Risk Factor Analysis Based on Deep Learning Models

Qiuling Suo
Computer Science and Engineering
University at Buffalo
State University of New York
qiulings@buffalo.edu

Hongfei Xue
Computer Science and Engineering
University at Buffalo
State University of New York
hongfeix@buffalo.edu

Jing Gao
Computer Science and Engineering
University at Buffalo
State University of New York
jing@buffalo.edu

Aidong Zhang
Computer Science and Engineering
University at Buffalo
State University of New York
azhang@buffalo.edu

ABSTRACT

Accurate rendering of diagnosis and prognosis for a disease with respect to a patient requires analysis of complicated, diverse, yet correlated risk factors (RFs). Most of the existing methods for this purpose are based on handcraft RFs by calculating their statistical significance to the disease. However, such methods not only incur intensive labor but also lack capability to discover or infer previously unknown complex relationships and combined effects among correlated RFs.

Nowadays, deep learning models have emerged as a hot topic, due to its ability to automatically extract useful and complex features from raw data. In this paper, we explore the effectiveness of deep learning on medical data by building a deep learning based framework to analyze risk factors and study its prediction performance in disease diagnosis. Specifically, we investigate the application of deep learning with a special focus on interpreting the latent features extracted or created from raw data by the model. Experimental results demonstrate that deep learning based methods are able to aggregate features sharing same characteristics, and reduce effects from unimportant and uncorrelated RFs. The abstract features obtained by deep learning methods can represent the essentials of raw inputs, and give a good prediction performance in disease diagnosis.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Health; H.2.8 [**Database Management**]: Database Applications—*Data Mining*

BCB '16, October 02 - 05, 2016, Seattle, WA, USA

ACM ISBN 978-1-4503-4225-4/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2975167.2975208

General Terms

Algorithms, Experimentation

Keywords

Deep learning, Integrated features, Risk factor analysis, Osteoporosis

1. INTRODUCTION

The wealthy data being captured in the healthcare process provides unprecedented opportunities to improve disease diagnose and prevention. The Electronic Health Record (EHR), which is a longitudinal electronic record of patient health information, is a valuable source for exploratory analysis which can assist clinical and medical research. Features of the EHR data can be converted to various risk factors (RFs), such as demographics, family history, lifestyle and so on. Risk factor analysis aims to assess the effect of potential RFs to a target disease, and evaluate the risk of a patient in developing the disease. With the success of selecting informative RFs which characterize a disease, patients can avoid unnecessary tests, and change their modifiable RFs for disease control or prevention.

However, it is a challenging task to extract informative RFs and capture the disease characteristics, due to the complexity and diversity of the EHR data. The difficulty mainly lies in two aspects. Firstly, the high-dimensional features and imbalanced class distributions often restrict model performance [18]. It is essential to properly reduce the dimensionality of the feature space and maintain sufficient information for accurate classification. Secondly, it is hard to disentangle the salient integrated features from heterogeneous information. Potential RFs may not be independent but have correlations with others because of the shared reasons behind them. It is possible that a single risk factor is not important or does not have a direct causal relation to the target disease, but its combination with other factors may be the triggering or causal factor of that disease. Thus integrated features cannot be ignored in the process of feature selection.

A variety of RF analysis methods have been developed by calculating the statistical significance of each potential RF.

Feature selection techniques based on regression or kernel methods can be used to select informative RFs, so as to improve estimator's performance on high-dimensional datasets. These methods focus on ranking the available features based on the predictive power. However, these methods pay little attention to the relationships among RFs and are often limited by explaining the selected features.

While these shallow models may struggle in correlated RFs, deep learning models can often discover and disentangle latent factors. Deep learning has achieved great success in many fields. We may expect that deep learning can achieve a similar good performance in the healthcare area, with the application of discovering disease phenotyping, identifying risk factors, and disease risk prediction. Recent work [13, 16, 7] has shown the potential utilization of deep learning for healthcare data. However, unlike the image data, healthcare data is more diverse and irregular without obvious spatial or sequential structure. There are many open questions when applying neural network on the practical healthcare problems.

