Methods 198 (2022) 96–106

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Prediction of coronary heart disease based on combined reinforcement multitask progressive time-series networks

Wenqi Li [a,b], Ming Zuo [c,d], Hongjin Zhao [a], Qi Xu [c], Dehua Chen [a,*]

[a] School of Computer Science and Technology, Donghua University, Shanghai, China
[b] Artificial Intelligence Lab, China UnionPay Headquarters, Shanghai, China
[c] Glorious Sun School of Business and Management, Donghua University, Shanghai, China
[d] Ruijin Hospital, Shanghai Jiaotong University School of Medicine, Shanghai, China

ARTICLE INFO

Keywords:
Combined reinforcement multitask progressive time-series networks
Coronary heart disease prediction
Asynchronous advantage actor-critic
Soft parameter sharing
Hard parameter sharing

ABSTRACT

Coronary heart disease is the first killer of human health. At present, the most widely used approach of coronary heart disease diagnosis is coronary angiography, a surgery that could potentially cause some physical damage to the patients, together with some complications and adverse reactions. Furthermore, coronary angiography is expensive thus cannot be widely used in under development country. On the other hand, the heart color Doppler echocardiography report, blood biochemical indicators and personal information, such as gender, age and diabetes, can reflect the degree of heart damage in patients to some extent. This paper proposes a combined reinforcement multitask progressive time-series networks (CRMPTN) model to predict the grade of coronary heart disease through heart color Doppler echocardiography report, blood biochemical indicators and ten basic body information items about the patients. In this model, the first step is to perform deep reinforcement learning (DRL) pre-training through asynchronous advantage actor-critic (A3C). Training data is adopted to optimize the recurrent neural network (RNN) that parameterizes the stochastic policy. In the second step, soft parameter sharing module, hard parameter sharing module and progressive time-series networks are used to predict the status of coronary heart disease. The experimental results show that after DRL pre-training, the multiple tasks in the model interact with each other and learn together to achieve satisfactory results and outperform other state-of-the-art methods.

1. Introduction

Coronary heart disease is known as "the first killer of human health." According to the data released by the WHO in 2019, the number of people dying from cardiovascular disease each year accounts for 31% of the total number of deaths worldwide, more than any other cause of death. However, the diagnose of the cardiovascular disease is still expensive which blocks the widely application of the current technology world wide, especially in some under development area. Therefore, it is urgent to find a method that can quickly and effectively diagnose coronary heart disease at a small cost.

Nowadays, the common way to diagnose coronary heart disease is coronary angiography, an invasive surgery which is considered as the "gold standard" for the diagnosis of coronary heart disease. However, for patients, coronary angiography has the following disadvantages: (1) Cause arrhythmia. Ventricular fibrillation is the most serious complication, which can be caused by many related factors, among which the main contributors are: the dose of the injected contrast agent, the duration of the process, and the depth of catheter inserted into the coronary artery orifice. If the coronary blood flow is blocked for too long, the surgery can lead to myocardial ischemia, cardiac electrical instability, pericardial tamponade, coronary artery dissection, and even acute myocardial infarction. (2) Adverse reactions of contrast agents, mainly allergic reactions and renal toxicity. (3) Vagus reflex. It is more common in coronary angiography, occurring after the surgery, or when the arterial sheath is removed. A variety of stimulating factors take actions on the cortical center and hypothalamus. By suddenly increasing the autonomic nerve tension of choline lipids, it causes strong reflexive expansion of small blood vessels in the internal organs and muscles. The patients appear pale, blood pressure drops, heart rate slows down,

* Corresponding author.
E-mail addresses: liwenqi210@163.com (W. Li), zm@rjh.com.cn (M. Zuo), zhaohongjin11@163.com (H. Zhao), xuqi@dhu.edu.cn (Q. Xu), chendehua@dhu.edu.cn (D. Chen).

https://doi.org/10.1016/j.ymeth.2021.12.009
Received 5 June 2021; Received in revised form 16 December 2021; Accepted 18 December 2021
Available online 23 December 2021

yawning, cold sweats, nausea, vomiting, blurred vision, and even cardiac arrest. (4) Inspection is invasive. Bleeding, hematoma, pseudoaneurysm and arteriovenous fistula will occur at the puncture site. It also takes a long time to stop bleeding after the operation. (5) Patients with femoral artery puncture need short-term hospitalization for about three days. (6) Radiation during coronary angiography has certain effects on the human body. (7) The cost of coronary angiography is high, which is a great economic expense for poor patients.

Blood biochemical test indicators can illustrate the degree of coronary artery obstruction[1–4]. After myocardial injury, cardiac troponin complexes are released into the blood, and the content of this substance in the blood increases gradually and remains for a long time. Therefore, when a patient has symptoms of heart disease, it is often necessary to do the blood tests to observe the troponin level. Doctors can tell whether the patient is suffering from heart disease according to the results of the test and to what extent. Therefore, blood tests can be used to predict the risk of heart disease. On the other hand, the heart color Doppler echocardiography reports are also very useful to diagnose coronary heart disease. It uses the Doppler principle and a series of electronic technologies to display the spectrogram of the blood flow of a certain volume (SV) at a certain point in the heart or large blood vessels in real time under the situation of two-dimensional echocardiogram positioning. Coronary atherosclerotic heart disease is a kind of coronary artery atherosclerotic lesion that causes stenosis or obstruction of the vascular cavity, resulting in myocardial ischemia, hypoxia, or necrosis. In this paper, a multitask progressive deep networks model is proposed to predict the degree of coronary artery occlusion by heart color Doppler echocardiography report, twenty-one items of blood biochemical tests and other basic body information about the patients(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history).

The model takes each coronary artery occlusion prediction as a task and eight tasks are executed simultaneously. The model is divided into four parts. The first part is deep reinforcement learning (DRL) pretraining. Asynchronous advantage actor-critic (A3C) method is used to solve the problem efficiently. We use training data to optimize the recurrent neural network that parameterizes the stochastic policy. The second part is a soft parameter sharing layer composed of eight bidirectional LSTM, which ensures the similarity of model parameters by regularizing the distance of model parameters. The third part is a hard parameter sharing layer composed of a shared dropout layer, which reduces the risk of overfitting. And the fourth part is composed of a progressive networks of three dense layers. The progressive neural networks can transfer the prediction model of one blood vessel to the prediction of another blood vessel by storing the migration knowledge and extracting valuable features layer by layer, which can improve the prediction accuracy of the blood vessel with less data. The experimental results show that the model achieves better results than the general machine learning models, and is also superior to the single task model. According to the experimental results, we can draw the conclusions that the combined reinforcement multitask progressive time-series networks do facilitate mutual learning between tasks, thereby improving the overall performance of the model.

2. Exploration of prediction models for coronary heart disease

2.1. Statistical regression models

Coronary heart disease has received widespread attention in the academic community. The academic community has conducted in-depth research on coronary heart disease risk prediction models from different aspects and achieved some results. Framingham pioneered the use of traditional risk factors of heart disease to establish a predictive model based on multiple regression equations to predict heart disease[5]. Subsequently, it published a number of heart disease risk prediction models to predict different types of heart disease[6]. In addition to the Framingham model, the researchers have also developed SCORE, PROCAM, Reynolds, QRISK and other risk assessment models to predict heart disease in different countries and regions. The Tianjin Information Sensing and Intelligent Control Laboratory used a correlation between cardiac motion signals and ECG (Electrocardiogram) signals to develop a regression prediction model to achieve prediction of heart disease. However, these heart disease prediction models are based on statistical regression models and have limitations when predicting heart disease.

