Without the activation function, each of the outputs is a linear function of the output from the upper layer. No matter how many layers are in the neural network, the output is only a linear combination of the input, and the effect is the same as without the hidden layers. The DNN introduces the activation function, which can effectively avoid the same effect as the single-layer linear function, improve the expression ability of the model, and make the model more differentiated. Generally, the activation function usually has the following properties: nonlinear, differentiability, and monotonicity.

A vanishing gradient occurs when the gradient of the back layer stacks to the front layer in a continuous way in the back propagation algorithm due to the chain rule. When a neural network uses the model S activation function, due to its saturation characteristic, when its input reaches a certain value, the output will not change significantly, and the derivative gradually tends to 0. When the neural network uses a gradient to update the parameters, if the continuously produced number in each layer is less than 1, the gradient will be increasingly smaller, and the error gradient for the top layer will decrease to almost zero; thus, it is unable to effectively update the parameters in the former layer.

The general rule in choosing an activation function and loss function in a DNN is that if the output of the neuron is linear, the square loss function is a right choice, and if the output neuron uses the model S activation function, the cross-entropy loss function is the right choice. The combination of the softmax activation function and the logarithmic likelihood loss is similar to the combination of the sigmoid function and the cross entropy; therefore, generally, the sigmoid activation function and cross entropy are used for the binary classification output and the softmax activation function and the logarithmic likelihood loss are used for the multiclassification output.

The DNN usually uses dropout regularization. Dropout (random inactivation) is as follows: In the training process, the training data can be divided into several groups. Then, when the DNN iterates the gradient descent using data, at a certain point, it randomly discards a part of the neuron node temporarily, and then it uses this to get rid of the hidden layer neuron network to fit a set amount of training data, and update all the weights and biases (W, b). Before the iteration of the next amount of data, the DNN model will be restored to the original fully connected model, then the neurons in the hidden layer will be randomly removed, and the weight and bias will be iteratively updated.

This work intends to present a novel deep neural network-based model (diabetes type prediction model) for diabetes prediction and the determination of the possible types of the disease in the future.

The main contributions of this paper are as follows:

(i) We propose a diabetes risk prediction model, DLPD, which can not only predict whether someone will have this disease in the future but also determine the type of disease that a person may have in the future: T1D or T2D.

(ii) We add normalization layers to the model that memorize the results from the training data and address data that it has not seen. During the training time, at each update, dropout randomly sets a fraction '*p*' of the input units to 0. The main purpose of dropout regularization is to prevent overfitting.

(iii) We choose the binary cross-entropy function as the loss function in this model. The hyper parameters are adjusted in the model. The window for accumulating past gradients is limited to a fixed size ω. The sum of the gradients is recursively defined as the attenuation average of all past square gradients. The running average at time step t then depends (as a fraction γ, similarly to the momentum term) only on the previous average and the current gradient.

(iv) The experimental results show the effectiveness and adequacy of the proposed DLPD model, the training accuracy of the diabetes type data set and the training accuracy of the Pima Indians diabetes data set are ideal.