In this paper, the major question we want to understand is, how a neural network represents latent features in different hidden layers. Unlike some domains where semantic interpretation of intermediate or latent features that yield final results may not be important, medical researchers do like to understand the process of identifying and selecting risk factors. This required transparency and need for interpretability is what we attempt to solve in this paper. Understanding the latent features would help to study the correlations of risk factors, disentangle integrated features, and extract a good set of risk factors. As a particular disease domain within the healthcare area, we focus on osteoporosis dataset [1] and its related RFs.

Our contribution can be summarized as:

- We investigate the performance of pretrained neural networks on health datasets. To understand the highly abstract features that a neural network extracts, we visualize the contribution of risk factors to hidden units. Both of the deep belief nets and stacked auto-encoder models give some interesting patterns that contain latent information of features.
- We conclude that deep learning has the potential to analyze risk factors for osteoporosis fracture. Some of the observed RFs may be caused by the same hidden reasons and are strongly correlated. Deep learning is used to find the shared reason of correlated RFs and extract salient integrated RFs. By analyzing the hidden units of neural networks, we find out a set of informative RFs which are important to osteoporosis and are supported by previous studies.

2. RELATED WORK

There are two main types of models to tackle problems of risk factor analysis, as investigated by [16], either based on expert knowledge or handcrafted feature set. The knowledge based models usually fix a small amount of risk factors which have been validated by experts in a certain field. These models may abandon valuable information from comprehensive risk factors that are underestimated by experts. As for the handcrafted feature set based models, the informative risk factors are identified by calculating their statistical significance. The basic idea of these feature selection methods is to rank all available features based on the predictive power in a specific condition. Regression models [4, 20, 24] such as linear regression, logistic regression, Poisson regression and Cox regression, are frequently used as the assessment methods. Most of these methods consider the importance of each feature separately, but lack the ability to evaluate the integrated role of features.

Deep learning has demonstrated impressive results in various areas, especially Computer Visionand Natural Language Processing. Many experiments show that it is powerful at extracting high-level abstract features which can better represent the essence of the original data [14]. Recently, the applications of deep learning on healthcare datasets have attracted a lot of interests from researchers. Recent works [13, 16, 7] have shown the potential utilization of deep learning for healthcare datasets. These works utilize the architecture with fully connected layers, either RBM or autoencoder, for phenotype discovering and disease classification.

Despite that deep learning has demonstrated many impressive results, it is still not clear of its internal operations and how they can achieve such performance. Previous work [15, 23] has demonstrated that deep learning extracts meaningful abstract concepts from simple inputs hierarchically: pixel, intensities, edges, object parts and objects. It is interesting to understand how a deep architecture represents features for health datasets. Therefore, we apply the approaches used from image researches to study the features of health data each unit represents in hidden layers.

The study of the inner performance of neural networks usually relies on visualization of features in each layer. [10] explores the optimal input image for each unit using gradient ascent in the image space to maximize the unit's activation. However, this requires careful initialization. [23] addresses the problem of convolutional network visualization, by proposing a deconvolutional architecture (DeconvNet), which aims to project feature activations back to the input space from its output. Contemporary work of [21] demonstrates DeconvNet reconstruction is equivalent to the gradient back-propagation. From the experiment of [17] which measures unit's relevance to the predicted class, the performance changes much faster by removing units with large gradients than units with small gradients.

3. METHODOLOGY

There have been developed large numbers of different deep learning architectures for various purpose and application scenarios. Here we study the performance of two kinds of widely used neural network architectures, deep belief nets (DBN) and stacked denoising auto-encoder (SDA). Both of the two architectures are based on fully connected bipartite graphs, so that we can investigate all the potential risk factors. Some currently popular architectures, such as convolutional neural network (CNN) and recurrent neural network (RNN) which take advantage of the spatial or sequential structure of data, are not suitable for EHR datasets because EHRs may not have such spatial/sequential structure.