2.2. Machine learning models

With the development of information science, many machine learning algorithms have been applied to disease prediction. At the same time, more and more models of risk prediction for heart disease have emerged. Yan et al.[7] proposed a decision support system to diagnose five types of heart diseases in 2003. The system is constructed by a multilayer perceptron neural network with an accuracy of 63.6% to 82.9%. Polat et al.[8] used artificial immune recognition and K-nearest neighbor algorithm in 2006 to establish a coronary heart disease screening and diagnosis system based on the clinical information of patients, and the accuracy of the system was as high as 87%. Tsipouras et al.[9] proposed a rule-based fuzzy decision support system in 2008, which has a certain effect on distinguishing patients with coronary heart disease from normal people, with an average sensitivity of 80% and an average specificity of 65%. Babagolu et al.[10] used a support vector machine to classify coronary heart disease in 2010. They combined the motion data of the examinee and reduced the number of variables in the data set from 22 to 18 through dimensionality reduction. The final classification accuracy was 81.46%. Alizadehsani et al.[11] used a variety of data mining algorithms to predict coronary heart disease in 2013, in which the accuracy of the Bagging algorithm and the sequence minimum optimization algorithm were both 89%, and the accuracy of the artificial neural network was 85%. The data set for this experiment comes from the Rajaie Cardiovascular Medical Research Center. The data set includes 54 characteristics of 303 patients. In 2016, the team published several papers on coronary heart disease research results, adding the degree of blockage of the left circumflex artery, left anterior descending artery, and right coronary artery among the eight coronary arteries of the heart as new features into the data set. Using the feature selection method represented by information gain, an association rule algorithm model was established to determine the blockage of these three vessels. The final accuracy rate was 83.17% for the left circumflex artery, 86.14% for the left anterior descending artery, and 83.50% for the right coronary artery[12]. In the same year, Luxmi Verma et al.[13] proposed a hybrid data mining model, which used correlation-based feature subsets and particle swarm optimization algorithm, combined with the K-means algorithm. The team first experimented with a variety of supervised learning algorithms, and found that polynomial logistic regression achieved the best results on a given data set. Then, the polynomial logistic regression was added to the improved hybrid data mining model, which improved the classification accuracy from 83.5% to 87.73%. Nahar et al.[14] used association rules to predict coronary heart disease. Benjamin et al.[1] obtained the relationship between heart disease mortality and reactive protein from the perspective of a single influencing factor. Wang Fengli[15] pointed out in his graduation thesis that the improved SVM-AR-SVM (Support Vector Machine-Auto Regressive-Support Vector Machine) algorithm has been applied to the prediction of heart disease by some scholars. Pang Xiantao[16] used a three-layer BP (Back Propagation) neural network to predict heart disease, and studied the number of hidden neurons, transfer function, learning rate and learning objectives to establish a risk prediction model for heart disease.

2.3. Combined models

In addition, some scholars have also tried to predict heart disease by

combining multiple mathematical models. Zhu Yue et al.[17] combined support vector machines with particle swarm optimization algorithms to establish a coronary heart disease classification prediction model, and compared the model with the artificial neural network, logistic regression and other algorithms. The experimental results showed that the artificial neural network had the worst effect, and support vector machine achieved the best classification results. Wang Fengli combined BP neural network and DS (Dempster-Shafer) evidence theory to predict heart disease[18], which improved the prediction accuracy and showed good robustness of the algorithm. Xu Dong et al. combined LMBP (Levenberg Marquardt BackPropagation) neural network and Rough Sets (RS) to predict heart disease[19]. Orphanoua et al.[20] used incorporating repeating temporal association rules in Naïve Bayes classifiers to diagnosis coronary heart disease. Zreik1 et al.[21] used a recurrent convolutional neural network in coronary CT angiography for automatic detection and characterization of coronary artery plaque and stenosis. For the detection of stenosis and the determination of anatomical significance, the accuracy of this method was 80.0%.

In the above-mentioned studies, most of the experiments used few indicators for risk assessment of coronary heart disease, and did not consider the overall situation of the patients. The accuracy of coronary heart disease prediction in most literature is below 90%, which cannot meet the requirements of clinical practice. In the literature using deep learning algorithms, most of them analyze images of coronary angiography. Because coronary angiography has a series of shortcomings mentioned in the introduction, many doctors and patients do not want to use coronary angiography. Therefore, it is urgent to find a method that can quickly, effectively and accurately diagnose coronary heart disease at a small cost.

3. The architecture of proposed approach

In clinical diagnosis, blood biochemical indicators are often used to assess if there is a risk for coronary heart disease. According to[2–4], blood test indicators will reflect the degree of clogging of coronary arteries to a certain extent. After myocardial injury, myocardial troponin complex will be released into the blood, and the content of this substance in the blood will then gradually increase and remain for a long time. As a result, one way of diagnosing acute myocardial infarction is by assessing myocardial injury markers, of which cardiac troponin is the main parameter. The examination is easy, cheap and less damage to the patients, making it a very popular means of assessing the risk of heart disease. In this article, we propose a combined reinforcement multitask progressive networks model to diagnose cardiovascular occlusion. The model predicts the level of coronary artery occlusion (I∼IV) by basic body information(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history) and multiple blood biochemical tests of people on different time. The progressive neural network can store the parameters and extract valuable features, then predict the degree of occlusion in eight branches through the heart color Doppler echocardiography reports, twenty-one blood indicators and some basic information such as patients' gender, age, diabetes, blood pressure, blood sugar and so on.

The input of the model are the heart color Doppler echocardiography reports, twenty-one blood biochemical indicators and patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history. The twenty-one indicators of blood biochemical test are: APB (apolipoprotein B), ApoA (apolipoprotein A), ApoB (apolipoprotein B), ApoE (apolipoprotein E), AST (aspartate aminotransferase), CHOL (total cholesterol), CK (creatine kinase), CK_MBM (CK-MB mass), CTNI (troponin I), LDH (lactate dehydrogenase), LDH2 (lactate dehydrogenase II), LPA (lipoprotein a), MYO (myoglobin quantification), NEFA (non-esterified fatty acid), PAB (pre-albumin), TG_B (triglyceride), TP (total protein), TP2 (total protein II), UHDL (high-density lipoprotein cholesterol), ULDL (low-density lipoprotein cholesterol), uPA (pre-albumin).

The model outputs the level of occlusion of the eight vessels. The eight arterial vessels are: LM (left main trunk), LAD (left anterior descending branch), LCX (left circumflex artery), OM (blunt margin branch), RCA (right coronary artery), LCA (left coronary artery), PLA (left ventricle posterior branch) and PDA (posterior descending branch).

According to the World Health Organization standard, the degree of coronary artery stenosis is divided into 4 levels according to the reduction of lumen area: coronary artery stenosis grade I lesion: lumen area is reduced by 1% to 25%; Grade II lesion: lumen area is reduced by 26% ∼ 50%; Grade III lesions: the lumen area is reduced by 51% to 75%; Grade IV lesions: the lumen area is reduced by 76% to 100%.

3.1. Architecture details

The model in this paper predicts the degree of occlusion of eight arteries by the heart color Doppler echocardiography reports, twenty-one blood parameters and some basic information(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history). In the previous complicated data processing process, we found that for the same subject, if one coronary artery occlusion is severe, other blood vessels will be more or less blocked, and there is a strong correlation. That is to say, the prediction of a blood vessel is related to the prediction of other vascular occlusions in the same person. Based on this, multi-task deep learning is the best choice for the prediction of eight coronary vascular occlusions. In this multi-task learning model, we use a combination of soft sharing of the network parameters and hard parameter sharing. In soft sharing, each vessel's predictive task has its own model and parameters. We can ensure the similarity of the parameters by regularizing the distance of the model parameters. The hard sharing mechanism of parameters reduces the risk of over-fitting. Further data analysis found that among the eight blood vessels, four to five vessels are often blocked, the others are less blocked. For the blood vessels that are not often blocked, we cannot get enough data for model training (the specific training data set situation is detailed in the experiments section of this article). The vessel LAD (left anterior descending branch) has the most training data, so we use LAD data as the first training data set to train the first column. The vessels like LAD, RCA and LCX have many data, the vessels like PLA and LCA have few data. So, the well-trained model parameters of blood vessels such as LAD, RCA and LCX are migrated to other blood vessels prediction branch models such as PLA and LCA. This method can improve the accuracy of overall model. Continual learning is a long-standing goal of machine learning, where agents not only learn a series of services experienced in sequence, but also have the ability to transfer knowledge from previous tasks to improve convergence speed [22]. In this article, the specific implementation method is to use the multitask progressive neural networks-based progressive deep learning method, which integrate these requirements directly into the model architecture, to predict the degree of coronary vascular occlusion. In this model, catastrophic forgetting is prevented by instantiating a new neural network (a column) for each task, while transfer is enabled via lateral connections to features of previously learned columns. The progressive neural network can migrate knowledge layer by layer and extracting valuable features. The knowledge of a coronary artery vascular progressive model can be used to predict another coronary arteries. This method can greatly improve the prediction accuracy of other vessels like PLA and LCA. The overall model is shown in Fig. 1.

Fig. 1 depicts the combined reinforcement multitask progressive networks (CRMPN). The input data of the network is heart color Doppler echocardiography reports, twenty-one blood test indicators(see the above for the twenty-one indicators) and patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history. The output is the degree of blockage of a patient's eight coronary arteries.