2 Related work

Health care is one of the most important areas that societies should develop through science [12, 13] and technology [14, 15]. Deep learning methods [16–18] are powerful tools that complement traditional machine learning and allow computers to learn from data [19] so that they can come up with ways to create smarter applications [20], to process electronic health records, and to use computational vision for clinical imaging and genomics. A novel deep learning approach for the detection of type 2 diabetes was proposed by Mohebbi et al. in [21], where the authors proved that it was possible to use CGM signals to detect T2D patients. To address the challenges of using DL techniques in healthcare today, the authors focused their discussions on deep learning in computer vision, natural language processing, reinforcement learning, and generalized methods in [22]. The opportunities and obstacles for deep learning-based methods in biology and medicine have also been discussed. It can match or surpass the previous state-of-the-art methods on a diverse array of tasks in patient and disease categorization, fundamental biological studies, genomics, and treatment development [23]. Li et al. [24] demonstrated ModelHub for deep learning lifecycle management, which includes the following components: a novel model versioning system (dlv), a domain-specific language for searching the model space (DQL), and a hosted service (ModelHub) to store learned models, explore existing models, and share models. Kim et al. [25] proposed a deep network structure using SVMs with CPONs to provide adequate structural depth and robust classification accuracy. To simulate the proposed model, the Wisconsin breast cancer dataset, Pima Indians diabetes dataset, BUPA liver disorder dataset, and ionosphere dataset from the UCI Machine Learning Repository and the MNIST dataset were tested. The following accuracies were calculated for the entire test datasets: (1) Wisconsin breast cancer, 98.55%; (2) Pima Indians diabetes, 83.11%; (3) BUPA liver disorders, 77.14%; (4) ionosphere, 97.22%; and (5) MNIST, 94.93%. Yousefi et al. [26] designed a new predictive model using a stepwise hidden variable approach to predict disease complications. Hammoudeh et al. [27] presented deep learning as an effective approach for predicting hospital readmissions among diabetic patients in their research. Pham et al. [28] presented another deep learning-based approach to predict healthcare trajectories using the medical records of patients. Bae et al. [29] studied predicting the risks of type 2 diabetes using common and rare variants. Kannadasan et al. [30] proposed a new model for type 2 diabetes data classification. A deep neural network (DNN) was built using stacked autoencoders cascaded with a softmax classifier and it achieved a classification accuracy of 86.26%. Zhu et al. [31] proposed a convolutional neural network (CNN) model to forecast the future glucose levels of patients with type 1 diabetes. The model was a modified version of WaveNet, which was very useful for acoustic signal processing. Kowsher et al. [32] proposed a deep neural network and machine learning classifier using performance measures such as accuracy and precision to determine the best deep neural network algorithm. Soniya et al. [33] proposed joining a hybrid evolutionary approach with a convolutional neural network (CNN) and determined the number of layers and filters based on the application and user needs. Ramazi et al. [34] developed a wide and deep neural network and used the data from demographic information, lab tests, and wearable sensors to create the model. Alharbi et al. [35] proposed a hybrid algorithm, the GA-ELM algorithm, which optimally diagnosed type 2-diabetes patients, and classified the data set with an accuracy of 97.5% using six effective features out of the original eight features given in the dataset. The use of machine learning and deep learning algorithms for diabetes prediction, as well as comparisons of the algorithms and established models for diabetes prediction, have been accomplished by some of these related works. However, the prediction algorithms could predict only the presence or absence of the probability of having the disease in the future. In this research, we propose a novel DNN-based model for diabetes risk prediction and the determination of a specific type of the disease, T1D or T2D, which can occur in the future.

3 Methods

The inspiration for DL models is rooted in the functioning of biological nervous systems. These models are not new because their roots trace back to the introduction of the McCulloch-Pitts (MCP) model, which is considered the ancestor of the artificial neural model that has now gone mainstream because of its many practical applications and the availability of consumable technology and affordable hardware. Figure 2 represents the deep neural network architecture that we have used for predicting diabetes. This deep learning model for predicting diabetes is called DLPD. The feature vector is directly fed into the input nodes of the network. Each node generates an output with an activation function, and the linear combinations of the outputs are linked to the next hidden layers. The activation functions among different layers are different. Then, the features are retrieved and the retrieved features are concatenated to form a new

