Feature Reinforcement using Autoencoders

Devesh Raj, Mercedes Benz Research and Development Bangalore, India

Rishabh Goyal, Mercedes Benz Research and Development Bangalore, India

Abstract——Cardiovascular disease (CVD) is the number one cause of death globally, more people die annually from CVDs than from any other cause. People with cardiovascular disease or who are at high cardiovascular risk need early detection and management using counselling and medicines, as appropriate. The early detection of CVDs needs an expert hand and awareness amongst people. Here is where Data analytics can help in predicting the cardiovascular cases before-hand by helping to make informed decisions faster, with great accuracy and at a much earlier date. The dataset used is the Cleveland Heart disease Database taken from UCI learning data set repository. The dataset is being divided into five classes, 0 corresponding to absence of any disease and 1,2,3,4 corresponding to grades of heart disease. The dataset has been bifurcated into absence (0) and presence (1, 2, 3 and 4) of the heart disease. Using medical profiles such as age, sex, blood pressure, cholesterol, sugar level etc. The classifiers can predict the probability of patients getting a heart disease. There is no dearth of classification techniques but feature engineering and data representation is the crux of the model building pre-activity. When done efficiently, this could make the model more robust and accurate. We are introducing an idea of feature reinforcement technique using Artificial Neural Networks (MLP)-Auto encoders. In this technique we would represent the features in an abstracted format using MLPAutoencoders and then reinforce the input features with the abstracted features. This activity would exhaustively capture the latency in input features thus making our feature representation more robust and resilient. We have tested our technique on Cleveland Heart disease dataset. The results obtained by using our technique had higher degree of accuracy than the results obtained with input features alone

Keywords—Deep Learning, Feature Engineering, Data Representation, Auto encoders

I. INTRODUCTION

Cardiovascular disease (CVD) is the number one cause of death globally, more people die annually from CVDs than from any other cause. An estimated 17.5 million people died from CVDs in 2012, representing 31% of all global deaths. Of these deaths, an estimated 7.4 million were due to coronary heart disease and 6.7 million were due to stroke. Over three-quarters of CVD deaths take place in low- and middle-income countries. Out of the 16 million deaths under the age of 70 due to non-communicable diseases, 37% are caused by CVDs. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established disease) need early detection and management using counselling and medicines, as appropriate. The early detection of CVDs need an expert hand and awareness amongst people. 82% of deaths due to CVDs happen in low and middle income countries who are not well equipped to handle such a crisis. Here is where Data analytics can help in predicting the cardiovascular cases before-hand by helping to make informed decisions faster, with great accuracy and at a much earlier date. There exist many techniques which gives good accuracy on this kind of classification problem. Deep Learning technique is one them. Deep Learning models are gaining more popularity than any other kind of model these days. Hence, We will talk about Multilayer Perceptron (MLP). A multilayer perceptron is a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a different mathematical function as providing a new representation of the input. The goal of a multilayer perceptron is to approximate some function f* . A feedforward network defines a mapping y = f(x; θ) and learns the value of the parameter θ that result in the best function approximation.

II. Methodology

When we need to deal with the problem of exhaustive data representation[1], first term which comes to our mind is Auto encoder[2][3][8]. Auto encoders may be thought of as being a special case of feedforward networks, and may be trained with all of the same techniques, typically minibatch gradient descent following gradients computed by back-propagation.

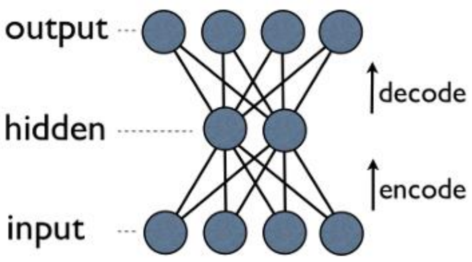

Fig. 1. Autoencoder

An auto encoder[2][3][8] is a neural network that is trained to attempt to copy its input to its output. Internally, it has a hidden layer h that describes a code used to represent the input. The network may be viewed as consisting of two parts: an encoder function h = f(x) and a decoder that produces a reconstruction r = g(h). An auto encoder[2][3][8] whose code dimension is less than the input dimension is called under complete and the one whose code dimension is greater than the input dimension is called over complete. In case of over complete Auto encoder, even a linear encoder and linear decoder can learn to copy the input to the output without learning anything useful about the data distribution. To avoid this, regularization is introduced in different forms. One of the ways of introducing regularization is the use of Denoising autoencoders. Traditionally, autoencoders minimize the function

$$L(x, g(f(x))) \tag{1}$$

where L is a loss function penalizing g(f(x)) for being dissimilar from x, such as the L2 norm of their difference. This encourages g.f to learn to be merely an identity function if they have the capacity to do so.