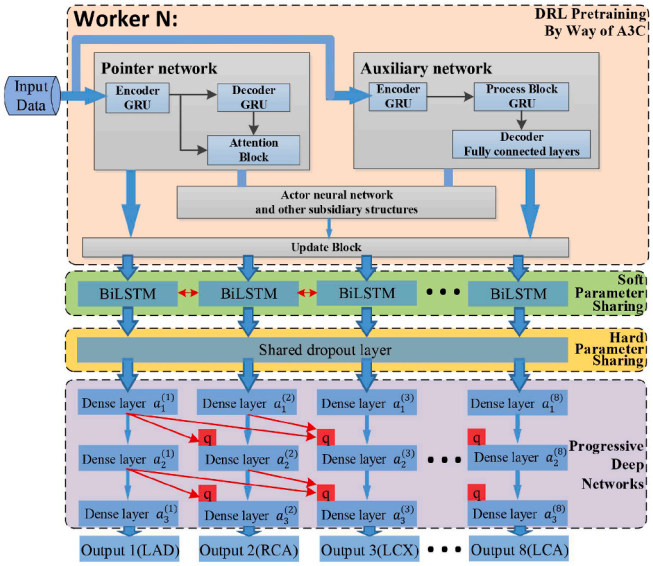

Fig. 1. The figure illustrates the main architecture of proposed combined reinforcement multitask progressive networks. It contains three parts: (1) Pre-training of reinforcement learning using A3C; (2) Parameter sharing layer containing BiLSTM layer using soft parameter sharing mechanism and shared dropout layer using hard parameter sharing mechanism; (4) Progressive deep networks.

3.1.1. DRL pretraining by way of A3C

In this model, the first part is pre-training of reinforcement learning using A3C(The pink part of Fig. 1). In the A3C architecture, there are multiple workers depending on the number of CPUs. Each worker includes a pointer network, an auxiliary network, actor network and other subsidiary structures. A pointer network consists of an encoder and a decoder. Each encoder and decoder is composed of a GRU. GRU has a control unit that can adjust the information flow within the unit, and there is no separate memory unit. And it does not have any mechanism to control the extent to which its state is exposed, but exposes the entire state every time. This encoder is the policy network in the framework. When the data is sent into the encoder, the encoder reads each symbol of the sequence. As the encoder reads each symbol, the hidden state of the GRU changes to

$$h_{<t>} = f(h_{<t-1>}, x_t) \tag{1}$$

where f is the nonlinear activation function.

Then the encoder transforms it into latent memory states. After reading the end of the sequence, the hidden state of the GRU is a summary of the entire input sequence. The output of the encoder serves as input to the decoder, and the outputs of both encoder and decoder are sent to the attention block to obtain a probability distribution. The decoder is another GRU that maintains its latent memory state. The pointing mechanism is parameterized by an attention vector u and two attention matrices M_ref, M_p, as follows:

$$w_i = \begin{cases} u^T \cdot tanh\left(M_ref \cdot d_i + M_b \cdot b\right), & i \neq \varphi(k), k < i \\ -\infty, & otherwise \end{cases} \tag{2}$$

$$G(ref, b; M_ref, M_b, u) \overset{def}{=} softmax(w) \tag{3}$$

The strategy gradient method and stochastic gradient descent method are used to optimize the parameters. The purpose of using policy gradients is to directly perform output operations. The system can select actions in a continuous range. In neural networks, backpropagation is used as a reward mechanism. Policy is expressed by $\varphi_\delta, log\varphi_\delta(s_t, a_t)$ indicates the degree of surprise caused by the selected action when in state s. We use the following reinforcement algorithm(Algorithm 1)[23] to express the gradient of the above formula.

Algorithm 1 Function REINFORCE

1: Initialize δ arbitrarily
2: **for** each epospde $\{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \sim \varphi_\delta$ **do**
3: **for** t = 1 to T-1 **do**
4: $\delta \leftarrow \delta + \alpha \nabla_\delta log\varphi_\delta(s_t, a_t) v_t$
5: **end for**
6: **end for**
7: **return** δ

All parameters of the pointer network are represented by δ. An auxiliary network consists of an encoder, a process block and a 2-layer ReLU neural network decoder. The encoder and process block are both composed of GRUs. When the encoder receives the data, it encodes it into latent memory states and the hidden state. The output of the encoder is used as the input to the process block. The output of the process block is then decoded into the scalars as the baseline prediction by the decoder. All parameters of the auxiliary network are represented by δ_u. The parameters δ_u and δ are sent into function to get iterative update. A3C algorithm is used for pre-training, which creates multiple parallel environments. Multiple agents with secondary structures update parameters in the main structure simultaneously in these parallel environments. The agents in parallel do not interfere with each other, and the update of the parameters in main structure is disturbed by sub-structure commit update discontinuously. Therefore, the correlation of updates is reduced, and convergence is improved.

3.1.2. Parameter sharing

The second part is the BiLSTM layer(The green part of Fig. 1). The trained parameters gained in A3C are used as the input of the second part. The parameters of the bidirectional LSTM adopt a soft sharing mechanism. Each LSTM has its own parameters, and the similarity of the parameters is ensured by regularizing the distance of the model parameters. The third part uses the shared dropout layer, which is hard parameter sharing(The yellow part of Fig. 1). In fact, the literature[24] proves that the order of the over-fitting risk of these shared parameters is T, where T is the number of tasks and is less risky than the over-fitting of the task-related parameters. This is very meaningful. The more tasks are learned at the same time, the more the same representation of the task can be captured by the model, resulting in less risk of over-fitting on the original task. Evidence from the network visualization suggests that hierarchical representations of deep networks are incrementally semantically moving from the bottom to the top. In multitask deep networks, the sharing of low-level semantic information helps to reduce the amount of computation. At the same time, the shared presentation layer can make several common tasks better combine the correlation information, and the task-specific layer can separately model for its task. This enables the unification of shared information and task-specific information.

3.1.3. Progressive deep networks

Next is the three layers of dense layers(The purple part of Fig. 1), using the progressive neural network, where the red box labeled q represents the adapters, the role is to keep the hidden layer activation value of the front row consistent with the original input dimension. In this model, the composition of the progressive neural network can be described as:

(1) A progressive network starts with a single column: a deep neural network having N layers with hidden activations $a_k^{(1)} \in \mathbb{R}^{u_k}$, with u_k the number unites at layer k ⩽ N, and parameters $\Phi^{(1)}$ trained to convergence. The one deep neural network constructed in the first column predicts the degree of occlusion of one coronary artery based on the heart color Doppler echocardiography reports, twenty-one

blood biochemical indicators and some basic information about patients(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history). Since there is no additional input in the first column as a parameter reference for the neural network, we select the vessel LAD (left anterior descending branch) data to train the first column. (As is mentioned above, LAD has the most data.)

(2) In order to train the second task, construct the second column deep neural network by fixing the parameters of the first column neural network $\Phi^{(1)}$, and a new column with parameters $\Phi^{(2)}$ is instantiated, where layer $a_k^{(2)}$ receives input from both $a_{k-1}^{(2)}$ and $a_{k-1}^{(1)}$ via lateral connections. The activation values of the various hidden layers in the first column of neural networks are then processed by the adapter and then connected to the corresponding layer of the second column of neural networks as an additional input. Because the vessel RCA has the second most data, we use RCA data to train the second task.

(3) In order to train the third task, construct the third column neural network by fixing the parameters of the first two columns of neural networks. The activation values of the hidden layers of the first two columns are processed by the adapter and then connected to the current neural network. The corresponding layer is used as an additional input. Because the vessel LCX has the third most data, we use LCX data to train the third task.

(4) Similarly, the fourth prediction task uses OM data, the fifth prediction task uses PDA data, and the sixth task uses LM data. The PLA data and LCA data are used for the seventh and eighth tasks. This generalizes to T tasks as follows:

$$a_k^{(t)} = f\left(W_k^{(t)} a_{k-1}^{(t)} + \sum_{i<t} L_k^{(t:i)} a_{k-1}^{(i)}\right) \tag{4}$$

Where $W_k^{(t)} \in \mathbb{R}^{u_k \times u_{k-1}}$ is the weight matrix of layer i of column k, $L_k^{(t:i)} \in \mathbb{R}^{u_k \times u_i}$ are the lateral connections from layer $k-1$ of column i, to layer k of column t and a_0 is the input of the network. f is an element-wise non-linearity.