In this section, we first introduce the evolution of deep learning models as the preliminaries of our framework. After obtaining integrated features using the two well-trained models, we perform two tasks: latent features analysis and disease prediction. We first introduce the method to investigate the latent representation of hidden nodes. The relations between latent features and input risk factors are

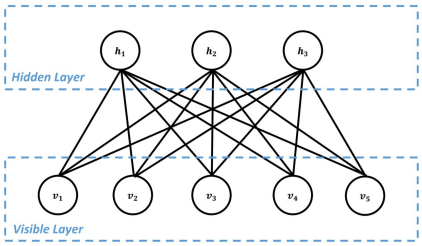


Figure 1: Shallow restricted Boltzmann machine including one visible layer and one hidden layer.

analyzed. Then, we give the training procedure by taking advantage of the labeled information for disease prediction. To explore the prediction performance using different levels of integrated features, we train the models with different numbers of hidden layers.

3.1 Preliminaries

3.1.1 Restricted Boltzmann machine

RBM [12] is a generative stochastic graphical model which learns a probability distribution over the inputs, with the restriction that its visible units and hidden units form a fully connected bipartite graph. A hidden unit h_i captures higher-order correlations of the visible units v connecting it. An illustration of RBM is shown in Figure 1. RBM investigates a representation of the input features, while requires less hidden units to represent the problem complexity. The training procedure minimizes the overall energy so that the data distribution can be well captured. The energy of a configuration of boolean vectors is defined as,

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i, \quad (1)$$

where $\theta = W, b, c$ are the model parameters, m and n are the number of visible units and hidden units. Joint probability distribution of a configuration is defined via the energy function

$$P(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} exp(-E(\mathbf{v}, \mathbf{h}; \theta)), \quad (2)$$

where $Z(\theta)$ is a normalizing factor called partition function. The marginal distribution over the visible layer $\mathbf{v}$ is:

$$p(\mathbf{v}; \theta) = \frac{1}{Z(\theta)} \sum_{h} exp(-E(\mathbf{v}, \mathbf{h}; \theta)). \quad (3)$$

3.1.2 Deep belief network

Although a shallow RBM (one layer RBM) can model some hidden causalities behind input features, there may be more reasons behind them (i.e. the reasons of reasons). To sufficiently extract high level abstract features and explore reasons behind the input features, we can stack more layers onto the shallow RBM to form a deep graphical model, namely, a deep belief network [11]. DBN is a probabilistic generative model which is composed of multiple layers of stochastic, latent variables. DBN can be trained via a greedy layer-by-layer procedure: one layer is added on top of the network at each step, and only the top layer is trained

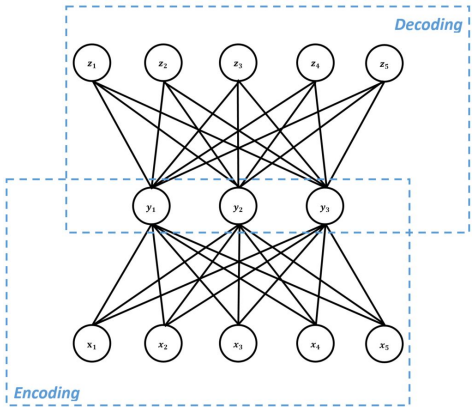


Figure 2: Autoencoder with an encoding and decoding procedure.

as a RBM using contrastive divergence (CD) strategy; after each RBM has been trained, weights are clamped and a new layer is added.

The bottom-up inference from the observed variables v and the hidden layers h_k follows a chain rule:

$$p(h_l, h_{l-1}, ..., h_l|v) = p(h_l|h_{l-1})p(h_{l-1}|h_{l-2})...p(h_1|v), \quad (4)$$

where the conditional distribution of a unit h_k in layer k can be represent by a sigmoid function of n units in layer $k-1$,

$$p(h_k|h_{k-1}) = \sigma(b_j^k + \sum_{i=1}^{n} W_{ji}^k h_i^{k-1}). \quad (5)$$

The top-down inference is a symmetric version of bottom-up inference,

$$p(h_{k-1}|h_k) = \sigma(a_i^{k-1} + \sum_{j=1}^{n} W_{ij}^k h_j^k). \quad (6)$$

