

Special Article



Perceptron: Basic Principles of Deep Neural Networks

Eung-Hee Kim , PhD¹, and Hun-Sung Kim , MD, PhD^{2,3}

¹Department of Artificial Intelligence and Software Technology, Sunmoon University, Asan, Korea

²Department of Medical Informatics, College of Medicine, The Catholic University of Korea, Seoul, Korea

³Division of Endocrinology and Metabolism, Department of Internal Medicine, Seoul St. Mary's Hospital, College of Medicine, The Catholic University of Korea, Seoul, Korea

OPEN ACCESS

Received: Jul 14, 2021

Accepted: Jul 31, 2021

Correspondence to

Hun-Sung Kim, MD, PhD

Department of Medical Informatics, College of Medicine, The Catholic University of Korea, 222 Banpo-daero, Seocho-gu, Seoul 06591, Korea.
E-mail: 01cadiz@hanmail.net

Copyright © 2021. Korean Society of Cardiovascular Disease Prevention; Korean Society of Cardiovascular Pharmacotherapy. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ORCID iDs

Eung-Hee Kim 

<https://orcid.org/0000-0002-9533-6521>

Hun-Sung Kim 

<https://orcid.org/0000-0002-7002-7300>

Funding

This research was supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. 2019M3E5D3073104).

Conflict of Interest

The author has no financial conflicts of interest.

Author Contributions

Conceptualization: Kim HS; Data curation: Kim EH, Kim HS; Formal analysis: Kim EH, Kim

ABSTRACT

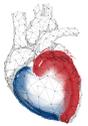
Big data, artificial intelligence, machine learning, and deep learning have received considerable attention in the medical field. Attempts to use such machine learning in areas where medical decisions are difficult or necessary are continuously being made. To date, there have been many attempts to solve problems associated with the use of machine learning by using deep learning; hence, physicians should also have basic knowledge in this regard. Deep neural networks are one of the most actively studied methods in the field of machine learning. The perceptron is one of these artificial neural network models, and it can be considered as the starting point of artificial neural network models. Perceptrons receive various inputs and produce one output. In a perceptron, various weights (ω) are given to various inputs, and as ω becomes larger, it becomes an important factor. In other words, a perceptron is an algorithm with both input and output. When an input is provided, the output is produced according to a set rule. In this paper, the decision rules of the perceptron and its basic principles are examined. The intent is to provide a wide range of physicians with an understanding of the latest machine-learning methodologies based on deep neural networks.

Keywords: Artificial intelligence; Deep learning; Machine learning; Neural networks, computer

INTRODUCTION

Machine learning improves information processing ability through learning acquired through various processing experiences of data inputs into a computer program. Previously,¹⁾ machine learning was interpreted, in an informal manner, as “a field in which a machine attempts to derive a solution with high performance, for a specific problem, by applying a learning strategy to given data.” Along with extensive research on various types of machine learning, information technology (IT) related to data production, storage, and management has been developed.²⁾³⁾ With the development of IT, machine learning in various fields, such as drug discovery,⁴⁾ medical diagnosis,⁵⁾ protein structure,⁶⁾ and functionality⁷⁾ prediction methodologies are being used successfully.

Machine learning can be largely grouped into supervised learning, unsupervised learning or reinforcement learning.⁸⁾⁹⁾ Supervised learning is used to provide the correct answer in



HS; Writing - original draft: Kim EH, Kim HS;
Writing - review & editing: Kim HS.

advance, with data already secured, and to learn it.¹⁰⁾ Good examples to illustrate this are “if your blood pressure is high, then you have hypertension” or “if you have high blood pressure and diabetes, then the incidence of cardiovascular disease increases.” Similarly, supervised learning favors this “if-then” structure. In such supervised learning, there are “classifications” that classify and divide data according to a set rule, and “regression,” which predicts the result by finding a pattern.¹¹⁾

In this article, the classification problem is discussed from the perspective of supervised learning, which is the most basic among machine-learning techniques and is most familiar to physicians. Decision trees are the most representative operating method, and the process is repeated continuously until a final diagnosis is made. Deep neural networks are one of the most actively researched and utilized methodologies in supervised learning. Machine learning is approached herein through a detailed explanation of the working principle of a “perceptron,” which is the basis of deep neural networks. In addition, this article is intended to help physicians in various fields to intuitively understand machine learning.