In a series of sequence tasks, the progressive neural network stores the knowledge of migration and extracts valuable features through layer-by-layer advancement, completing the migration of knowledge. The advantage of this progressive neural network-based progressive DRL method is that[25]: for the new task, the hidden layer state of the previous training model is preserved during training, and the useful features of each hidden layer in the previous network are hierarchically combined. So that progressive learning has a priori knowledge of long-term dependence. As the lateral connections $L_k^{(t:i)}$ are only from column t to columns $i < t$, previous columns are not affected by the newly learned features in the forward pass. Because also the parameters $\Phi^{(i)}; i < t$ are kept frozen when training $\Phi^{(t)}$, there is no interference between tasks. Therefore, the catastrophic forgetting is well avoided.

In practice, the progressive network layer of Eq. 6 is augmented with non-linear lateral connections. The lateral connections are called adapters, as indicated by the red box a in Fig. 1. The adapters can not only improve initial conditioning but also perform dimensionality reduction. The vector of anterior features of dimensionality $u_{k-1}^{(<t)}$ are defined as follows:

$$a_{k-1}^{(<t)} = \left[a_{k-1}^{(1)} \ldots a_{k-1}^{(i)} \ldots a_{k-1}^{(t-1)}\right] \tag{5}$$

in the case of dense layers, the linear lateral connection is replaced with a single hidden layer MLP. Its role is to adjust for the different scales of the different inputs. The hidden layer of the non-linear adapter is a projection onto a u_k dimensional subspace. This ensures that the number of parameters stemming from the lateral connections is in the same order as $\left|\Phi^{(1)}\right|$ when the index t grows. Omitting bias terms, we get:

$$a_k^{(t)} = \lambda\left(W_k^{(t)} a_{k-1}^{(t)} + L_k^{(t:i)} \lambda\left(P_k^{(t:i)} p_{k-1}^{(<t)} a_{k-1}^{(<t)}\right)\right) \tag{6}$$

where $P_k^{(t:i)} \in \mathbb{R}^{u_{k-1} \times u_{k-1}^{(<t)}}$ is the projection matrix.

A analytical method derived from the Fisher Information[26] is explored to study in detail which features and at which depth transfer actually occurs. A local approximation to the perturbation sensitivity can be got by using the Fisher Information matrix[26]. The diagonal matrix $\widehat{D}$ with elements $r \times r$ is defined, and the derived Average Fisher Sensitivity (AFS) of feature r in layer k of column t as:

$$\widehat{D}_k^{(t)} = E_{\sigma(h,q)}\left[\frac{\partial \log \varphi}{\partial \widehat{a}_k^{(t)}} \frac{\partial \log_\varphi T}{\partial \widehat{a}_k^{(t)}}\right] \tag{7}$$

$$AFS(k,t,r) = \frac{\widehat{D}_k^{(t)}(r,r)}{\sum_t \widehat{D}_k^{(t)}(r,r)} \tag{8}$$

Where φ is the network policy, and the expectation is over the joint state-action distribution $\sigma(h,q)$ induced by the progressive network trained on the target task. It is useful to consider the AFS score per-layer as follows:

$$AFS(k,\ r) = \sum_r AFS(k,t,r) \tag{9}$$

Thus, the AFS estimates how much the network relies on each feature or column in a layer to compute its output.

3.2. Asynchronous advantage actor-critic

Asynchronous advantage actor-critic (A3C) method is used as the training algorithm. This approach can solve the non-convergence problem of using an actor-critic alone with a much faster training speed. A3C creates multiple parallel environments and allows multiple agents with secondary structures to update parameters in the primary structure simultaneously in these environments.

The agents in parallel do not interfere with each other, and the update of the parameters in main structure is disturbed by substructure committing update discontinuously. Therefore, the correlation of updates is reduced, and convergence is improved. The A3C algorithm(Algorithm 2) actually puts Actor-Critic on multiple threads for synchronous training. The Asynchronous Advantage Actor-Critic(A3C in Algorithm 2) [27] maintains a policy $\delta(a_t|s_t;\delta')$ and an estimate of the value function $V(s_t,\delta_u')$. The policy and the value function are updated after every t_{max} actions or when a terminal state is reached. The update performed by the algorithm can be seen as $\nabla_{\delta'} log\varphi(a_i|s_i;\delta')A(s_t,a_t;\delta,\delta_u)$ where $A(s_t,a_t;\delta,\delta_u)$ is an estimate of the advantage function given by $\sum_{i=0}^{k-1}\gamma^i r_{t+1} + \gamma^k V(s_{t+k};\delta_u) - V(s_t;\delta_u)$, where k can vary from state to state and is upper-bounded by t_{max}.

Fig. 2 shows the architecture of A3C system. The number of sync and workers can be set according to the number of cores in the current experimental processor, so that each core has a worker running in it. The experimental environment in the current study is a 16-core processor. So sixteen workers (work 0 ~ work 15) were assigned which ran on sixteen different cores in parallel. Work 0 is the 0th worker, and each worker can share the *Global Net*. If we call pull in sync, the worker will get the latest parameters from *Global Net*. If we call *push* in sync, the worker will push its own personal update to the Global Net. Tensorflow was employed to build the neural network. To calculate the actor virtual loss, the *td error* provided by the critic was used as a guide for gradient ascent. The critic

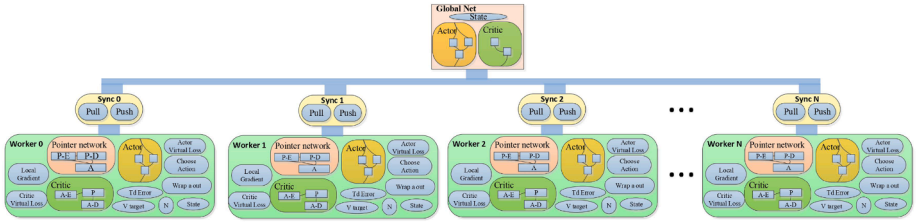

Fig. 2. The architecture of A3C system. A3C creates multiple parallel environments and puts the actor-critics on multiple threads for simultaneous training.

is very simple: it only needs to get its value for the state. It is used to calculate the *td error*.

Algorithm 2 Asynchronous Advantage Actor-Critic

1://Assume global shared number of training steps T
2: Initialize thread step counter $t \leftarrow 1$
3: Initialize network params
4: **repeat**
5: Reset gradients: $d\delta \leftarrow 0$ and $d\delta_u \leftarrow 0$.
6: Synchronize thread-specific parameters $\delta' = \delta$ and $\delta'_u = \delta_u$
7: $t_{start} = t$
8: Get state s_t
9: **repeat**
10: Perform a_t according to policy $\delta(a_t|s_t;\delta')$
11: Receive reward r_t and new state s_{t+1}
12: $t \leftarrow t+1$
13: $T \leftarrow T+1$
14: **until** terminal s_t or $t - t_{start} == t_{max}$
15: $R = \begin{cases} 0, & for terminals_t \\ V(s_t, \delta'_u), & for\ non-terminal\ s_t \end{cases}$
16: **for** i $\in \{t-1, \ldots, t_{start}\}$ **do**
17: $R \leftarrow r_i + \gamma R$
18: $d\delta \leftarrow d\delta + \nabla_{\delta'} log\varphi(a_i|s_i;\delta')(R - V(s_i;\delta'_u))$
19: $d\delta_u \leftarrow d\delta_u + \partial(R - V(s_i;\delta'_u))^2 / \partial\delta'_u$
20: **end for**
21: Perform asynchronous update of δ using $d\delta$ and of δ_u using $d\delta_u$
22: **until** $T > T_{max}$

3.2.1. Actor-critic neural network

In Actor-Critic, Actor's predecessor was policy gradients, which allowed it to effortlessly select the appropriate action in continuous action. Critic's predecessor is Q-learning or other value approximation methods that allow single-step updates, while the traditional policy gradients are round-up updates. Single-step update can improve learning efficiency. Critic can see the potential rewards of the current state by learning the relationship between the environment and rewards. So using it to direct the Actor will make the Actor update in every step. Actors choose various actions in practice. When Gradient ascent is implemented using the Policy Gradient method, Critic tells the actor whether the current action is good or bad. If the next state is better than the current state, the *td error* is larger and the actor update rate is increased. If the score is not good this time, the critic will tell the actor that ascent is less.