3.1.3 Stacked denoising autoencoder

Given an input $\mathbf{x}$, an autoencoder constructs it to a hidden representation $\mathbf{y}$ through a deterministic mapping, which is an encoder process,

$$\mathbf{y} = \sigma(W\mathbf{x} + b). \quad (7)$$

The latent representation $\mathbf{y}$ is then mapped back into a reconstruction $\mathbf{z}$ with a decoder,

$$\mathbf{z} = \sigma(W'\mathbf{y} + b'), \quad (8)$$

where $\mathbf{z}$ is of the same shape as $\mathbf{x}$. An illustration of an autoencoder is shown in Figure 2. $\mathbf{Y}$ is expected to be a good representation of original input $\mathbf{x}$, and therefore the reconstruction error between $\mathbf{z}$ and $\mathbf{x}$ should be minimized. The reconstruction error can be measured by the squared error $L(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|^2$, or the cross entropy,

$$L_H(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{d}[x_k log z_k + (1 - x_k) log(1 - z_k)]. \quad (9)$$

The training process minimizes the reconstruction error using gradient descent. A denoising autoencoder [22] aims to discover more robust features and prevent the hidden layer from simply learning the identity. Therefore, the autoencoder is trained using a corrupted version of the input. The

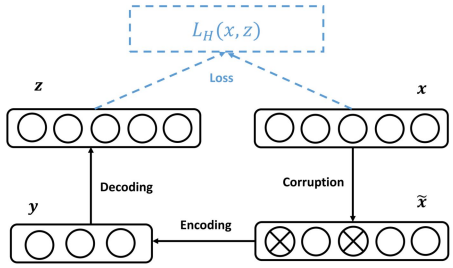


Figure 3: Denoising autoencoder with the procedure of corruption, encoding, decoding and loss measurement.

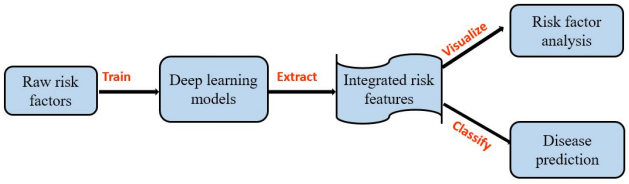


Figure 4: Pipeline of the framework. The deep learning models are trained using raw risk factors. After obtaining integrated features which form an expressive representation of the inputs, we perform two phases, feature analysis and disease prediction.

corrupted input $\tilde{\mathbf{x}}$ is mapped by a basic encoder to a hidden representation, and reconstructed to $\mathbf{z}$ by a decoder. Since $\mathbf{z}$ is a deterministic function of $\tilde{\mathbf{x}}$ rather than $\mathbf{x}$, the joint distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$ is involved in calculating the reconstruction error. An illustration of a denoising autoencoder is shown in Figure 3. Previous experiments [22] show that the denoising autoencoder has a better ability in finding interesting filters on MNIST samples. Stacked denoising autoencoder is a deep network formed by stacking several corrupted autoencoders, by feeding the output representations of one autoencoder as the input of the autoencoder on the top of the current layer. Similarly to DBN, SDA is trained via a layer-wise procedure.

3.2 Model pipeline

The pipeline of our framework includes two main components which can be described in Figure 4. The deep learning models are first trained on the original dataset of a target disease with numerous potential risk factors. Specifically, we train the model using different numbers of layers, in order to investigate the prediction performance of different levels of latent features. It is expected that higher-level latent features extracted by deep learning models can better represent the data distribution, and yield better prediction performance. A well-trained model contains multiple layers of integrated latent features. The training model can be seen in Figure 5. The input layer contains raw risk factors of a healthcare dataset, such as age, gender, medication usage and so on. The hidden layers are either formed by RBMs or autoencoders. On top of the layers, we use logistic regression to classify positive and negative samples.

The first aspect of our work is to analyze the integrated latent features, and understand how a neural network system represents risk factors for a target disease. This task is performed by visualizing and interpreting the contribu-

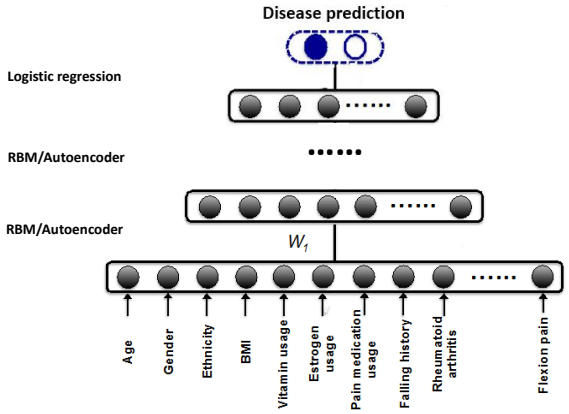


Figure 5: Deep learning Architecture. There are several hidden layers. The latent representations are trained by RBM and autoencoder separately. On the top of the layers, a classifier is added to make use of the labeled information.

