The Potential of using Neural Networks in the diagnosis of Cardiovascular Disease

Salil Belgal

Abstract

Cardiovascular diseases are the leading cause of death globally. Early detection and prognosis of these diseases is vital, which is why scientists have turned over to new technologies such as neural networks to speed up the long manual process of identifying key features from both cardiac imaging data and cardiac time-series data that can help doctors predict and diagnose cardiovascular risk. This paper specifically looked at the effectiveness of Convolutional neural networks (CNNs) in analyzing echocardiograms and the effectiveness of Recurrent neural networks (RNNs) in analyzing electrocardiograms and patient records. This paper also explains the basics on how both types of networks are structured, trained, and evaluated. These basics help to understand the existing studies that this paper analyzes to evaluate the effectiveness of both CNNs and RNNs in the cardiovascular healthcare field. The key statistics from the studies suggest that both RNNs and CNNs have great capabilities in helping doctors find key features and predict cardiovascular risk from both cardiac imaging and time-series data.

Introduction

Cardiovascular diseases (CVDs) are the leading cause of death globally, killing an estimated 17.9 million people every year[1]. And since early detection and treatment can be the difference between life and death for a cardiovascular patient, it is necessary for scientists to implement ways to detect and treat CVDs more efficiently. This is why neural networks have emerged as an innovative technology in the diagnosis and prognosis of many diseases, including cardiovascular diseases.

Neural networks are inspired by the human brain, mimicking the way that neurons of the human brain signal to one another. Once these networks are trained for accuracy, they can classify data at very high velocity[2]. More specifically, in tasks such as image recognition, neural networks can be way more efficient than manual identification done by human experts[2]. Traditional risk predictors such as the American College of Cardiology/American Heart Association(ACC/AHA) risk model and the Framingham risk score are developed based on clinical data through medical records and patient interviews[3]. However, these models tend to be inaccurate and inefficient due to the large manpower and time required to properly analyze all of the clinical data provided by patients[3]. This again is where the fast data classification abilities of neural networks can help healthcare workers. All in all, in the realm of cardiovascular disease, these networks can help with efficiently identifying and understanding the features of cardiac data such as medical imaging data, electrocardiographs, and patient records.

Although neural networks offer a promising path in the treatment of cardiovascular disease, it is necessary to understand how they work, their capabilities, and their limitations in regards to their integration in the fight against cardiovascular disease. The purpose of this paper is to do just that by first explaining the basics of how neural networks work and how they can be trained to do pattern recognition tasks. Then, this paper will analyze existing literature and studies to determine the effectiveness of neural networks in the interpretation of electrocardiograms(ECGs) and cardiac imaging such as echocardiograms(echos). This paper will also examine existing literature to discuss the ability of neural networks to assist in cardiovascular risk prediction by analyzing patient records and also any limitations or problems involved in using these networks.

Neural Network types

There are two main types of neural networks that I will discuss in this paper, recurrent neural networks(RNNs) and convolutional neural networks(CNNs). A RNN is a type of neural network that uses time series data, which is data that is recorded over consistent intervals of time, which can include a collection of observations over time[4]. A RNN analyzes the characteristics of this data and recognizes patterns to predict the next likely scenario[5]. This is why RNNs can be effective at analyzing both patient records for risk prediction and ECG data for arrhythmia detection. On the other hand, a CNN is a type of neural network that is suited for computer vision tasks involving the processing of pixel data[6]. This makes CNNs the best neural networks for object classification and pattern recognition from images[6]. This means that CNNs can effectively be used to interpret cardiac imaging such as echocardiograms(echos). Before I examine the effectiveness of the application of RNNs and CNNs in the cardiovascular healthcare field, it is first necessary to explain the basics of how both of them work, how they are trained, and how they are tested.

Basics of CNN structure and training

The basic concept of a CNN is that it learns filters of increasing complexity layer by layer. Each feature that we want the model to detect usually has a filter, such as a filter for seeing noses which would give an indication of how strongly a nose appears in an image and also where it appears in the image[7]. A typical CNN has three layers, including the convolution layer, the pooling layer, and the fully connected layer[7]. In the Convolutional layer, the filters, also known as kernels, are moved across the pixels of an image from the top left to the bottom right. The values in the filters are also known as weights[8]. The filters perform a dot product with the source pixel, resulting in a value being stored at each point in the image[7]. The resulting map of values is called a feature map, which exists for each filter. The values on the feature maps are then taken through an activation function, which adds non-linearity to combine the feature maps produced by all the filters and also to decide whether a certain feature exists at a location[7]. The most common activation function used in the convolutional layers is known as ReLu. Typically, the pooling layer is applied after the convolutional layer. The pooling layer takes the feature maps as an input and usually selects the largest values on the feature maps, also known as max pooling which helps the network find outliers[7]. These largest values from the pooling layers can be used as inputs to subsequent layers in the network, which can be more convolutional or pooling layers depending on the task at hand. Usually, the final layer placed before the classification output of a CNN is the fully connected layer. This layer uses an activation function, most commonly Softmax, to combine elements of the final feature maps to ultimately create the final classification output[7].