The critic network can make predictions based on the final state. The GRU architecture plays an important role in this. Critics are parameterized by δu, which is trained using stochastic gradient descent with the mean square error as the objective function:

$$\mathscr{L}(\delta_u) = \frac{1}{Q}\sum_{i=1}^{Q}\left|\left|q_{\delta_u}(a_i) - L(\varphi_i|a_i)\right|\right|_2^2 \tag{10}$$

Actor and critic are combined into a complete system, which is easy to run. In addition to actor and critic, the worker network also has its own class that is used to perform the work in each thread. The coordinator class in Tensorflow is used to make multiple threads parallel computing while the join in coordinator is used for thread scheduling. When syncing, we update global parameters by *pull* and get global parameters by *push*.

3.3. Combined reinforcement multitask progressive time-series networks model

In order to further improve the accuracy of predicting the degree of blockage of the patient's eight coronary arteries, we propose combined reinforcement multitask progressive time-series networks model(Fig. 3). On the basis of the above model framework, the last part of the model is changed to an incremental combined progressive time-series networks frameworks. The network will update itself through incremental learning, and adjust the balance between the new and old observations by adding a forgetting factor to the incremental feature base update. Incremental combined progressive time-series networks are very different from progressive neural networks. The overall model is shown in Fig. 3.

The progressive time-series networks model(The purple part of Fig. 3) mainly includes three parts: feature processing layer, incremental combination layer, and prediction layer. In the feature processing layer, the real-valued vector converted from the text is processed. After processing, the output vector is combined with the previous vector and input to the next layer. In incrementally combined progressive time-series networks, it is very important to reduce the weight of early data. When tracking a changed target, the latest data of the target reflects its recent trend of change more than the previous historical data. Therefore, when predicting the degree of coronary artery blockage in a patient, the network will pay more attention to the blood test data

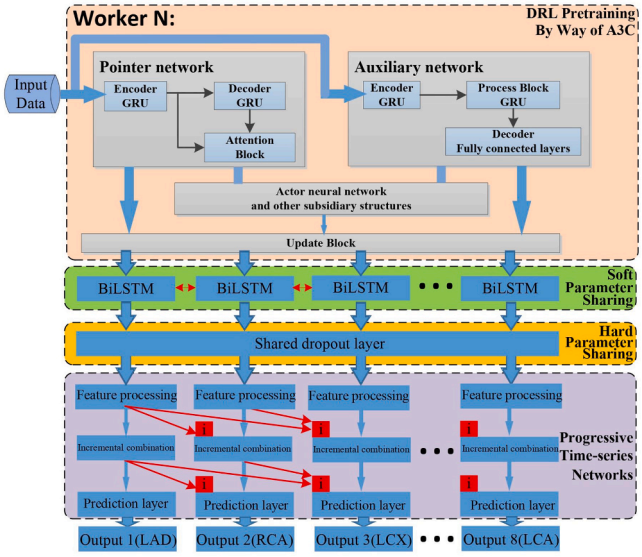

Fig. 3. The figure illustrates the main architecture of combined reinforcement multitask progressive time-series networks model. It contains three parts: (1) Pre-training of reinforcement learning using A3C; (2) Parameter sharing layer containing BiLSTM layer using soft parameter sharing mechanism and shared dropout layer using hard parameter sharing mechanism; (3) Incremental combination progressive time-series networks.

obtained recently, rather than the previous blood test history result data. And with the passage of time, the amount of historical data will become very large. In order to prevent the weight of historical data from exceeding the weight of the latest data and reflecting the trend of the latest data in future forecasts, the network will update itself through incremental learning, and adjust the balance between the new and old observations by adding a forgetting factor to the incremental feature base update. When the network receives the data vector values of multiple blood biochemical examinations of a patient, the weight of the most recent blood test result is set to the maximum, and the weights of other blood test results decrease sequentially with the examination time. The objective function of the model is a binary cross-entropy function, through training, the value of the function will reach the minimum. Use the following formula to update the weight of the hidden layer in the model:

$$w_i = w_i - \eta \frac{\partial E}{\partial w_i} \tag{11}$$

$$\delta_i = \begin{cases} \left(y_i^{(n)} - \widehat{y}_i^{(n)}\right)\left(1 - tanh^2(net_i)\right), & i \in \mathbf{o} \\ \left(\sum_{j \in J} \delta_j w_{ij}\right)\left(1 - tanh^2(net_i)\right), & i \in \mathbf{h} \end{cases} \tag{12}$$

3.3.1. Incremental combination progressive time-series networks

The first two parts of the model are the same as before. In the third part of the model, incremental combined progressive time-series networks use an incremental algorithm(The purple part of Fig. 3). We input the patients' multiple blood test records into the model in chronological order. Whenever new blood test data is added, it only needs to be updated based on the original knowledge base according to the data changes, without the need to reconstruct the knowledge base. The latest data is set with the highest weight, so that the latest data can reflect the recent trend of the patients' condition more than the previous historical data. The prediction accuracy and overall efficiency of the model have been improved. This is more in line with the principle of human thinking.

When there is a new input, the incremental learning model will update itself, which ensures that the model can quickly and timely capture the latest trends in the patient's condition and is available for a long time. Therefore, after the model pre-training stage, overall networks after learning including the pseudo-inverse matrix A_n and the weight matrix W_n can be updated. A_n is an $n \times m$ matrix, representing the input matrix. This matrix contains all the input vectors and the reinforcement part. $a^{'}$ is a new sample of m $\times$ 1 dimensions used to update the model. Therefore, the new matrix A_{n+1} is defined as follows:

$$A_{n+1} \triangleq \begin{bmatrix} A_n \\ a^{'} \end{bmatrix} \tag{13}$$

Where n represents the number of samples, which is also a discrete time series instance, and m is the dimension of the input paradigm. Therefore, the update of the pseudo-inverse matrix A_{n+1} can only be based on the pseudo-inverse matrix A_n and the new input vector a. The calculation process is as follows:

$$A_{n+1}^{+} = \left[A_n^{+} - db^{'} \middle| d\right] \tag{14}$$

among them,

$$b^{'} = a^{'} A_n^{+} \tag{15}$$

$$d = (1 + b^{'} b)^{-1} A_n^{+} b \tag{16}$$

Like the matrix A_n, in the model, the output vector is represented by Y_n, and the weight matrix is represented by W_n. The new output is y. According to the least squares method, we can get:

$$W_{n+1} = A_{n+1}^{+} Y_{n+1} \tag{17}$$

$$W_n = A_n^{+} Y_n \tag{18}$$

Therefore, the new weight matrix W_{n+1} can be calculated according to the following formula:

$$W_{n+1} = W_n + (y^{'} - a^{'} W_n) \tag{19}$$

In the above formula, d is the optimal learning rate for weight update. The pseudo-inverse matrix A_n and the weight matrix W_n are updated in the incremental learning stage according to the following steps:

1) When a new example is input to the network, the model decomposes the signal and obtains a new input paradigm A_{n+1} by combining all the new outputs from all subsequences. Then according to the above formula, the new weight matrix W_{n+1} can be updated.
2) Use the validation set to check the error rate. If the error rate decreases, the model will remain updated; otherwise, the weight matrix will remain unchanged.
3) Whenever a new sample is entered into the network, repeat steps one and two.

3.3.2. Forgetting factor

In predicting the degree of coronary artery blockage in patients, more attention needs to be paid to the recently obtained blood test data, rather than the previous blood test history result data. Because when tracking a changed target, the target's latest data reflects its recent trend of change more than the previous historical data. In incrementally combined progressive time-series networks, it is very important to reduce the weight of early data. And with the passage of time, the amount of historical data will become so huge that the weight of historical data exceeds the weight of the latest data, so that the trend of the latest data cannot be well reflected in future forecasts. Therefore, in the model, when the patients' blood test results are input for multiple times, the weight of the most recent blood test result is set to the maximum, and the weights of other blood test results decrease sequentially with the inspection time.

One way to adjust the balance between the old and new observations is to add a forgetting factor to the incremental feature base update.[28] For this reason, in each update, multiply the singular value by the scalar factor $f \in [0, 1]$. When $f = 1$, it means that there will be no forgetting. Therefore, in the formula,

$$R = \begin{bmatrix} f\Sigma & U^T \widehat{B} \\ 0 & \widetilde{B}\left(\widehat{B} - UU^T \widehat{B}\right) \end{bmatrix} \tag{20}$$

The above formula is equivalent to using QR decomposition $[fU\Sigma\widehat{B}]$ instead of $[U\Sigma\widehat{B}]$.