A Denoising auto encoder or DAE instead minimizes

$$L(x, g(f(\tilde{x}))) \tag{2}$$

where $\tilde{x}$ is a copy of x that has been corrupted by some form of noise. Denoising auto encoders must therefore undo this corruption rather than simply copying their input.

These techniques provides feature representation [5] in the form of h(x). We have come up with a technique of feature reinforcement using auto encoders which further enhances this feature representation. The reinforced features are then passed onto our classifier and then a comparative study is done amongst the results obtained from a plain vanilla classifier and our technique.

III. Data Analysis

Data cleaning was performed on the dataset by eliminating the rows having unrecorded data. Then, the categorical features were converted into binary features. For example - In our dataset, we have a feature ‘cp (chest pain)’ which can has 4 values i.e. 1 (typical angina), 2 (atypical angina), 3 (nonanginal pain) and 4 (asymptomatic). We need to convert it to binary feature to make it more suitable for use in the model. This lead to four features cp 1, cp 2, cp 3 and cp 4. So, value 0 of cp 1 denotes absence of category 1 chest pain and value 1 denotes presence of category 1 chest pain. After the data cleaning a total of 297 records with 20 medical attributes (factors, variables) were obtained from the Cleveland Heart Disease database. The attribute num i.e. diagnosis of heart disease (angiographic disease status) was identified as the predictable attribute with value 1 i.e. greater than 50% diameter narrowing for patients with heart disease and value 0 i.e. < 50% diameter narrowing for the patients with no heart disease. The attributes with description is shown in Table 1.

After that we did exploratory data analysis on the the dataset and tried to find out which features out of these 20 features are more important. For this, we used Random forest method[4] which provide us the best features on which the decision tree should be split upon. We found out that feature cp 4 (present or absence of asymptotic chest pain), oldpeak (ST depression induced by exercise relative to rest) and age were the three most important features amongst others. These three features are denoted by feature 10, 7 and 0 respectively in figure 2.

TABLE I. DATASET ATTRIBUTES

Attribute	**Description**
age	Age in years
sex	1 = male; 0 = female
cp	Chest pain type 1 – typical angina 2 – atypical angina 3 – non-anginal pain 4 – asymptomatic
trestbps	Resting blood pressure(in mm Hg on admission to the hospital)
chol	Serum cholesteral in mg/dl
fbs	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
restecg	resting electrocardiographic results 0 – normal 1 – having ST-T wave abnormality (T wave inversion and/or ST elevation or depression of > 0.05 mV) 2 – showing probable or definite left ventricular hypertrophy by Estes' criteria
thalach	Maximum heart rate achieved
exang	Exercise induced angina (1 = yes, 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	The slope of the peak exercise ST segment 1 – upsloping 2 – flat 3 – downsloping
ca	Number of major vessels (0-3) colored by flourosopy
thal	3 – normal 6 – fixed defect 7 – reversable defect
num (the predicted attribute	Diagnosis of heat disease (angiograhic disease status)
output	0: less than 50% diameter narrowing 1: greater than 50% diameter narrowing

IV. USAGE OF AUTOENCODERS IN FEATURE REINFORCEMENT

In the previous section, we described how we have processed and come up with the dataset which contains 20 features. In our experiment, we used the autoencoder whose hidden layer has more dimensions than the input layer. The above mentioned 20 features were fed as an input to an autoencoder which has 30 hidden units. We have used the hyperparameters provided in Table 2 to train our autoencoder.

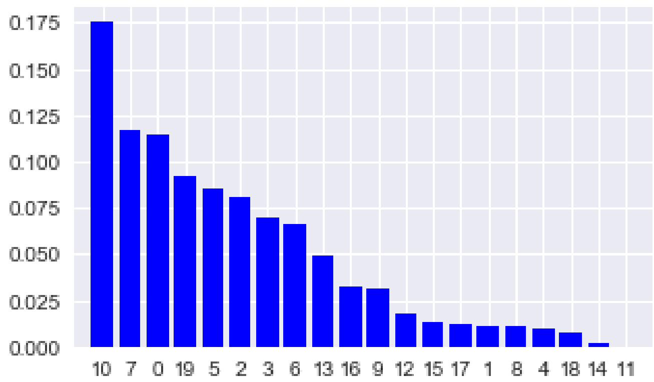

Fig. 2. Feature Importance

TABLE II. HYPERPARAMETERS

Hyperparameter	Value
Activation Function	Linear
Optimizer	Adam
Loss	Mean Absolute Error

Once we get reasonable amount of similarity between the input and the decoded features, the weights of the autoencoder were preserved for the later use. Now using the autoencoder's encoder, we converted the input features to encoded features. In our technique, we incorporated these encoded features with the input features to create a 50 dimension input space. Our approach derived its name from this reinforcement of features onto the input features. This approach fulfills our intention of representing the input data in an exhaustive feature space. The approach is represented in figure 3.