This processing of the image from the input to the output is known as the forward pass of the neural network[9]. However, the CNNs are trained backwards through a process called backpropagation. I will explain the basic concept of how backpropagation works with steps below

Loss function-Backpropagation works by first calculating the loss function at the end of a forward pass, which is essentially a function that shows the error between the predictions shown by the output of the model and the actual truth[9].

Gradient calculations- The loss function is sent backwards from the output layers to the previous layers of the CNN. Then, the chain rule of calculus is used to find the gradient of the error with respect to each of the weights[9].

Gradient descent- Optimization algorithms use the gradients to update the weights in order to minimize the loss[9].

The above backpropagation process is iterative for many forward passes, meaning that backpropagation occurs every time after multiple forward passes of the same dataset in order to tune the model for accuracy[10].

Basics of RNN structure and training

One main attribute of RNNs that differentiates them from regular neural networks is their ability to memorize parts of the inputs in order to make predictions. A typical RNN has three layers: the input layer referred to as x(t), the hidden state referred to as h(t), and the output layer referred to as o(t)[11]. The t is the time step, which ranges from 1 to the number of iterations required to process a whole sequence of data. RNNs also have weights that parameterize the connections between the three layers and are also shared across time[11]. Also, RNNs use activation functions, typically at the hidden state, for the same reason as CNNs- to introduce nonlinearity. The hidden state h(t) serves as the “memory” of the RNN, which is why h(t) depends on both the current input x(t) and also the previous hidden state h(t-1)[11]. This memorizing ability is what makes RNNs suitable in analyzing time-series data.

RNNs are trained with a type of backpropagation called backpropagation through time (BPTT). BPTT is very similar to regular backpropagation performed by CNNs, as it still involves a loss function and the calculation of gradients[12]. However, in BPTT gradients are calculated backwards at each time step before the gradient descent algorithm is used to update weights[12]. Just like for CNN backpropagation, BPTT is iterative.

One of the main issues with the training of regular RNNs is the vanishing gradient problem. This problem can occur because since gradients are multiplied together, as you go backpropagate to previous time steps, the gradients can become vanishingly small[13]. When the gradients become too small, they can’t be used to update the weights and hence the model cannot improve its performance[13]. This is why for long-range dependencies, regular RNNs are replaced by types of RNNs that are structured to diminish the vanishing gradient, including LSTM(Long Short-Term Memory) and GRU (Gated Recurrent Unit)[13].

Testing of Neural Networks
To understand how classification models like neural networks are tested and evaluated, it is first important to understand the definitions of the following terms: True Positive, True Negative, False Positive, and False Negative. True Positive means that the model correctly identified that a feature is present, True Negative means that the model correctly identified that a feature is not present, False Positive means that the model incorrectly claimed that a feature is present, and False Negative means that the model incorrectly claimed that a feature is not present[14]. For example, let's say a model has to predict whether someone is sick or healthy. True positive would mean that the model correctly identified sick people as sick, True Negative would mean that the model correctly identified healthy people as not sick or healthy, False Positive would mean that the model incorrectly identified healthy people as sick, and False Negative would mean that the model incorrectly identified sick people as healthy. In this case, the positive class would be sick people while the negative class would be healthy people. Almost all methods of neural network evaluation use these values. Below is a numbered list of many of the most commonly used evaluation methods and how they are calculated.

Accuracy: Measures overall accuracy of all predictions made by the model. Calculated by doing (True Positive + True Negative)/(Total number of predictions made by model)[15].

Precision: Measures the accuracy of all positive predictions. Calculated by doing True Positive/(True Positive + False Positive)[15].

AUC(Area under the curve): AUC measures the area under the curve with the True Positive rate on the y-axis and the False Positive rate on the x-axis. This test demonstrates how well the model can distinguish between a feature being present or not, with an area of 1 showing a perfect distinguishing ability[15].

Recall: This metric measures the amount of values that are actually predicted positive out of all actually positive values. Calculated by doing True Positive/(True Positive + False Negative)[15].