The essence of this method is that every time the feature base is updated, a carefully selected additional vector is used to correct the time-varying average, thereby adding new training data. Give the following lemma:

Lemma 1. Use $A = [I_1, I_2, \ldots, I_n], B = [I_{n+1}, I_{n+2}, \ldots, I_{n+m}]$ to represent the data matrix, and use $C = [A \quad B]$ as the matrix after they are concatenated. Use $\bar{I}_A, \bar{I}_B$, and $\bar{I}_C$ to represent the mean values of A, B, and C, and use S_A, S_B, S_C to represent the hash matrix of A, B, and C respectively. Therefore, we can get,

$$S_C = S_A + S_B + \frac{nm}{n+m}\left(\bar{I}_B - \bar{I}_A\right)\left(\bar{I}_B - \bar{I}_A\right)^T \tag{21}$$

In this lemma, we define the scatter matrix as the outer product of the central data matrix. E.g,

$$S_B = \sum_{i=n+1}^{m} \left(I_i - \bar{I}_B \right) (I_i - \bar{I}_B)^T \tag{22}$$

Therefore, the difference between the scatter matrix and the sample covariance matrix is only the scalar multiple:

$$S_B = mcov(B) \tag{23}$$

Through Lemma 1, we can see that the singular value decomposition of $(C - \bar{I}_C)$ is equivalent to the horizontally connected singular value decomposition of $(A - \bar{I}_A), (B - \bar{I}_B)$ and an additional vector $\sqrt{\frac{nm}{n+m}}(\bar{I}_B - \bar{I}_A)$. The incremental algorithm with mean update is as Algorithm 3.

Algorithm 3 Incremental algorithm with mean update

1://Givn U and Σ from the SVD of $(A - \bar{I}_A)$, as well as $\bar{I}_A, n$, and B, compute $\bar{I}_C$ as well as U' and Σ' from the SVD of $(C - \bar{I}_C)$

2: Compute the mean vectors $\bar{I}_B = \frac{1}{m}\sum_{i=n+1}^{n+m} I_i$, and $\bar{I}_C = \frac{n}{n+m}\bar{I}_A + \frac{n}{n+m}\bar{I}_B$

3: From the matrix $\widehat{B} = \left[\left(I_{m+1} - \bar{I}_B \right) \ldots \left(I_{n+m} - \bar{I}_B \right) \sqrt{\frac{nm}{n+m}} (\bar{I}_B - \bar{I}_A) \right]$

4: Compute $\widehat{B} = orth(\widehat{B} - UU^T\widehat{B})$ and $R = \begin{bmatrix} \Sigma & U^T\widehat{B} \\ 0 & \widehat{B}\ (\widehat{B} - U^T\widehat{B}) \end{bmatrix}$. Note that $\widehat{B}$ will be one column larger than in the SKL algorithm.

5: Compute the SVD of R: $R \overset{SVD}{=} \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$

6: Finally $U' = [U\ \widehat{B}]\widetilde{U}$ and $\Sigma' = \widetilde{\Sigma}$

It can be seen that the algorithm has low complexity and only needs a small amount of fixed overhead to store, update and correct the constantly changing sample mean.

In this model, we use the following lemma to analyze the influence of the forgetting factor on the result feature base:

Lemma 2. The forgetting factor of f reduces the contribution of each data block to the overall covariance, which is modeled by an additional f^2 factor when each singular value decomposition is updated.

Therefore, after the *Kth* update of the feature base, the covariance of the m observation blocks $(j < k)$ added during the *jth* update will have a weight drop, which is achieved by reducing the weight factor $f^{2(k-j)}$. The goal of dimensionality reduction is to find a $k-$*dimensional* subspace that retains as much data covariance as possible. The forgetting factor selects the direction with the larger covariance in the most recent data as the base vector direction, at the expense of the direction shown by the earlier data.

Since the previously observed data contributes less to the covariance, it is also necessary to reduce its contribution to the mean of the results. When the forgetting factor f is used, the mean update is modified as follows:

$$\bar{I}_C = \frac{fn}{fn+m}\bar{I}_A + \frac{m}{fn+m}\bar{I}_B \tag{24}$$

The effective size of the observation history is calculated at each update as $n \leftarrow fn + m$.

One benefit of including the forgetting factor in the mean update is that even if the actual observations are close to infinity, the mean can still change based on the new observations. Specifically, using $n \leftarrow fn + m$, the effective number of observations will reach equilibrium $n = fn + m$, or $n = m/(1-f)$. For example, when $f = 0.95$ and $m = 5$, new observations are included in each update, and the effective length of the observation history will be close to $n = 100$.

Although the structure of our model is relatively complex, but because of this, it can obtain more convincing results clinically than simple models. In the diagnosis of coronary heart disease, doctors will make an overall judgment based on comprehensive information such as the patients' age, chest pain, blood pressure, medical history, and some markers of myocardial damage. For patients with greater risk, the doctor will recommend coronary angiography for confirmation. By replacing coronary angiography, which is an invasive, expensive and associated with adverse reactions, with a more accurate model, patients can achieve the same clinical diagnosis at the lowest cost. In order to apply the model to clinical scenarios, we design 4 modules: (1) Data collection module. The input data of the model included basic patient information data(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history), 21 blood biochemical indicators data, color Doppler echocardiography reports data and coronary angiography data. We fused the first three types of data in series and extracted the degree of blockage of 8 coronary arteries from the fourth type of data as a label. (2) Word vector generation module. We converted the color Doppler echocardiography reports, blood biochemical indicators and patient information into word vector after word segmentation for subsequent training. (3) Model training module. We train and evaluate the multi-task progressive deep networks model and the multi-task deep reinforcement learning time series model respectively. (4) Model prediction module. We just put the trained model into the clinic settings. In the actual application, the doctor only needs to enter the patients' color Doppler echocardiogram report text, blood test index values, and patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history. When the model predicts, the data is first cleaned, then the trained word vector is used to process the cleaned data, and finally the trained model is used to predict the degree of blockage of the patient's eight coronary arteries.

4. Results

4.1. Experimental data

Experiments are conducted to investigate the behavior of the proposed combined reinforcement multitask progressive networks. The experiments in this paper were running on an Intel Core i9 7960X 16 cores CPU and a Nvidia Tesla K80 GPU. In the experiments of this paper, data come from several triple A hospitals in Shanghai. Since the same patient has multiple blood tests at different times, we will incrementally combine the multiple blood test records of the same patient. For example, we first select patients whose blood tests are between 2 and 15 times, and sort their blood test records in chronological order as part of the input data. Another part of the input data is the examination report text data (color Doppler echocardiogram report text) and patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history. These three parts of data are fused, and then the degree of blockage of the eight coronary arteries extracted from the coronary angiography data is used as a label, and finally matched according to the patients' medical card number and detection time. When the patient has only one blood test record, the multitask deep progressive networks model predicts better; when the patient has multiple blood test records, the multitask progressive time-series networks model predicts better. These information are the final experimental data. The data are divided into training sets, validation sets and test sets, and their proportions are 7/10, 1/10, and 2/10. (Not at a patient level. The total number of patients is 17,885. Among them, 70% are used as the training set, that is, 12520 patients. 10% are used as the validation set, that is, 1788 patients. 20% as the test set, that is, 3577 patients. It does not split the blood test results of a patient at multiple time points into training set, validation set, and test set.) The blood biochemical indicators and basic body information of patients are described in Table 1.

Cardiac color Doppler ultrasound can display the spectrogram of blood flow of a certain volume at a certain point in the heart or large blood vessels in real time. It is a heart examination technique that is non-invasive and painless to the human body. It can directly observe the heart chambers, myocardial thickness, valve shape and activity, and heart functions on the human body. It has now become an indispensable

Table 1
Description of data attributes.

Category	Attribute	Value
Blood Biochemical Indicators	The twenty-one indicators include:APB, ApoA, ApoB, ApoE, AST, CHOL et al. (see the above for all names)	Decimal
Basic Information of Patients	Gender	Bool
	Age	Integer
	Diabetes	Bool
	Blood pressure	Numerical value
	Blood sugar	Numerical value
	Heart rate	Numerical value
	Chest pain	Bool
	City	Integer
	Family coronary heart disease history	Bool

examination method in cardiology.

The report text of cardiac color Doppler echocardiography mainly includes: the patients' basic personal information(medical card number, name, age, examination time, diagnosis number), examination name, examination route, image level, and examination items. Among them, the examination items can be subdivided into four parts: M-mode main measurement values, two-dimensional echocardiogram description, color Doppler ultrasound description, and left heart function measurement.