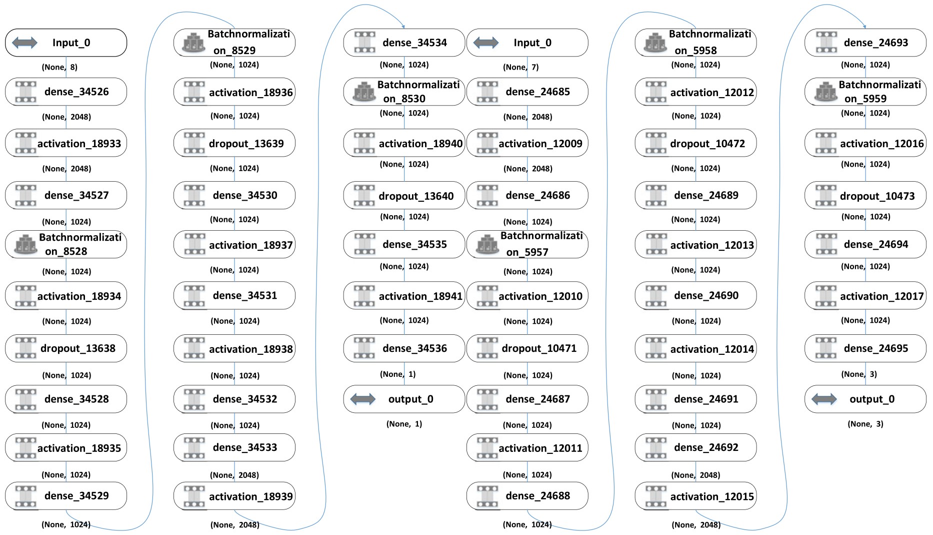

Fig. 2 Proposed model architecture. The model was created and trained on the Deep Learning Studio by Deep Cognition AI. The platform simplifies and accelerates the process of working with deep learning across popular frameworks such as TensorFlow and MXNet. By using advanced pretrained networks such as the Mask RCNN, DenseNet, MobileNet, InceptionV3, ResNet, and Xception, complete custom networks can be created in seconds with an AI Wizard on Deep Learning Studio. The feature vector is directly fed into the input nodes of the network. Each node generates an output with an activation function, and the linear combinations of the outputs are linked to the next hidden layers. The activation functions among different layers are different. Then, the features are retrieved and the retrieved features are concatenated to form a new feature vector. The output layer's activation function is the softmax. The softmax classifier uses the new feature vector to get the confidence of each relation. The dimension of the output vector is the number of classes, while the confidence of each classification equals the value of each dimension. The number of neurons in the input layer is the same as the input feature dimension, and the number of neurons in the output layer is the same as the output classes. For the neurons in the input layer, they receive a single value on their input and are sent to all of the hidden nodes. The nodes of the hidden and output layers are active, and each layer is fully interconnected. The output layer is responsible for producing and presenting the final network outputs, which are generated from the procedure performed by the neurons in the previous layers. During the training process, the input feature goes through the input nodes at the bottom of the deep learning network, where the weights are initialized with random values. After that, the weight vectors are fine-tuned in sequence. The main training goal is to minimize the loss function and maximize the accuracy function of the process

feature vector. The *softmax* classifier receives the new feature vector to get the confidence of each relation. The classifier can get the output vector. The dimension of the output vector is the number of classes while the confidence of each classification equals the value of each dimension. During the training process, the input feature goes through the input nodes at the bottom of the deep learning network, where the weights are initialized with random values. After that, the weight vectors are fine-tuned in sequence. The main training goal is to minimize the loss function and maximize the accuracy function of the process. Table 1 shows the data needed to predict diabetes, as well as the descriptions and values of these attributes. The proposed model is tested on two datasets: the Pima Indians diabetes data from the UCI repository [36] and the diabetes type dataset based on blood sugar, plasma glucose, and *HbA1c* (glycated hemoglobin) from the Data World repository [37] (Table 2).

The DLPD model consists of the following parts: (i) data pre-processing, (ii) building and training the DLPD model, (iii) adding dropout regularization to address overfitting, and (iv) hyper parameter tuning. They are described as follows (Tables 3, 4, and 5):

Table 1 Features of Pima Indians diabetes for diagnosing diabetes

No #	Attributes	Descriptions and attributes values
1	Number of times pregnant (NTP)	Numerical values
2	Plasma glucose concentration (PGC)	Numerical values
3	Diastolic blood pressure (DBP)	Numerical values in (mm Hg)
4	Triceps skin fold thickness (TSFT)	Numerical values in (mm Hg)
5	2-h serum insulin (2-HIS)	Numerical values in mm
6	Body mass index (BMI)	Numerical values in (mu U/ml)
7	Diabetes pedigree function (DPF)	Numerical values in (weight in kg/(height in m)^2)
8	Age	Numerical value

9	Diagnosis of diabetes	Yes = 1
		No = 0

3.1 Data preprocessing

Data preprocessing is necessary to prepare the diabetes type data and Pima Indians data in a manner that a deep learning model can accept. Separating the training and testing datasets ensures that the model learns only from the training data and tests its performance with the testing data. The dataset was divided into training and test data. The training data contain 70% of the total dataset, and the test and validation data contain 15% each. At first, all of the data were shuffled.

3.2 Building and training the DLPD model

The construction of a deep learning model includes three types of layers.

- The input layer is the layer to which the features of datasets will be passed. There is no computation that occurs in this layer. It serves to pass features to the hidden layers.
- The hidden layers are the layers between the input layer and the output layer. There can be various numbers of hidden layers and not only one. These layers perform the computations and pass the information to the output layer at the end.
- The output layer represents the layer of our neural network. It will give the results after training a new created model. It is responsible for producing the output variables.