R-squared value: This evaluation method measures the goodness of fit of a regression-like model by measuring the variance in the predictions explained by the dataset. A value of 1 means that the model perfectly fits the data[16].

Effectiveness of RNNs for Arrhythmia detection and risk prediction
Electrocardiograms(ECGs) record heart rate and heart rhythm, giving doctors important information such as the possibility of arrhythmia[17]. In order to determine the effectiveness of RNNs in Arrhythmia detection, I will examine and explain a study that was a part of the International Conference on Computational Intelligence and Data Science issue and is published on ScienceDirect. In their simulation, the authors trained three types of RNNs- a regular RNN, an LSTM, and a GRU- to detect arrhythmia from ECG recordings[18]. The data set

they used to train and test the RNNs was the MIT-BIH arrhythmia dataset, which contains 47 ECG records. They performed a 70:30 split on the data, meaning that 70% of the data was used for training while the remaining 30% was used for testing[18]. The main evaluation method used by the authors for the RNN models was accuracy. The results showed that for arrhythmia detection the regular RNN had an accuracy of 85.4%, the GRU had an accuracy of 82.5%, and the LSTM had an accuracy of 88.1%[18].

Along with interpreting ECGs for arrhythmia detection, RNNs, especially LSTM models, also have the ability to make cardiovascular risk predictions based on patient records. In order to evaluate RNN effectiveness in making cardiovascular risk predictions based on patient records,I will be reviewing a study published in the BioMed Central Journal where the authors used an LSTM model to predict cardiovascular health trajectories from patient data. The dataset used by the authors is the The Guideline Advantage (TGA) dataset, which contains electronic health records (EHRs) that provided the authors with 14-year longitudinal cardiovascular health measures[19]. The authors used this data to train the LSTM model with an 80:20 split, meaning 80 percent of the data was used to train the model and 20 percent was used to test the model[19]. The LSTM model was employed to predict cardiovascular health measure categories in five different submetrics, which included smoking status (SMK), A1C levels (A1C), blood pressure (BP), body mass index (BMI), and LDL cholesterol (LDL)[19]. The main evaluation method the authors used was AUC. The authors split the patients from the dataset into either the poor, intermediate, or ideal category for each submetric. The authors tested the model and found an AUC value for each category for each of the submetrics. They found out that the LSTM model proved to be accurate in predicting all five submetrics: the micro-average AUC was 0.99 for SMK prediction, 0.97 for BMI, 0.84 for BP, 0.91 for A1C, and 0.93 for LDL prediction[19].

Effectiveness of CNNs in analyzing cardiac imaging

One of the most common cardiac imaging techniques used by doctors is the use of echocardiograms(echos), which checks the structure of the heart, the surrounding blood vessels, blood flow, and the pumping chambers of the heart[20]. In order to examine the effectiveness of CNN's in analyzing echos, I will review a study published in the NPJ Digital Medicine journal. In this study, the authors trained a CNN model called EchoNet to see if the model could identify cardiac structures, estimate cardiac function, and predict phenotypes that affect cardiovascular risk[21]. They trained the model on a data set of over 2.6 million echo images all from an accredited echocardiography lab[21]. The main evaluation methods used by the authors for EchoNet were AUC and R-squared. EchoNet was able to accurately to identify the presence of many features: pacemaker leads with an AUC of 0.89, enlarged left atrium with an AUC of 0.86, left ventricular hypertrophy with an AUC of 0.75, left ventricular end systolic volume with an R-squared value of 0.74, left ventricular end diastolic volume with an R-squared value of 0.70, and ejection fraction with an R-squared value of 0.50[21]. Lastly, EchoNet was able to also predict phenotypes that affect cardiovascular risk with moderate accuracy: age with an R-squared value of 0.46, sex with an AUC of 0.88, weight with an R-squared value of 0.56, and height with an R-squared value of 0.33[21].

The Black Box problem of neural networks

The inability to see how deep learning systems such as neural networks make their decisions is known as the black block problem[22]. For example, if even after training, a neural network model continues to produce unwanted outcomes, the trainers won't be able to see why exactly the neural network is making bad decisions. In the context of neural network application in healthcare, the black box problem can serve as a big obstacle to the concept of patient trust, where patients want transparency in medical recommendations. The lack of explainability of neural networks can make it very hard for both patients and doctors to understand exactly why the model is predicting or recommending something. There is still ongoing research to try and look into this "box" to find out why neural networks make certain decisions. This ongoing research field is known as "explainable AI", where computer scientists are trying to develop algorithms to make deep learning models like neural networks more transparent and accountable[22].