PRELIMINARY KNOWLEDGE ON TRAINING DATA EXPRESSED BY VECTORS

To gain insight into machine learning, it is necessary to understand the vectors representing data and basic operations on these vectors. Supervised learning refers to the process of deriving a function that provides an appropriate output when specific data are input. The data used in this process are called “training data,” and a typical form of the training data is shown in **Table 1**. **Table 1** shows the data of 8 patients and consists of the results of 2 tests (tests A and B), that each patient received, together with the resulting diagnosis (+1: cardiovascular disease, -1: normal). At this time, each test is referred to as a “feature,” and the diagnosis result is referred to as a “label.” In summary, given the feature list (result values of tests A and B), a function to predict the appropriate label (whether the patient has cardiovascular disease or is normal) is derived based on the training data expressed in **Table 1**. This process is called “supervised learning.” “Learning” is the process of finding the pattern between the features and the label existing in the training data as a function.

The training data in **Table 1** can be expressed in a 2-dimensional space, as shown in **Figure 1**. The data C (5,3), in **Figure 1**, expressed on the coordinate system can be interpreted as a movement of 5 along the test A axis and 3 along the test B axis, with respect to the origin. Therefore, **Table 1** can be described as a vector in a 2-dimensional space as follows (**Figure 1**): $\{A[1,3], -1\}, \{B[7,2], +1\}, \{C[5,3], +1\}, \{D[1,1], -1\}, \{E[6,3], +1\}, \{F[3,2], -1\}, \{G[2,1], -1\}, \{H[6,5], +1\}$.

Table 1. Typical form of the training data in supervised learning: diagnosis results, where “cardiovascular disease” and “normal” are denoted by +1 and -1, respectively

Patient	Feature		Label	
	Test A	Test B	Diagnosis	Interpretation
Patient A	1	3	-1	Normal
Patient B	7	2	+1	Cardiovascular disease
Patient C	5	3	+1	Cardiovascular disease
Patient D	1	1	-1	Normal
Patient E	6	3	+1	Cardiovascular disease
Patient F	3	2	-1	Normal
Patient G	2	1	-1	Normal
Patient H	6	5	+1	Cardiovascular disease

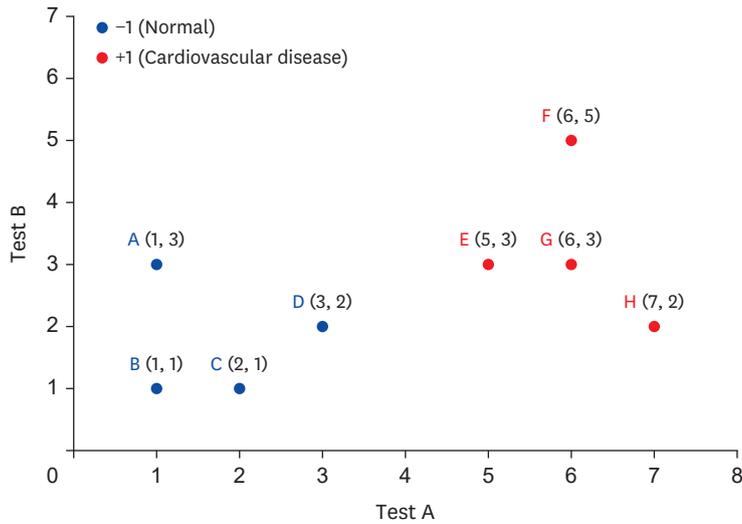


Figure 1. Training data expressed by vector in 2-dimensional space.

In general, a vector (U) described in the form of $U=[U_1, U_2, U_3, \dots, U_n]$ is a quantity having a direction and a magnitude. Each U_n is a component of U , where U is called an n -dimensional vector. Moreover, if each element of the vector U is multiplied by α , which is a positive number, a vector that is doubled by α , while maintaining the direction of the vector, is obtained. However, if α is a negative number, it is possible to obtain a vector that has a magnitude of α times the direction opposite to the existing direction. The notation $|v|$ is used to indicate the magnitude of the vector and is calculated using Equation 1. One of the advantages of examining the training data from a vector point of view is that various operators related to vectors can be employed.