The input layers represent the particular input ports in networks. The regular densely connected layers with their output dimensions use softmax activation and linear

Table 2 Features of diabetes type dataset

No #	Attributes	Type	Description and values
1	Age	Integer	Age of patient
2	BS_fast	Decimal	Blood sugar while fasting (before meal), mmol/L
3	BS_PP	Decimal	Blood sugar 90 minutes after a meal, mmol/L
4	Plasma_R	Decimal	Plasma glucose test randomly taken at any time, mmol/L
5	Plasma_F	Decimal	Plasma glucose test usually taken in the morning or 8 hours after a meal, mmol/L
6	HbA1c	Integer	mmol/mol
7	Type	String	Type 1/Type 2/Normal
8	Class	Boolean	Yes = 1/No = 0

Table 3 Data Preprocessing: separation

Dataset	Samples	Validation	Train	Split	Test
Diabetes type	1009	151	706	3	151
Pima Indians	768	76	614	4	76

activation with the weight initialization function. The activation layers apply an activation function to an output. Batch normalization layers are used to normalize the activations of the previous layer at each batch. Applied transformation maintained the mean activation close to 0 and the activation standard deviation close to 1.

3.3 Adding dropout regularization to fight overfitting

Predictive models can often face a problem known as overfitting. Overfitting occurs when the difference in accuracies is very high. The model memorizes the results from the training data and cannot be applied to data that it has not seen. To help fight overfitting in our model, we added normalization layers to our model. During the training, at each update, dropout randomly sets a fraction "p" of the input units to 0. The main purpose of dropout regularization is to prevent overfitting.

3.4 Hyper parameter tuning

Unlike machine learning models, deep learning models are literally full of hyper parameters, which determine the network structure (number of hidden units) and the variables that determine how the network is trained (learning rate). We set the number of epochs as 10 (the number of times the whole training data is shown to the network while training). The batch size is the number of sub samples given to the network after the parameter update happens, and we set the batch size to 32. The binary cross entropy was selected as the loss function. In multilabel problems, when one example can belong to multiple classes at the same time, the model tries to determine whether the example belongs to that class or not for each class. Binary cross entropy measures how far away from the true value the prediction is for each of the classes and then averages these classwise errors to obtain the final loss.

AdaDelta is an extension of AdaGrad. Instead of accumulating all past squared gradients, AdaDelta restricts the window of accumulated past gradients to some fixed size ω. Instead of inefficiently storing ω previous squared gradients, the sum of the gradients is recursively defined as a decaying average of all past squared gradients. The running average $\mathrm{E}[\mathrm{g}^2]_\mathrm{t}$ at time step t then depends (as a fraction γ similar to the Momentum term) only on the previous average and the current gradient:

Table 4 Data preprocessing: mapping

Columns	Age	BS fast	BS pp	PlasmaR	PlasmaF	Hb1A1c	Type	Class
Data type	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Category	Numeric
Input/output	InputPort	InputPort	InputPort	InputPort	InputPort	InputPort	Output	InputPort
	50	6.8	8.8	11.2	7.2	62	T1	1
	31	5.2	6.8	10.9	4.2	33	Normal	0
	41	6.3	4.2	12.2	7.8	57	T2	1

Table 5 Hyper parameter tuning

Dataset	No of epoch	Batch size	Loss function	Optimizer	Learning rate	Fuzz factor	Learning rate decay over each update
Diabetes type	10	32	Binary cross entropy	Binary cross entropy	1	1E-08	0.95
Pima Indn	10	32	AdaDelta	AdaDelta	1	1E-08	0.95

$$\mathrm{E}\left[\mathrm{g}^2{}_{\mathrm{t}} = \gamma \mathrm{E}\mathrm{g}^2\right]_{\mathrm{t}\text{-}1} + (1\text{-}\gamma)\ \mathrm{g}_{\mathrm{t}}{}^2 \tag{1}$$

Then, γ is set to a similar value as the momentum term, which is approximately 0.9. For clarity, we now rewrite our vanilla SGD update in terms of the parameter update vector Δθt:

$$\begin{aligned}\Delta\theta_{\mathrm{t}} &= \text{-}\eta\boxtimes \mathrm{g}_{\mathrm{t,i}}\\ \theta_{\mathrm{t+1}} &= \theta_{\mathrm{t}} + \Delta\theta_{\mathrm{t}}\end{aligned} \tag{2}$$