tion of each input feature to the hidden units. Gradients of the output values in a hidden unit to its inputs are used to rate the importance of each input feature to a unit, by measuring how the latent features are distributed among the visible inputs. We then analyze the coherence among those risk factors which are rated as important, by looking up the categories each of them belongs to. Potential risk factors are manually separated into several categories, and risk factors in the same category are assumed to have strong correlations. Therefore, the latent features each hidden unit represents can be visualized using the visible risk factors. In this process, the category information is only used to validate the performance of the model.

The second aspect of our work aims to explore the necessity of deep learning in predicting the risk of a disease. A shallow RBM or autoencoder can get a sense of how the data is distributed, so as to capture some basic characteristics of the original data. It is expected that multi-layers can enhance this representation ability by using the optimally weighted, non-linear combination of the lower layers. We wonder if highly abstract features learned by a multi-layer model are more expressive than the integrated features learned by a single-layer model for healthcare datasets. Therefore, we train the deep learning models with different numbers of layers and compare their performance results, in order to evaluate the effect of using one and multiple layers. Moreover, we compare the pretrained neural network models with traditional classifiers trained on raw input features to evaluate the significance of deep learning.

3.3 Analyzing latent features

Latent features extracted by a hidden layer is an integrated representation of visible inputs, which captures the intrinsic characteristics of the inputs. It is known that the latent feature extracted by a hidden layer is an integrated representation of visible inputs. In image datasets, deep neural network learns abstract feature hierarchically, from pixel to edges and corners, and finally the object. We wonder how a deep neural network represent features on EHR datasets which have no spatial information.

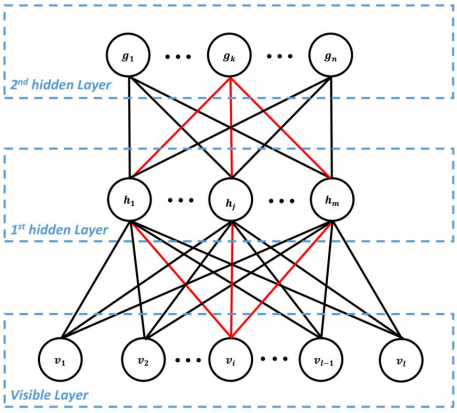


Figure 6: Connections between inputs and hidden layers. Weights involved in calculating gradient $\partial g_k/\partial v_i$ are marked red.

We investigate a latent feature by calculating the contribution of all visible risk factors connecting to it. This contribution score is measured using the gradient of a latent node with respect to each of the input features. Since a gradient may measure the change rate for a point $\mathbf{o}$ in different directions denoted as $\mathbf{x}$, if a dimension x_i has a large gradient change, $\mathbf{o}$ will be affected much more than x_i' with weak gradient. Therefore, the gradient quantitatively measures the contribution of each input feature to a latent node, which can be used for ranking and evaluating risk factors.

The gradient $\frac{\partial h_j}{\partial v_i}$ can be calculated as an intermediate step during backpropagation:

$$w_{ij} = w_{ij} + \lambda \frac{\partial J}{\partial w_{ij}}, \tag{10}$$

where w_{ij} is the weight connecting v_i and h_j, and J is the loss function measuring the error between the target and prediction. The derivative of the loss function is calculated using the chain rule [3]:

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial h_j}\frac{\partial h_j}{\partial v_i}\frac{\partial v_i}{\partial w_{ij}}, \tag{11}$$

where h_j is the activation function $h_j = f(\sum w_{ij}v_i + b_i)$ and f is the sigmoid function. Therefore, the gradient connecting layers has already been calculated in the process of learning. For a single layer model, the gradient between two layers measures the relevance of the visible layer to the hidden layer. It is derived based on applying derivative on h_j,

$$\frac{\partial h_j}{\partial v_i} = h_j(1-h_j)w_{ij}. \tag{12}$$

The relative quantity of inputs' contribution to the first layer is dominated by the weight $\mathbf{W}$. Thus we get the same results either using weight w_{ij} or gradient $\frac{\partial h_j}{\partial v_i}$. To measure the connection of higher layers with the inputs, the gradient of the lower layers should be involved, as shown in Figure 6. For a unit g_k in the second hidden layer, the contribution from v_i is measured as,

$$\frac{\partial g_k}{\partial v_i} = \sum_j \frac{\partial g_k}{\partial h_j}\frac{\partial h_j}{\partial v_i} = \sum_j g_k(1-g_k)w_{kj}h_j(1-h_j)w_{ij}. \tag{13}$$

For every hidden unit using Equation (12) or (13), we can get an importance score vector so that we can rank all the potential risk factors. The procedure of analyzing latent features is shown in Algorithm 1.

Algorithm 1 Understanding latent features.

```
train a deep learning model
for each hidden node o_i do
  for each input RF f_j do
    calculate the importance score using Equation (12) or (13)
    find out the category f_j belonging to
  end for
  visualize the score of all features
  interpret the characteristics of important features
end for
measure the coherence of latent nodes
```

To understand the relations among the important RFs extracted by hidden layers, we evaluate our models using the information already known from feature semantics. During data preprocessing, various input RFs are manually categorized into several groups based on their semantics and characteristics (see Section 4.1 for details). The importance score of input RFs can be visualized by assigning grey levels to each RF, with the darker ones for more important RFs, and the whiter for unimportant ones. We then interpret the latent features learned by deep learning through the categories of its important RFs, in order to understand the representation performance of neural network architectures. In this paper, the category information is only used for understanding latent features obtained by neural networks, and is not involved in the training procedure.

To quantitatively measure the coherence of features deep learning model extracts, we calculate the entropy for all the hidden units in one layer: firstly, for each unit, the RFs are ranked in a descending order according to their importance scores; then for the top-k features of each unit, we look up the categories each feature belongs to; after that, entropy is calculated to measure the purity of the important features for the nodes,

$$Entropy = -\sum_i^N \sum_j^{C_i} \frac{A_{ij}}{A_i} \log \frac{A_{ij}}{A_i}, \tag{14}$$

where N is the number of hidden units, C_j is the number of categories which contain any of the top-k features of unit i, A_i is the number of top scored features in unit i, and A_{ij} is the number of features in unit i which also fall into category j. Here we measure top-10 risk factors for every unit in the last hidden layer. Thus A_i is fixed to be 10 in our experiment.

The entropy is a measurement of impurity, with small entropy meaning pure system. If one unit is dominated by a diverse category distribution of RFs, which means that the important features are from different categories and there is a weak coherence of features extracted by this node, so that the entropy will be high. To evaluate that the hidden nodes tend to capture coherent RFs, we compare the entropy values of the two deep learning models and a network containing randomly selected RFs.

3.4 Disease prediction on integrated features

The training procedure to obtain integrated features and predict disease includes two stages: pretraining and finetuning. Both RBM and auto-encoder are unsupervised training procedures. This procedure aims to capture the distribution and characteristics among all the inputs. Both DBN and SDA are trained in the greedy layerwise procedure: after training the k-th layer with minimized reconstruction error, the representation of the k-th layer is used as the input for the $(k+1)$-th layer. This pre-training procedure has been shown to yield obviously better local minima than random initialization [5]. After obtaining a representative initialization point established by pre-training, a classifier such as logistic regression or SVM can be added on top of the network. We take advantage of the labeled information to train our model in a supervised fashion, naming as a fine-tuning stage. Errors between the predicted result and groundtruth are backpropagated from top to bottom and minimized through gradient descent to update model parameters to a better state.

The process of training is shown in Algorithm 2. To reduce random error introduced by the samples, 5-fold cross validation is conducted throughout our experiment. The prediction performance is measured by area under curve (AUC) of receiver operating characteristic curve (ROC) and precision-recall curve (PR). The larger AUC value indicates a better performance (an AUC of 1.0 indicates a perfect performance).