$$|v| = \sqrt{\sum_{i=1}^n (v_i)^2} \quad (1)$$

BASIC OPERATORS REQUIRED TO UNDERSTAND PERCEPTRONS

In this section, the addition, subtraction, and dot product of the vectors are briefly discussed. Addition, subtraction, and dot product for given 2-dimensional vectors U and U' are defined as shown in Table 2. As can be seen in Table 2, the result of addition and subtraction between 2 n -dimensional vectors is another n -dimensional vector, whereas the result of the dot product is a single scalar value. In addition and subtraction, where the result of the operation is a new vector, as shown in Figure 2, the meaning of the operation can be understood and

Table 2. Definitions of “addition,” “subtraction,” and “dot product” of vectors

Operation	Definition	Example
Addition	$v+u=\langle v_1+u_1, v_2+u_2, \dots, v_n+u_n \rangle$	$\langle 2, 5 \rangle + \langle 1, 1 \rangle = \langle 3, 6 \rangle$
Subtraction	$v-u=\langle v_1-u_1, v_2-u_2, \dots, v_n-u_n \rangle$	$\langle 2, 5 \rangle - \langle 1, 1 \rangle = \langle 1, 4 \rangle$
Dot product	$\vec{v} \cdot \vec{u} = \sum_{i=1}^n (v_i \times u_i)$	$\langle 2, 5 \rangle \cdot \langle 1, 1 \rangle = 2 \times 1 + 5 \times 1 = 7$

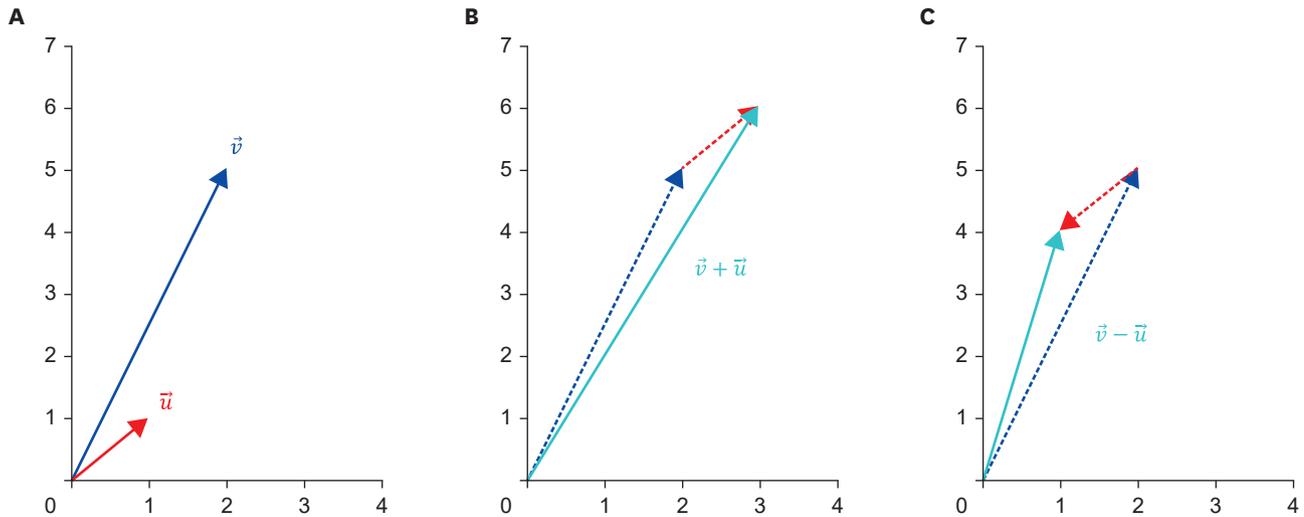
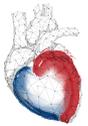


Figure 2. Understanding of the meaning of the operation as a physical movement. (A) Two dimensional sample vectors \vec{v} and \vec{u} , (B) geometric representation of vector addition and (C) geometric representation of vector subtraction.

interpreted as a physical movement. Considering the addition of 2 vectors “U+U” as an example, it is interpreted as the direction and magnitude of moving along vector U from the origin and then additionally moving along U (the starting point of the vector is not necessarily the origin; however, it is assumed so for convenience).

However, because the dot product of a vector is the result of a single number (scalar) operation, there is a limit to understanding its meaning through movement. Equation 2 defines the dot product of a vector as follows:

$$v \cdot u = |v| \times |u| \times \cos\theta \quad (2)$$

where θ is the difference in direction between U and U’.