Conclusion

Through the review of the existing studies, it is clear that CNNs have great capabilities in analyzing echocardiograms and RNNs have great capabilities in detecting arrhythmia from electrocardiograms and predicting sub metrics that can help doctors assess cardiovascular risk. However, the integration of these neural networks in the cardiovascular healthcare field still has its limitations because the black box problem can make it hard for patients to understand why certain predictions are being made by the model.

References

1. World Health Organization. (2021). Cardiovascular diseases. https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1
2. IBM. (n.d.). What are neural networks? https://www.ibm.com/topics/neural-networks
3. Kim, J. O., Jeong, Y.-S., Kim, J. H., Lee, J.-W., Park, D., & Kim, H.-S. (2021). Machine Learning-Based Cardiovascular Disease Prediction Model: A Cohort Study on the Korean National Health Insurance Service Health Screening Database. *Diagnostics, 11*(6), 943.
4. IBM. (n.d.). What are recurrent neural networks? https://www.ibm.com/topics/recurrent-neural-networks
5. Laskowski, N. (2021). What are recurrent neural networks? TechTarget. https://www.techtarget.com/searchenterpriseai/definition/recurrent-neural-networks#:~:text=A%20recurrent%20neural%20network%20is,predict%20the%20next%20likely%20scenario
6. Awati, R. (2023). Convolutional neural network (CNN). TechTarget. https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network?Offer=abMeterCharCount_var1#

7. Stewart, M. (2019). Simple Introduction to Convolutional Neural Networks. Towards Data Science. https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac
8. Shah, S. (2022). Convolutional Neural Network: An Overview. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/#:~:text=The%20filters%20are%20learned%20during,called%20the%20weights%20of%20CNN.&text=A%20feature%20map%20is%20a,inputs%20with%20the%20same%20weights
9. Roy, R. (2022). Neural Networks: Forward Pass and Backpropagation. Towards Data Science. https://towardsdatascience.com/neural-networks-forward-pass-and-backpropagation-be3b75a1cfcc
10. Arie, L. G. (2021). Neural Network Backpropagation made easy. Towards Data Science. https://towardsdatascience.com/neural-networks-backpropagation-by-dr-lihiur-arie-27be67d8fdce
11. Nabi, J. (2019). Recurrent Neural Networks (RNNs). Towards Data Science. https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85
12. Brownlee, J. (2020). A Gentle Introduction to Backpropagation Through Time. Machine Learning Mastery. https://machinelearningmastery.com/gentle-introduction-backpropagation-time/
13. Patil, S. (2023). Vanishing Gradient Problem in RNNs. Medium. https://medium.com/@sagarpatiler/vanishing-gradient-problem-in-rnns-d362235005c
14. Dilmegani, C. (2022). Machine Learning Accuracy: True-False Positive/Negative. AIMultiple. https://research.aimultiple.com/machine-learning-accuracy/
15. Mankad, S. (2020). A Tour of Evaluation Metrics for Machine Learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/11/a-tour-of-evaluation-metrics-for-machine-learning/
16. Kharwal, A. (2021). R2 Score in Machine Learning. The Clever Programmer. https://thecleverprogrammer.com/2021/06/22/r2-score-in-machine-learning/
17. Mayo Clinic Staff. (2022). Electrocardiogram (ECG or EKG). Mayo Clinic. https://www.mayoclinic.org/tests-procedures/ekg/about/pac-20384983
18. Singh, S., Pandey, S. K., Pawar, U., & Janghel, R. R. (2018). Classification of ECG Arrhythmia using Recurrent Neural Networks. Procedia Computer Science, 132, 1290-1297.
19. Guo, A., Beheshti, R., Khan, Y. M., Langabeer II, J. R., Foraker, R. E. (2021). Predicting cardiovascular health trajectories in time-series electronic health records with LSTM models. BMC Medical Informatics and Decision Making, 21(1), 5.
20. NHS. (2022). Echocardiogram. https://www.nhs.uk/conditions/echocardiogram/

21. Ghorbani, A., Ouyang, D., Abid, A., He, B., Chen, J. H., Harrington, R. A., ... Zou, J. Y. (2020). Deep Learning interpretation of echocardiograms. npj Digital Medicine, 3, 10.
22. Blouin, L. (2023). AI's mysterious “black box” problem, explained. University of Michigan-Dearborn. https://umdearborn.edu/news/ais-mysterious-black-box-problem-explained