The parameter update vector of AdaDelta that was derived previously thus takes the following form:

$$\Delta\theta_{\mathrm{t}} = \text{-}\left(\eta/\left(\sqrt{G_t} + \varepsilon\right)\right)\odot \mathrm{g}_{\mathrm{t}} \tag{3}$$

Now, we simply replace the diagonal matrix Gt with the decaying average over past squared gradients $\mathrm{E}[\mathrm{g}^2]_{\mathrm{t}}$:

$$\Delta\theta_{\mathrm{t}} = \text{-}\left(\eta/\left(\sqrt{\mathrm{E}[\mathrm{g}^2]_{\mathrm{t}}} + \varepsilon\right)\right)\mathrm{g}_{\mathrm{t}} \tag{4}$$

Since the denominator is just the root mean squared (RMS) error criterion of the gradient, it is replaced with the short-hand criterion:

$$\Delta\theta_{\mathrm{t}} = \text{-}\left(\eta/\mathrm{RMS}[\mathrm{g}]_{\mathrm{t}}\right)\mathrm{g}_{\mathrm{t}} \tag{5}$$

4 Visualizing loss and accuracy

Visualization of the performance of any deep learning model is an easy way to make sense of the data being output by the model and make an informed decision about the changes that need to be made on the parameters or hyper parameters that affect the deep learning model. The plots provided in Fig. 3 show the training loss/accuracy and validation loss/accuracy. From the accuracy plot, we can see that the model achieved high training accuracy on both datasets. We can also see that the model has not yet overlearned the training dataset, showing comparable results for both datasets.

5 Results and discussion

The model was created and trained on the Deep Learning Studio using Deep Cognition AI. The platform simplifies and accelerates the process of working with deep learning across popular frameworks such as TensorFlow and MXNet. By using advanced pretrained networks such as Mask RCNN, DenseNet, MobileNet, InceptionV3, ResNet, and Xception, complete custom networks can be created in seconds with AI Wizard in Deep Learning Studio. Regarding the software, all experiments were conducted on a new platform provided by Deep Cognition AI, Deep Learning Studio.

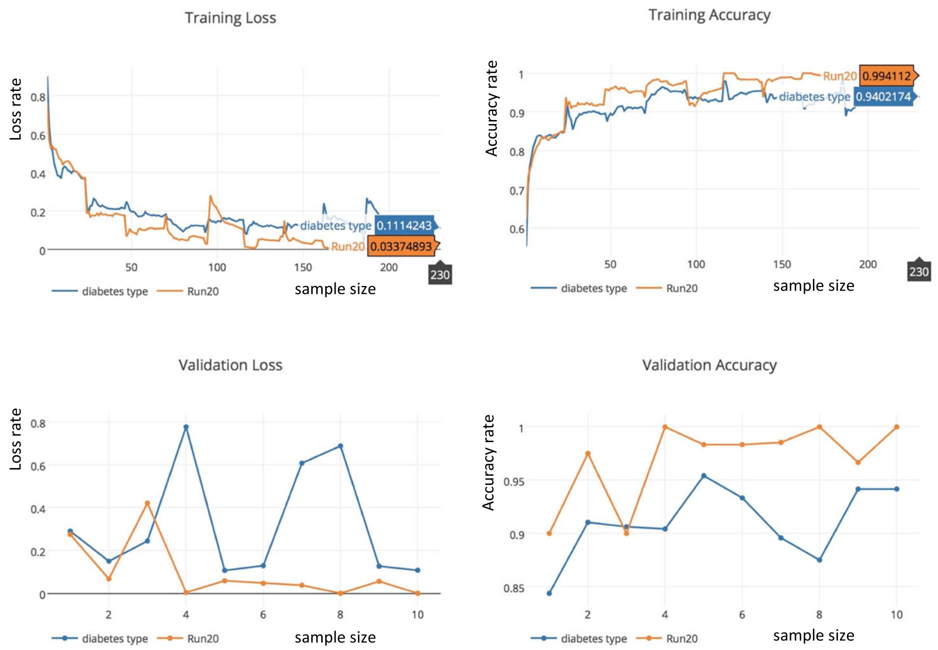

Fig. 3 Visualization of the performance of the trained model. This figure describes the results of the finished experimental work. Four plots are drawn for the following: the training accuracy, the training loss, the validation accuracy, and the validation loss. All of them describe the achieved accuracies during experiments. The main goals were to achieve high accuracy and the lowest loss of data. From these figures, we can see that experimental results of this work have reached a very good ratio between the accuracy and loss. The experiments proved that our proposed model can perform well on a different type of data. The proposed model can not only predict if a person will be diabetic in the future but also determine and predict the specific type of the disease, type 1 or type 2

Deep Learning Studio is a special tool for AI developers to build, train, and deploy their deep learning models. It is available in cloud and desktop versions. The hardware that was used for the experiments was a MacBook Pro 2018 with 128 GB of 2133 MHz LPDDR3n and a 2.3 GHz Intel Core i5. The experiments analyzed medical records data and built the deep learning model with Deep Learning Studio in this study.