Equation 2 can be interpreted as reflecting the difference θ in the directions between vectors after multiplying the magnitudes of the vectors. Because the cosine graph has a value of 1 when θ is 0° ($\cos 0^\circ = 1$), if the directions of the 2 vectors are the same, the dot product has the same result as multiplying the magnitudes of the 2 vectors. However, because the directions of the 2 vectors are different, the value of the vector dot product decreases. Furthermore, if it is an obtuse angle, a negative sign is used. As the magnitude of the vector cannot be negative, it can be recognized that the sign of the resulting value of the dot product operation, on the 2 vectors, is ultimately determined by the angle θ that formed between the 2 vectors.

PERCEPTRON AND THE DECISION RULE

The perceptron is a machine-learning methodology that can be regarded as the progenitor of deep neural networks.¹²⁾¹³⁾ The perceptron utilizes 1-dimensional data (i.e., points), 2-dimensional data (lines), 3-dimensional data (planes), and 4-dimensional data (hyperplanes). The perceptron is a representative linear classifier model that bisects the space in which the training data exist in this manner.¹⁴⁾

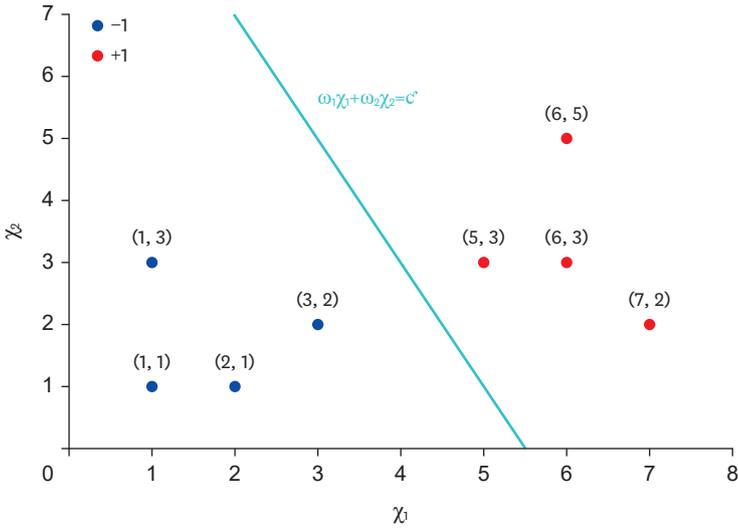


Figure 3. Model that classifies learning data in a 2-dimensional space using straight lines.

As shown in Figure 3, the perceptron is a model that classifies the learning data existing in a 2-dimensional space using a straight line. One straight line, in the 2-dimensional space, is in the form of $y = \alpha\chi + \beta$ using the slope α and the intercept β , and the point (χ_0, y_0) through which the straight line passes and the slope α based “ $y - y_0 = \alpha(\chi - \chi_0)$ ” can be expressed in various ways. In this section, the most common (standard) form, $\alpha\chi + \beta y = c$, is used for ease of explanation. At this time, because the constants α and β are weights, multiplied by the values of χ and y , α is substituted with ω_1 and β with ω_2 . Furthermore, because the variables χ and y are arbitrarily assigned simple variable names, if χ_1 and χ_2 are replaced, the straight line in Figure 3 can be expressed as $\omega_1\chi_1 + \omega_2\chi_2 = c$.

Among the training data shown in Figure 3, the 4 blue dots belonging to the normal class are the points where the result of the operation is smaller than c when substituted into $\omega_1\chi_1 + \omega_2\chi_2$. In addition, the 4 red dots can be interpreted as the points where the operation result is larger than c . In other words, the perceptron shown in Figure 3 can be regarded as a model that divides a given space into -1 and +1 based on the straight line $\omega_1\chi_1 + \omega_2\chi_2 = c$. This is described in more detail as follows.

$$\begin{aligned} &\text{If } \omega_1\chi_1 + \omega_2\chi_2 < c, \text{ then “-1”} \\ &\text{If } \omega_1\chi_1 + \omega_2\chi_2 > c, \text{ then “+1”} \end{aligned} \tag{3}$$

The decision rule of the perceptron expressed in Equation 3 is limited to 2-dimensional data, that is, when data are described with 2 characteristics, χ_1 and χ_2 . Therefore, if the classification rule of the perceptron is generalized and expressed in a form applicable to data existing in the n -dimensional space, it is expressed as Equation 4.