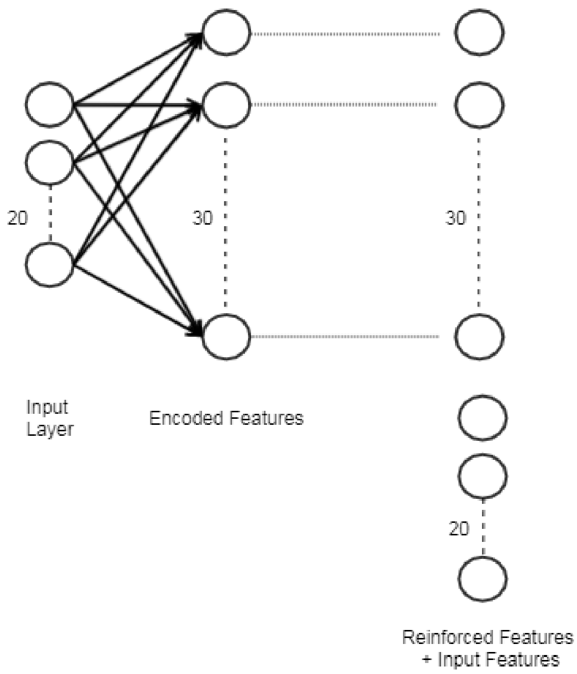

Fig 3. Feature Reinforcement

V. BUILDING THE CLASSIFIER

We experimented with two different sought of classifiers on the dataset. They are discussed below.

A. Classifier with MLP

A multilayer perceptron classifier is formed with 20 input features, 1 hidden layer and a binary output. The hidden layer consist of 10 neurons and the output layer consist of 1 neuron. If the output is near to value 0, it depicts that a person does not have heart disease while output value near 1 depicts that the person has heart disease. The hyper parameters used for the MLP are mentioned in the below table.

TABLE III. HYPER PARAMETERS

Hyperparameter	Value
Activation Function	tanh (for Hidden Layer) sigmoid (For Output Layer)
Optimizer	Adam

Hyperparameter	Value
Loss	Binary Cross Entropy Loss
Metric	Accuracy

The dataset containing 20 input features were fed to this model. The model was then trained on a batch size of 20 and 100 epochs and the results were obtained. A representation of this model can be found in figure 4.

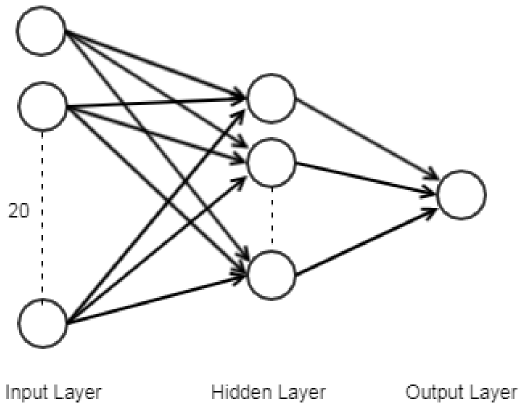

Fig 4. Multilayer Perceptron

B. Classifier with MLP using feature reinforcement

A multilayer perceptron classifier is formed with 50 input features, 1 hidden layer and a binary output. The hidden layer consist of 25 neurons and the output layer consist of 1 neuron. The dataset containing 20 input features and 30 reinforced features was fed to this MLP. The MLP is then trained on a

TABLE IV. HYPERPARAMETERS

Hyperparameter	**Value**
Activation Function	tanh (for Hidden Layer) sigmoid (For Output Layer)
Optimizer	Adam
Loss	Binary Cross Entropy Loss
Metric	Accuracy

batch size of 20 and 100 epochs and the results were obtained. A representation of this model can be found in figure 5.

VI. COMPARISON OF RESULTS

We performed comparison between the two techniques discussed in last section. Comparison is made in terms of Precision, Recall, F1 Score and Area under the Receiver Operating Characteristic Curve. Below table shows a comparison between different metrics.