According to the previous data processing, the obtained data analysis results are as follows: There are total 17885 data, in which the number of occlusion cases and the percentage of occlusion cases in each branch are described in Table 2. Furthermore, in this paper, we defined that as long as the blood vessel is clogged, if value of the occlusion level is between 10% and 100%, it is regarded as a clogging case.

As we can see in Table 2, LAD is a frequently blocked blood vessel, it accounts for 52.94%, followed by RCA and LCX, accounts for 29.89% and 23.68% respectively. The other five vessels have fewer occlusions: OM accounts for 7.36%, PDA accounts for 3.94%, LM accounts for 3.54%, and PLA and LCA are rarely occluded, accounts for 1.91% and 0.04% respectively. In clinical practice, as long as the lumen area of one coronary artery is reduced by more than 50%, it will be diagnosed as coronary heart disease. It can be seen from the table that the cases of left anterior descending artery obstruction(LAD) accounted for the highest percentage of the total number. In fact, obstruction of the left anterior descending artery(LAD), right coronary artery(RCA), and left circumflex artery(LCX) is extremely fatal in medicine. The occlusion cases of the right coronary artery(RCA) and the left circumflex branch(LCX) are second only to the left anterior descending branch. In the data, it is often seen that a patient's multiple blood vessels are blocked to varying degrees. The reason is that there is a strong correlation between different blood vessels, and the blockage of one blood vessel often affects other blood vessels. The statistical results provide a basis for the ordering and transfer sequence of each task of the progressive neural network.

Table 2
Coronary Angiography data overview.

Coronary vessel name	The number of cases in which the blood vessel is blocked	The percentage of blocked cases to the rightarrow tal number of people
LM	633	3.54%
LAD	9568	52.94%
LCX	4236	23.68%
OM	1316	7.36%
RCA	5346	29.89%
LCA	7	0.04%
PLA	341	1.91%
PDA	705	3.94%

4.2. Performance evaluation

In this study, average accuracy and F1 score are used to evaluate the model. The F1 score which is calculated from the precision and recall rate is defined as follows:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{25}$$

Where P denotes the precision and R is the recall rate.

Comparing the CRMPN model in this paper with the commonly used machine learning models, including the Naive Bayesian algorithm, decision tree, single RNN, single GRU, single LSTM, bidirectional LSTM, bidirectional GRU, and so on. And comparing with the model itself, including single task model, double task model, three task model …seven task model, these models reduce the number of tasks by reducing the number of predicted vessels in each experiment. For example, the CRMPN in this paper predicts the degree of occlusion of eight coronary arteries, and the seven-task model removes the prediction of the last coronary artery LCA. Similarly, the single-task model predicts the degree of occlusion of the first coronary artery LAD. In the comparison, the average accuracy is calculated as follows: the predictive accuracy of each coronary artery in every experiment was summed and divided by the number of vessels in this experiment.

Table 3 shows the accuracy and the F1 score of CRMPN compared with other currently used machine learning algorithms and compared with the model itself by reducing the number of tasks. It can be seen from the experimental results that the average accuracy and F1 score of the CRMPN model proposed in this paper are significantly better than other commonly used machine learning algorithms. In comparison with the model itself, we found that increasing the number of tasks can improve average accuracy, and the F1 score is also generally improved. It shows that the parameter sharing between multi-tasks and the model combined with progressive networks really facilitate mutual learning between tasks. And this improved the overall performance of the model, which coincides with Sebastian's view[29]. In Table 3, the calculation time of the traditional machine learning method is shorter than that of the deep learning model but their accuracy is lower. LSTM is a variant of RNN, and its structure is more complex, so the calculation time is longer. GRU is simpler, and its calculation speed is faster than LSTM, but still slower than RNN. The bidirectional model takes longer time than the single-step model. The introduction of attention mechanism also deepens the complexity of the model and increases the computation time. Due to its complexity, the computation time of the 3-layer progressive network is longer than that of the previous models. For different multitasking models, it is worth noting that computation time does not

Table 3
Performance comparison of different methods.

Method	Average Accuracy	F1 Score
Naive Bayesian algorithm	0.7037	0.6334
Decision Tree	0.7002	0.4554
Single RNN	0.7482	0.4859
Single GRU	0.7497	0.6290
Single LSTM	0.7543	0.5951
BiGRU	0.7962	0.6369
BiLSTM	0.8095	0.6857
BiLSTM + Attention model	0.8102	0.6892
3 layers progressive networks	0.8328	0.7071
Single-task model	0.8677	0.7369
Double-task model	0.8837	0.7028
Three-task model	0.8993	0.6963
Four-task model	0.9052	0.7167
Five-task model	0.9083	0.7393
Six-task model	0.9172	0.7625
Seven-task model	0.9294	0.7518
Combined reinforcement multitask progressive networks (CRMPN)	0.9372	0.7652

increase in a multiple relationship as the number of tasks increases. The reason is that in the actual clinical use of the trained model, multiple tasks are calculated in parallel.

The model proposed in this article is to be used by doctors in clinical diagnosis after the training is completed. The model is trained on an Intel Core i9 7960X 16 cores CPU and a Nvidia Tesla K80 GPU. The whole training process takes several hours. After training, we give the model to the doctor. The doctor can directly predict the result by inputting the relevant information of the patient into the model without retraining. The model takes less than 10 min to make a prediction. The structure of the model proposed by us is relatively complex, so the speed of calculation is sacrificed. But for accurate clinical application, this kind of waiting is worthwhile and necessary.

Fig. 4 show the learning cures of our methods, including CRMPN, seven-task model, six-task model, five-task model …single-task model. With the increase of the number of iterations, both the accuracy and the F1 score are increasing.

Similar to the above method, we also evaluated the combined reinforcement multitask progressive time-series networks. We use accuracy, recall, and F1 score to evaluate the accuracy of the model. The comparison results are shown in Fig. 5.

The input data of the model is divided into three parts: Basic patient information data(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history), 21 blood test index data and color Doppler echocardiogram report text. In the basic information of patients, family coronary heart disease history usually has a greater influence on the prediction results. Usually if a patient has a family history, it will greatly increase his probability of coronary heart disease. The reason is that in the clinic, the family genetic susceptibility of coronary heart disease is relatively obvious. Among the 21 blood test index data, myocardial injury markers, especially troponin, have a greater impact on the prediction results. The detection of troponin in myocardial injury is sensitive and specific. The color Doppler echocardiography report contains m-type main measurement values, two-dimensional echocardiography description, color Doppler echocardiography description, and left heart function measurement. The structural features extracted from the report also have a certain degree of influence on the prediction results.

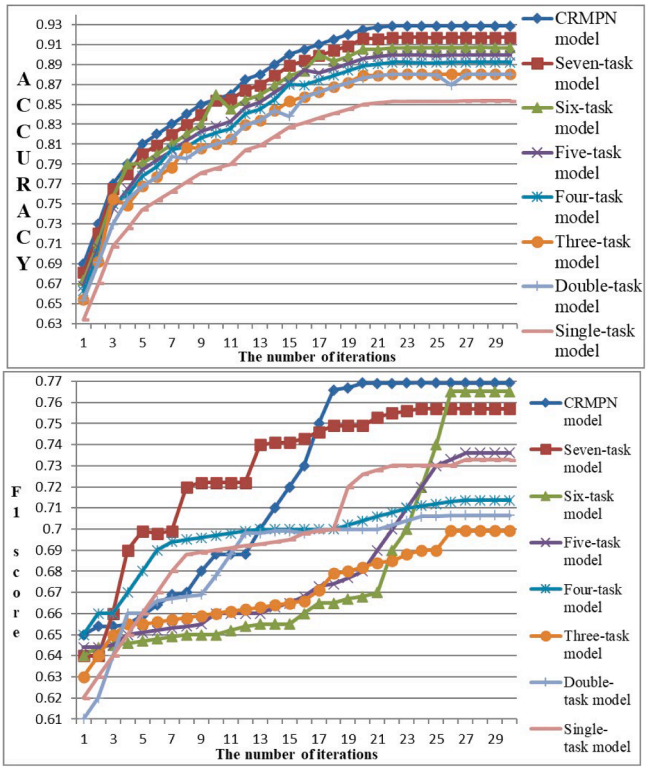

Fig. 4. The learning curve of accuracy and F1 score for models with different number of tasks.