$$\begin{aligned} &\text{if } \sum_{i=1}^n \omega_i\chi_i < c, \text{ then } -1 \quad \text{if } \sum_{i=1}^n \omega_i\chi_i - c < 0, \text{ then } -1 \\ &\text{if } \sum_{i=1}^n \omega_i\chi_i > c, \text{ then } +1 \quad \text{if } \sum_{i=1}^n \omega_i\chi_i - c > 0, \text{ then } +1 \end{aligned} \tag{4}$$

In addition, if χ_{n+1} of a hypothetical (bias) having all values of “+1” and a corresponding weight ω_{n+1} (however, $\omega_{n+1}=-c$) are given, Equation 4 is transformed, as shown in Equation 5.

$$\begin{aligned} & \text{if } \sum_{i=1}^{n+1} \omega_i \chi_i < 0, && \text{then } -1 \\ & \text{if } \sum_{i=1}^{n+1} \omega_i \chi_i > 0, && \text{then } +1 \end{aligned} \tag{5}$$

Considering the decision rule of the perceptron expressed in Equation 6, it can be interpreted that the given data are classified according to the sign of $\sum_{i=1}^{n+1} \omega_i \chi_i$. The operation process and result of $\sum_{i=1}^{n+1} \omega_i \chi_i$ are the same as the dot product of 2 vectors ω and χ of the $n+1$ dimension introduced earlier. Therefore, the decision rule of the perceptron can be described in a simplified form using the dot product of a vector, as in Equation 6. At the same time, the perceptron can be interpreted as an $n+1$ -dimensional vector ω .

$$\begin{cases} \text{if } \text{sign}(\omega \cdot \chi) < 0, & \text{then } -1 \\ \text{if } \text{sign}(\omega \cdot \chi) > 0, & \text{then } +1 \end{cases} \tag{6}$$

PERCEPTRON LEARNING ALGORITHM (PLA)

Previously, the classification rules used by the perceptron, to classify the given data, have been discussed. In this section, the derivation of a perceptron that perfectly classifies the given training data (red and blue dots), similar to the straight line in **Figure 3**, is discussed. In other words, the PLA is a method of finding a perceptron ω that perfectly divides “-1” and “+1” among countless candidates of perceptrons in space.

By inputting the training data $D=\{(\chi_1, y_1), (\chi_2, y_2), (\chi_3, y_4), \dots, (\chi_k, y_k)\}$ composed of k data, the PLA learns and returns the perceptron ω through the following process (**Table 3**). Similar to other machine-learning methodologies, the PLA randomly generates perceptron ω and then updates it to learn (line 1). Each datum belonging to the training data (line 5) is classified according to the perceptron decision rule described in Equation 6, and whether it matches the actual value (line 6) is checked. If even one data point is misclassified, the perceptron ω is updated (line 8). If all data are correctly classified, the training process ends,

Table 3. Perceptron learning algorithm

Input: training data $D=\{(x_1,y_1), (x_2,y_2), \dots, (x_k,y_k)\}$
Output: perceptron ω
1: Randomly generate ω
2: $flag \leftarrow true$
3: While $flag=true$ do
4: $flag \leftarrow false$
5: For each $(x,y) \in D$ do
6: If $\text{sign}(w \cdot x) \neq \text{sign}(y)$ then
7: $flag \leftarrow true$
8: $\omega \leftarrow \omega + y \cdot x$
9: Break
10: End if
11: End for
12: End while
13: Return ω

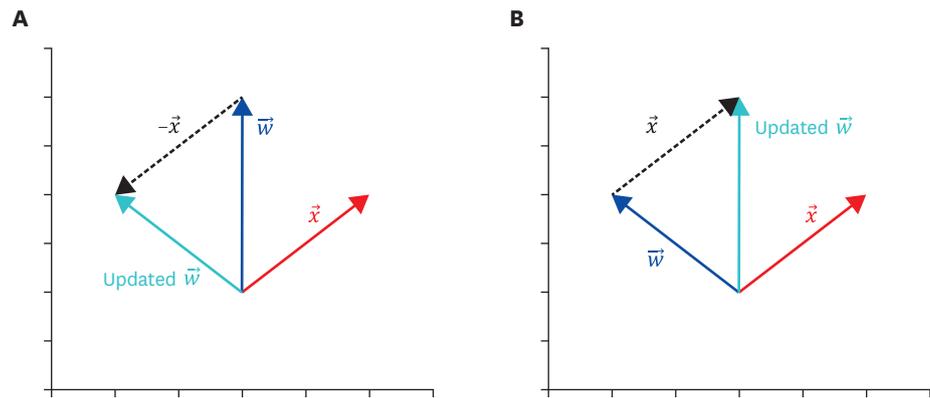
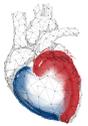