The experimental results of this work have reached a very good ratio between the accuracy and loss. The experiments proved that our proposed model can perform well on different types of data. The proposed model not only can predict if a person will be diabetic in the future but also can determine and predict the specific type of the disease, type 1 or type 2.

Table 6 shows the prediction results of the proposed model on the DT dataset. From Table 7, we can conclude the final performance of the proposed model on the Pima Indians dataset is close to 100%. Table 8 represents the different performance values of proposed DNN-based model on the basis of the classified instances. The main factor is that related works presented various methods and models based on DL and ML for predicting diabetes only, but our proposed model predicts diabetes and determines the possible type of the disease that can occur in the future. Therefore, it is not entirely fair to compare the accuracy measures of the methods. A deep cognition platform can provide a WebApp/API from a trained DNN model. To organize the inference of the proposed model, a user should only enter data such as the following: age, plasma R,

Table 6 Prediction results for the diabetes type dataset

Age	BS Fast	BS pp	Plasma R	Plasma F	Hb1Ac	Type	Class	Predictions	Probabilities
22	6.8	8.8	11.2	7.2	62	Type 1	1	Type 1	0.996
24	19	6.3	7.9	3.9	40	Normal	0	Normal	1.0
41	6.3	4.2	12.2	7.8	57	Type 2	1	Type 2	1.0
44	6.8	8.2	11.6	7.4	69	Type 2	1	Type 2	0.999
22	31	6.3	7.9	3.9	40	Normal	0	Normal	1.0
60	6.7	8.7	11.6	7.4	69	Type1	1	Type2	0.925
67	6.8	4.8	13.1	9.1	58	Type2	1	Type2	0.998
23	28	7.7	11.0	6.1	36	Normal	0	Normal	0.999
35	6.7	8.7	11.6	7.4	69	Type1	1	Type1	0.911
55	5.2	6.8	10.9	4.2	33	Normal	0	Normal	1.0
26	6.8	8.2	11.6	7.4	69	Type2	1	Type2	0.921
34	5.8	4.2	11.4	8.4	53	Type2	1	Type2	0.999
27	30	7.7	11.0	6.1	36	Normal	0	Normal	0.997
26	46	5.6	10.2	5.4	32	Normal	0	Normal	1.0

plasma F, BSpp, HbA1c, and BS Fast. Submitted data will be analyzed and will give a personal prediction and probability of it as a percentage ratio.

6 Conclusion

The main key in finding the right treatment for diabetes is to detect the disease in an early stage. In the present study, a new DNN model (DTP model) was proposed for diabetes type prediction. The deep neural network was pretrained with 2 datasets with each dataset containing more than a thousand records. The number of epochs for the training phase was low, which ensures that the method can work rapidly, even on any mobile platform. The experimental results show the effectiveness and adequacy of the proposed DTP model. The best result for the diabetes type dataset was 94.02174% and

Table 7 Prediction results for the Pima Indians dataset

Preg	Glucose	Blood Pressure	Skin Thickness	Insullin	BMI	Diabetes Pedigree Function	Age	Outcome	Predictions
1	89	66	23	94	28.1	0.1669	21	0	1.0
2	108	80	28	27	27.0	0.259	52	1	0.993
7	105	90	50	87	25.3	0.305	24	0	1.0
0	135	68	42	99	42.3	0.365	24	1	0.997
6	151	62	31	120	35.5	0.692	28	0	1.0
2	81	72	15	76	30.1	0.547	25	0	0.991
0	180	78	63	14	59.4	2.42	25	1	1.0
4	173	70	14	168	29.7	0.361	33	1	1.0
1	196	76	36	249	36.5	0.875	29	1	1.0
2	112	68	22	94	34.1	0.315	26	0	0.999
0	141	84	26	0	32.4	0.433	22	0	0.998
8	151	78	32	210	42.9	0.516	36	1	0.996
3	120	70	30	135	42.9	0.452	30	0	1.0
5	187	76	27	207	43.6	1.034	53	1	0.992