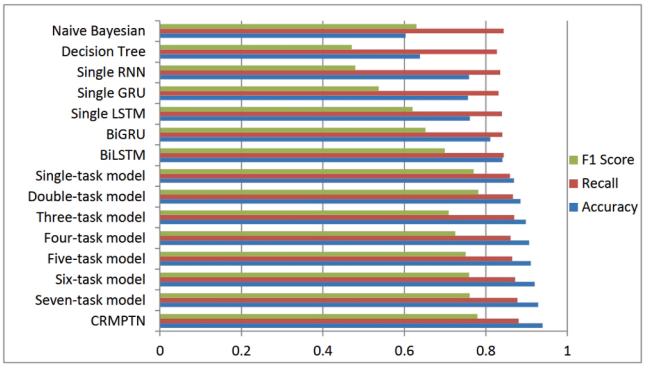

Fig. 5. Comparison of experimental results.

5. Conclusion

For the prediction of the degree of blockage of the eight coronary arteries of patients, when the input data is the patients' multiple blood test results, the color Doppler echocardiogram report text, and the patients' basic information(gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history), multi-task deep reinforcement learning can obtain higher prediction accuracy Rate, recall and F1-score. At the same time, it can be seen that for the simultaneous prediction of eight tasks, as the number of tasks increases, the prediction accuracy is also continuously improved. Using multi-task model parameter sharing, multiple models can influence each other, learn together and achieve better results.

Coronary heart disease prediction is an important and challenging issue in medical field. In this paper we propose a combined reinforcement multitask progressive time-series networks to predict the degree of coronary vascular occlusion. The model combines DRL pre-training, soft parameter sharing, hard parameter sharing and progressive time-series networks, and uses attention model, bidirectional LSTM and other modules. Every task interacts with each other and learns together to achieve satisfactory prediction results. In the future, this model can be applied to the prediction of other diseases, as well as other multi-task learning areas.

CRediT authorship contribution statement

Wenqi Li: Conceptualization, Methodology, Designing models, Writing code to implement models, Processing data, Conducting multiple sets of experiments, Writing the paper. **Ming Zuo:** Formal analysis, Data curation, Writing - review & editing. **Hongjin Zhao:** Processing data, Providing suggestions. **Qi Xu:** Validation, Supervision. **Dehua Chen:** Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2019YFE0190500.

References

[1] B. Au, U. Shaham, S. Dhruva, G. Bouras, E. Cristea, A. Lansky, A. Coppi, F. Warner, S.-X. Li, H. Krumholz, Automated characterization of stenosis in invasive coronary angiography images with convolutional neural networks, ArXiv abs/1807.10597.

[2] T. Madl, Deep neural heart rate variability analysis, ArXiv abs/1612.09205.

[3] C. Zhaojin, S. chunxian, Value analysis of different blood test indicators in patients with coronary heart disease, Integr. Cardiovasc. Med. 3 (32) (2015) 73–74.

[4] S. Hasic, E. Kiseljakovic, R. Jadric, J. Radovanovic, M. Winterhalter-Jadric, Cardiac troponin i: the gold standard in acute myocardial infarction diagnosis 3 (3) (2003) 41–44.

[5] P. Xiangdong, Z. Hua, L. Jizhong, Compressive sensing ecg reconstruction based on over-complete dictionary for body area network, Acta Automatica Sinica 40 (7) (2014) 1421–1432.

[6] D. Rs, V. Rs, P. Mj, et al., General cardiovascular risk profile for use in primary care: the framingham heart study, Circulation 117 (6) (2008) 743–753.

[7] H. Yan, J. Zheng, Y. Jiang, C. Peng, Q. Li, Development of a decision support system for heart disease diagnosis using multilayer perceptron, in: International Symposium on Circuits and Systems, Vol. 5, 2003, pp. V-V. doi:10.1109/ISCAS.2003.1206411.

[8] K. Polat, S. Sahan, S. Günes, Automatic detection of heart disease using an artificial immune recognition system (airs) with fuzzy resource allocation mechanism and k-nn (nearest neighbour) based weighting preprocessing, Expert Syst. Appl. 32 (2) (2007) 625–631, https://doi.org/10.1016/j.eswa.2006.01.027.

[9] M.G. Tsipouras, T.P. Exarchos, D.I. Fotiadis, A.P. Kotsia, K.V. Vakalis, K.K. Naka, L. K. Michalis, Automated diagnosis of coronary artery disease based on data mining and fuzzy modeling, IEEE Trans. Inf Technol. Biomed. 12 (4) (2008) 447–458, https://doi.org/10.1109/TITB.2007.907985.

[10] I. Babaoglu, O. Findik, M. Bayrak, Effects of principle component analysis on assessment of coronary artery diseases using support vector machine, Expert Syst. Appl. 37 (3) (2010) 2182–2185, https://doi.org/10.1016/j.eswa.2009.07.055.

[11] R. Alizadehsani, J. Habibi, M.J. Hosseini, H. Mashayekhi, R. Boghrati, A. Ghandeharioun, B. Bahadorian, Z.A. Sani, A data mining approach for diagnosis of coronary artery disease, Comput. Methods Programs Biomed. 111 (1) (2013) 52–61, https://doi.org/10.1016/j.cmpb.2013.03.004.

[12] R. Alizadehsani, M.H. Zangooei, M.J. Hosseini, J. Habibi, A. Khosravi, M. Roshanzamir, F. Khozeimeh, N. Sarrafzadegan, S. Nahavandi, Coronary artery disease detection using computational intelligence methods, Knowl.-Based Syst. 109 (2016) 187–197, https://doi.org/10.1016/j.knosys.2016.07.004.

[13] L. Verma, S. Srivastava, P. Negi, A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data, J. Med. Syst. 40 (2016) 1–7.

[14] G. Chalumporn, P. Kijsanayothin, R. Hewett, A sub-linear scalable mapreduce-based apriori algorithm, in: Proceedings of the 2019 3rd International Conference on Big Data Research, ICBDR 2019, 2019, p. 6–11. doi:10.1145/3372454.3372463.

[15] W. Fengli, Research on disease prediction model based on bp neural network and ds evidence theory, Master's thesis.

[16] P. Xiantao, Research and implementation of heart disease prediction based on bp neural network, Master's thesis.

[17] Z. Yue, Research on the application of svm in the classification and prediction of coronary heart disease, Master's thesis.

[18] Z. Hanqing, Differential diagnosis method for coronary heart disease based on artificial neural network, Chin. Foreign Med. Care (12) (2011) 191–192.

[19] J. Nahar, T. Imam, K.S. Tickle, Y.-P.P. Chen, Association rule mining to detect factors which contribute to heart disease in males and females, Expert Syst. Appl. 40 (4) (2013) 1086–1093, https://doi.org/10.1016/j.eswa.2012.08.028.

[20] K. Orphanoua, A. Dagliatic, L. Sacchib, A. Stassopouloue, E. Keravnoua, R. Bellazzi, Incorporating repeating temporal association rules in naïve bayes classifiers for coronary heart disease diagnosis, Biomed. Inform. 81 (2018) 74–82.

[21] M. Zreik1, R.W. van Hamersvelt, J.M. Wolterink, T. Leiner, M.A. Viergever, I. Išgum, Automatic detection and characterization of coronary artery plaque and stenosis using a recurrent convolutional neural network in coronary ct angiography, 1st Conference on Medical Imaging with Deep Learning (MIDL 2018).

[22] M.E. Taylor, P. Stone, An introduction to inter-task transfer for reinforcement learning, AI Magazine 32 (1) (2011) 15–34.

[23] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning 8 (3–4). doi:10.1007/BF00992696.

[24] J. Baxter, A bayesian/information theoretic model of learning to learn via multiple task sampling, in, Mach. Learn. (1997) 7–39.

[25] T. Schaul, D. Horgan, K. Gregor, D. Silver, Universal value function approximators, in: Proceedings of the 32nd International Conference on Machine Learning, Vol. 37 of Proceedings of Machine Learning Research, 2015, pp. 1312–1320.

[26] Shun-ichi Amari, Natural gradient works efficiently in learning, Neural Comput. 10 (2) (1998) 251–276, https://doi.org/10.1162/089976698300017746.

[27] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning.

[28] A. Levey, M. Lindenbaum, Sequential karhunen-loeve basis extraction and its application to images, IEEE Trans. Image Process. 9 (8) (2000) 1371–1374, https://doi.org/10.1109/83.855432.

[29] S. Ruder, An overview of multi-task learning in deep neural networks, ArXiv abs/1706.05098.