Figure 4. Perceptron update effect where blue \vec{w} : current perceptron, \vec{x} : misclassified data by \vec{w} , green \vec{w} : updated perceptron. (A) representation of update mechanism for \vec{w} that classifies \vec{x} whose label is -1 as $+1$ and (B) representation of update mechanism for \vec{w} that classifies \vec{x} whose label is $+1$ as -1 .

the update is completed, and ω is returned (line 13). In the end, the process of updating the current perceptron ω to “ $\omega + y \times \chi$ ” using the misclassified data (χ, y) is the core of learning. Understanding the effectiveness of this operation is essential. The update process is performed when “ $\text{sign}(\omega \cdot \chi) \neq \text{sign}(y)$ ”, the condition described in line 6, is satisfied.

Figure 4A shows the case where the sign of the dot product operation (resulting from the data χ and the current perceptron ω , where the correct answer is -1 ($y = -1$) is positive. As mentioned, the sign of the resulting value of the inter-vector dot product operation is determined by the angle θ between the 2 vectors. As shown in **Figure 4A**, if the sign is positive, it is an acute angle. In other words, the current perceptron misclassifies the data ω because the angle θ , formed by the 2 vectors, is an acute angle. In this case, if ω is updated according to the perceptron update rule in line 8 (**Table 3**), the angle formed by the updated ω and χ becomes an obtuse angle, as shown in **Figure 4A**, and χ is correctly classified.

Figure 4B can be interpreted in a similar way. **Figure 4B** can be considered an error because θ , which is the difference in direction between the current perceptron ω , and the data where the correct answer is $+1$ ($y = +1$), is an obtuse angle. If θ were an acute angle, the error could be avoided. In this case, if ω is updated according to the perceptron update rule $\omega \leftarrow \omega + y \times \chi$, the size of the angle between the updated ω and the data χ becomes an acute angle, confirming that the data are correctly classified.

CONCLUSION

The decision rules and learning algorithms of the perceptron, the progenitor of deep neural networks, were examined. The perceptron is an algorithm that derives a hyperplane that in turn divides a given space into 2 subspaces.¹⁵⁾ At this time, only data belonging to the same label exist in each divided space. However, data that cannot be distinguished by points, straight lines, planes, or hyperplanes can be easily encountered in the real world. To solve this nonlinear problem, a network structure composed of multiple perceptrons in multiple layers, called a “deep neural network”¹⁶⁾¹⁷⁾ is proposed. Deep neural networks, such as the input layer, hidden layer, output layer, and activation function, are composed of more diverse components than perceptrons.¹⁸⁾¹⁹⁾ However, the goal of learning is the same in that it performs the task of dividing the space into a form suitable for the given learning data.



Various deep neural networks are already being used in many fields of cardiovascular disease research, and there are many actual clinical applications.²⁰⁾²¹⁾ The decision rules and principles of the perceptron, introduced in this article, are the basis for understanding the latest machine-learning methodologies based on deep neural networks, such as convolutional neural networks²²⁾ and recurrent neural networks.²³⁾