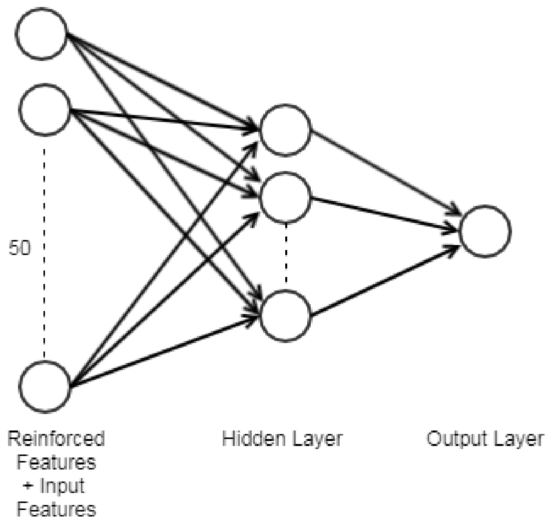

Fig 5. MLP with Feature Reinforcement

Below table shows a comparison between different metrics.

TABLE V. RESULTS OBTAINED

Model	**Precision**	**Recall**	**F1 Score**	**ROC_ AUC**

MLP without Feature Reinforcement	0.79	0.79	0.79	0.866
MLP with Feature Reinforcement	0.81	0.81	0.81	0.871

If we compare the ROC AUC curves in figure 6 and figure 7, we observe that there is an improvement in area under the curve, which shows reinforcement of features to input features improves the performance of a multilayer perceptron.

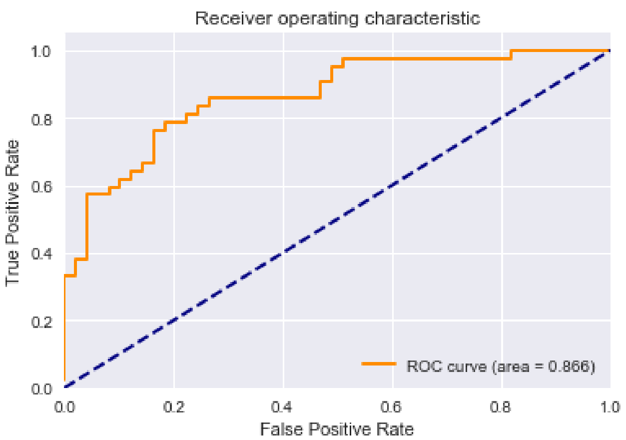

Fig 6. ROC_AUC curve MLP without Feature Reinforcement

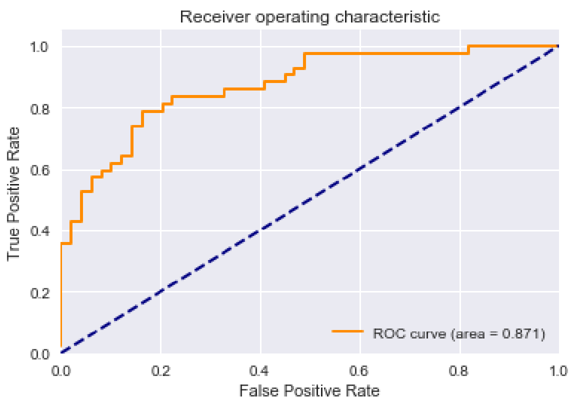

Fig 7. ROC AUC curve MLP with Feature Reinforcement

VII. Conclusion

The novel approach we followed to extract latent features using autoencoders and reinforcing those features to our input features helps us to create a robust input feature space. This approach can be used in other usecases to perform exhaustive Feature engineering which helps in better data representation as compared to conventional approaches.

References

[1] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, ”Extracting and Composing Robust Features with Denoising Autoencoders”, 25th International Conference on Machine Learning (ICML’08), pp. 1096-1103, 2008.

[2] [Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786):504507.

[3] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning Book

[4] Random Foreset –Random Features Leo Breiman Statistics Department University of California Berkeley, CA 94720 Technical Report 567

[5] Alain, G. and Bengio, Y. (2012). What regularized auto-encoders learn from the data generating distribution. Technical Report Arxiv report 1211.4246, Universite de Montreal.

[6] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011a). Contractive auto-encoders: Explicit invariance during feature extraction. In ICML2011.

[7] Weston, J., Ratle, F., and Collobert, R. (2008). Deep learning via semisupervised embedding. In ICML 2008.

[8] Representation Learning: A Review and New Perspectives. Yoshua Bengio, Aaron Courville, and Pascal Vincent Department of computer science and operations research, U. Montreal also, Canadian Institute for Advanced Research (CIFAR) 13-15

[9] Hinton, G., Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. Science, 313, 504507.