Table 8 Performance of the proposed model on the basis of the classified instances

Total instances	Name of dataset	Correctly classified	Incorrectly classified	Test positive	Test negative	Test Type 1	Test Type 2
768	Pima Indians (UCI)	763	5	263	505	Not provided	
1099	Diabetes type (Data World)	1033	66	655	378	107	548

that for the Pima Indians diabetes dataset was 99.4112%. A future study will likewise focus on improving the model for determining all possible complications, including an orderly sequence in terms of the percentage of possible complications that can occur. The work can be extended and improved for automated diabetes analysis by including some other deep learning algorithms and techniques. The amount of data that the model can handle is high. In the hyper parameter tuning method, because training too many parameters can easily result in overfitting, the algorithm can also simply modify the last output layer. If the data are too different from the original dataset, the model can tune half of the layer after fine-tuning the output of the top layer.

Abbreviations

DL: Deep learning; DNN: Deep neural networks; DTP: Diabetes type prediction; T1D: Type 1 diabetes; T2D: Type 2 diabetes; DQL: Documentation query language; SVM: Support vector machine; CPON: Certified pediatric oncology nurse; CNN: Convolutional neural networks; MCP: McCulloch-Pitts

Acknowledgements

Not applicable.

Authors' contributions

All authors read and approved the final manuscript.

Funding

This work was supported by the National Natural Science Foundation of China (61703005), the Project of the Huaibei Mining Group Intelligent Property Management System and Key Research and Development Projects in Anhui Province (202004b11020029).

Availability of data and materials

The datasets are available from the UCI and Data World repositories online.

Competing interests

The authors declare that they have no competing interests.

Received: 18 March 2020 Accepted: 6 July 2020
Published online: 17 July 2020

References

1. International Diabetes Federation, *IDF Diabetes Atlas*, 8th edn. (2017)
2. G. Li, S. Peng, C. Wang, J. Niu, Y. Yuan, An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks. Tsinghua Sci. Technol. **24**(1), 86–96 (2019)
3. American Diabetes Association, Classification and diagnosis of diabetes: standards of medical care in diabetes-2018. Diabetes Care **41**, 13–27 (2018)
4. H. Liu, H. Kou, C. Yan, L. Qi, Link prediction in paper citation network to construct paper correlated graph. EURASIP J. Wirel. Commun. Netw., 233 (2019)
5. R. Miotto, F. Wang, S. Wang, X. Jiang, T. Dudley, Deep learning for healthcare: review, opportunities and challenges. Brief. Bioinform. **19**(6), 1236–1246 (2017)
6. W. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)
7. Y. Zhang, G. Cui, S. Deng, et al., Efficient query of quality correlation for service composition. IEEE Trans. Serv. Comput. (2018). https://doi.org/10.1109/TSC.2018.2830773
8. Q. Liu, G. Wang, F. Li, S. Yang, J. Wu, Preserving privacy with probabilistic indistinguishability in weighted social networks. IEEE Trans. Parall. Distrib. Syst. **28**(5), 1417–1429 (2017)
9. X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled IoT. IEEE Internet Things J. (2019). https://doi.org/10.1109/JIOT.2019.2944007
10. W. Zhong, X. Yin, X. Zhang, S. Li, W. Dou, R. Wang, L. Qi, Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment. Comput. Commun. (2020). https://doi.org/10.1016/j.comcom.2020.04.018

11. Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X.S. Shen, Energy efficient dynamic offloading in mobile edge computing for internet of things. IEEE Trans. Cloud Comput. (2019). https://doi.org/10.1109/TCC.2019.2898657
12. H. Liu, H. Kou, C. Yan, L. Qi, Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph. Complexity **2020**, 2085638, 15 pages (2020)
13. J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, F. Xia, Community-diversified influence maximization in social networks. Inf. Syst. **92**, 1–12 (2020)
14. Y. Huang, Y. Chai, Y. Liu, J. Shen, Architecture of next-generation E-commerce platform. Tsinghua Sci. Technol. **24**(1), 18–29 (2019)
15. Q. Liu, Y. Tian, J. Wu, T. Peng, and G. Wang., "Enabling verifiable and dynamic ranked search over outsourced data," IEEE Trans. Serv. Comput., 2019, doi: https://doi.org/10.1109/TSC.2019.2922177
16. X. Xu, X. Zhang, H. Gao, et al., BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing. IEEE Trans. Ind. Inform. **16**(6), 4187–4195 (2020)
17. Y. Zhang, K. Wang, Q. Wang, et al., Covering-based web service quality prediction via neighborhood-aware matrix factorization. IEEE Trans. Serv. Comput. (2019). https://doi.org/10.1109/TSC.2019.2891517
18. Y. Zhang, C. Yin, Q. Wu, et al., Location-aware deep collaborative filtering for service recommendation. IEEE Trans. Syst. Man Cybern. Syst. (2019). https://doi.org/10.1109/TSMC.2019.2931723
19. L. Qi, W. Dou, C. Hu, Y. Zhou, J. Yu, A context-aware service evaluation approach over big data for cloud applications. IEEE Trans. Cloud Comput. **8**(2), 338–348 (2020)
20. C. Zhou, A. Li, A. Hou, Z. Zhang, Z. Zhang, F. Wang, Modeling methodology for early warning of chronic heart failure based on real medical big data. Expert Syst. Appl., Published online. https://doi.org/10.1016/j.eswa.2020.113361
21. A. Mohebbi, B. Aradottir, R. Johansen, H. Bengtsson, M. Fraccaro, M. Morup, in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. A deep learning approach to adherence detection for type 2 diabetics (2017), pp. 2896–2899
22. A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, J. Dean, A guide to deep learning in healthcare. Nat. Med. **25**(1), 24–29 (2019)
23. T. Ching, S. Himmelstein, K. Beaulieu-Jones, A. Kalinin, T. Do, P. Way, S. Greene, Opportunities and obstacles for deep learning in biology and medicine. J. R. Soc. Interface **15**(141), 20170387 (2018)
24. H. Miao, A. Li, S. Davis, A. Deshpande, in *Proceedings-International Conference on Data Engineering*. Model hub: deep learning lifecycle management (2017), pp. 1393–1394
25. S. Kim, Z. Yu, R. Kil, M. Lee, Deep learning of support vector machines with class probability output networks. Neural Netw. **64**, 19–28 (2015)
26. L. Yousefi, A. Tucker, M. Al-Luhaybi, L. Saachi, R. Bellazzi, L. Chiovato, in *Proceedings - IEEE Symposium on Computer-Based Medical Systems*. Predicting disease complications using a stepwise hidden variable approach for learning dynamic Bayesian networks (2018), pp. 106–111
27. A. Hammoudeh, G. Al-Naymat, I. Ghannam, N. Obied, Predicting hospital readmission among diabetics using deep learning. Procedia Comput. Sci. **141**, 484–489 (2018)
28. T. Pham, T. Tran, D. Phung, S. Venkatesh, Predicting healthcare trajectories from medical records: a deep learning approach. J. Biomed. Inform. **69**, 218–229 (2017)
29. S. Bae, T. Park, Risk prediction of type 2 diabetes using common and rare variants. Int. J. Data Min. Bioinform. **20**(1), 77–90 (2018)
30. K. Kannadasan, D.R. Edla, V. Kuppili, Type 2 diabetes data classification using stacked autoencoders in deep neural networks. Clin. Epidemiol. Glob. Health. **7**(4), 530–535 (2018)
31. T. Zhu, K. Li, P. Herrero, J. Chen, P. Georgiou, in *KHD@ IJCAI*. A deep learning algorithm for personalized blood glucose prediction (2018), pp. 64–78
32. M. Kowsher, M.Y. Turaba, T. Sajed, et al., in *International Conference on Computer and Information Technology (ICCIT)*. Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers (2020)
33. S.P. Soniya, L. Singh, Application and need based architecture design of deep neural networks. Int. J. Pattern Recogn. Artif. Intell. **34**(13), 2052014 (24 pages) (2020)
34. R. Ramazi, C. Perndorfer, E. Soriano, et al., Multi-modal predictive models of diabetes progression. 2019.
35. A. Alharbi, M. Alghahtani, Using genetic algorithm and ELM neural networks for feature extraction and classification of type 2-diabetes mellitus. Appl. Artif. Intell. **33**(1-4), 311–328 (2019)
36. UCI world datasets repository. https://archive.ics.uci.edu/ml/. (Accessed date: 2017; Last updated in 2018)
37. DataWorld datasets repository. https://data.world/. (Accessed date: 2018)

Publisher's Note