REFERENCES

1. Mitchell TM. Machine Learning. 1st ed. New York, NY: McGraw-Hill; 1997.
2. Kim HS, Yoon KH. Lessons from use of continuous glucose monitoring systems in digital healthcare. *Endocrinol Metab* 2020;35:541-8.
[PUBMED](#) | [CROSSREF](#)
3. Lee H, Kim HS. Logistic regression and least absolute shrinkage and selection operator. *Cardiovasc Prev Pharmacother* 2020;2:142-6.
[CROSSREF](#)
4. Lee M, Kim H, Joe H, Kim HG. Multi-channel PINN: investigating scalable and transferable neural networks for drug discovery. *J Cheminform* 2019;11:46.
[PUBMED](#) | [CROSSREF](#)
5. Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Sci Rep* 2016;6:26094.
[PUBMED](#) | [CROSSREF](#)
6. Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Židek A, Nelson AW, Bridgland A, Penedones H, Petersen S, Simonyan K, Crossan S, Kohli P, Jones DT, Silver D, Kavukcuoglu K, Hassabis D. Improved protein structure prediction using potentials from deep learning. *Nature* 2020;577:706-10.
[PUBMED](#) | [CROSSREF](#)
7. Xu J, Zhang H, Zheng J, Dovoedo P, Yin Y. eCAMI: simultaneous classification and motif identification for enzyme annotation. *Bioinformatics* 2020;36:2068-75.
[PUBMED](#) | [CROSSREF](#)
8. Russell S, Norvind P. Artificial Intelligence: a Modern Approach. 3rd ed. Englewood Cliffs, NJ: Prentice Hall; 2010.
9. Herold D, Lutter D, Schachtner R, Tome AM, Schmitz G, Lang EW. Comparison of unsupervised and supervised gene selection methods. *Annu Int Conf IEEE Eng Med Biol Soc* 2008;2008:5212-5.
[PUBMED](#) | [CROSSREF](#)
10. Goudbeek M, Swingle D, Smits R. Supervised and unsupervised learning of multidimensional acoustic categories. *J Exp Psychol Hum Percept Perform* 2009;35:1913-33.
[PUBMED](#) | [CROSSREF](#)
11. Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. *J Biomed Inform* 2002;35:352-9.
[PUBMED](#) | [CROSSREF](#)
12. Lorencin I, Anđelić N, Španjol J, Car Z. Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis. *Artif Intell Med* 2020;102:101746.
[PUBMED](#) | [CROSSREF](#)
13. Castro W, Oblitas J, Santa-Cruz R, Avila-George H. Multilayer perceptron architecture optimization using parallel computing techniques. *PLoS One* 2017;12:e0189369.
[PUBMED](#) | [CROSSREF](#)
14. Metsker O, Magoev K, Yanishevskiy S, Yakovlev A, Kopanitsa G, Zvartau N. Identification of diabetes risk factors in chronic cardiovascular patients. *Stud Health Technol Inform* 2020;273:136-41.
[PUBMED](#) | [CROSSREF](#)
15. Rossi F, Conan-Guez B. Theoretical properties of projection based multilayer perceptrons with functional inputs. *Neural Process Lett* 2006;23:55-70.
[CROSSREF](#)
16. Ghosh K, Stuke A, Todorović M, Jørgensen PB, Schmidt MN, Vehtari A, Rinke P. Deep learning spectroscopy: neural networks for molecular excitation spectra. *Adv Sci (Weinh)* 2019;6:1801367.
[PUBMED](#) | [CROSSREF](#)
17. Shafi N, Bukhari F, Iqbal W, Almustafa KM, Asif M, Nawaz Z. Cleft prediction before birth using deep neural network. *Health Informatics J* 2020;26:2568-85.
[PUBMED](#) | [CROSSREF](#)



18. Kim EH, Oh SK, Pedrycz W. Design of double fuzzy clustering-driven context neural networks. *Neural Netw* 2018;104:1-14.
[PUBMED](#) | [CROSSREF](#)
19. Li Y, Zhang T. Deep neural mapping support vector machines. *Neural Netw* 2017;93:185-94.
[PUBMED](#) | [CROSSREF](#)
20. Hong S, Zhou Y, Wu M, Shang J, Wang Q, Li H, Xie J. Combining deep neural networks and engineered features for cardiac arrhythmia detection from ECG recordings. *Physiol Meas* 2019;40:054009.
[PUBMED](#) | [CROSSREF](#)
21. Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adam M, Gertych A, Tan RS. A deep convolutional neural network model to classify heartbeats. *Comput Biol Med* 2017;89:389-96.
[PUBMED](#) | [CROSSREF](#)
22. Cai L, Gao J, Zhao D. A review of the application of deep learning in medical image classification and segmentation. *Ann Transl Med* 2020;8:713.
[PUBMED](#) | [CROSSREF](#)
23. Mishra S, Banerjee M. Automatic caption generation of retinal diseases with self-trained RNN merge model. In: Chaki R, Cortesi A, Saeed K, Chaki N, editors. *Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing*, Vol. 1136. Springer: Singapore; 2020. pp.1-10.