Algorithm 2 training algorithm for risk factors.

Input: All risk factors, learning rate, hidden layer structure
Output: Model parameters θ

```
Pre-training stage
Randomly initialize θ
for each layer h_l do
    RBM: run contrastive divergence to maximize likelihood
    Autoencoder: run gradient decent to minimize reconstruction error
    clamp θ_l of current layer
end for
Fine-tunning stage
for each epoch do
    for each sample do
        calculate cost c between predicted label and ground truth
        perform backpropagation to update θ
        c' is larger than c'_{-1} for d rounds
        break
    end for
end for
repeat the above for 5 times
```

Using the above training procedure, models with different numbers of layers are trained and compared. Multi-layer models are expected to have better performance than using a single layer, since high-level abstract features are often more expressible.

4. EXPERIMENTS

To analyze the performance of neural networks on a healthcare dataset, we study two neural networks on an osteoporotic fracture dataset. The neural networks in our experiments are implemented and modified from Theano [6]. Through visualizing the gradients of nodes in each hidden layer, we find that a pretrained neural network tends to aggregate features sharing same hidden reasons for the target disease, and reduces the contributions from unimportant and uncorrelated features. This behavior of neural networks can be used to select informative risk factors, and analyze the interactions among them. To quantitatively evaluate the proposed models' performance, we propose a measure of entropy and show that both models have a small entropy in comparison to the randomly generated network models, which indicates the interpretation is meaningful. Different levels of the representations of raw inputs are used to predict the risk of developing osteoporosis disease.

4.1 Dataset

The Study of Osteoporotic Fractures (SOF) [1] is the largest and most comprehensive study focused on risk factors of bone diseases. It includes 20 years prospective data about osteoporosis of 9704 Caucasian women aged 65 years and older. Potential risk factors and confounders were classified into several groups such as demographics, family history, medical history, and so on. Detailed description for each risk factor can be found from the SOF website. Potential RFs are organized into 672 variables which serve as the input of our model. The labels are processed on the Dual-energy x-ray absorptiometry (DXA) scan results on bone mineral density (BMD) measure. Based on the WHO standard, T-score of less than BMD -1.0 indicates the osteopenia condition that is the precursor to osteoporosis, which is used as the positive label. In this study, there are 5708 positive cases with osteopenia condition and 2366 negative cases with non-disease condition. The 1630 unlabeled cases are used in the unsupervised training process.

4.1.1 Data preprocessing

In order to better analyze feature relations, we adopt those features with a high coverage. Columns (features) with more than 30% missing values are removed, such that there are 411 features remain; then a column-wise mean is calculated to fill out the blank for the surviving columns.

The 411 input risk factors are not independent with each other. Some of the RFs are interacted and may have a combined effect to the disease. To evaluate the features obtained by our model, we manually categorize all the RFs into several groups based on their semantics. For example, a category *lifestyle*, contains several subcategories such as *exercise*, *drink*, *smoke*, etc.; a subcategory *drink* contains several RFs such as drink amount per week/year, alcohol increase or not, etc.; a subcategory *exercise* contains RFs such as low/medium/high intensity activity per year/at different age and so on. Features in the same category are assumed to have stronger correlations than those from different categories. Table 1 shows the categories and subcategories we have for all the RFs. This categorization information is not involved in our proposed methods, and it is only used for evaluating latent features obtained by neural networks.

4.2 Risk factor analysis

4.2.1 Analysis of latent features

The analysis of latent features aims to understand the per-

Table 1: Categories and subcategories of potential risk factors.

Categories	Subcategories
Anthropometric	weight, height, body
Demographics	district
Endpoints/Outcomes	post fracture, fracture time
Exam Bookkeeping	bone mass
Family History	father, mother
Female History	pregnant, breast cancer, menopause
Fractures and Falls History	fracture, mother fall, sister fall, fall
Lifestyle	drink, smoke, nutrition, caffeine, exercise, walk, sitting/sleeping
Medical History	diabetes
Medications	thyroid, estrogen, medication usage
Vertebral fractures	4mm vertebral fractures
Physical Function	walking difficulty, climbing difficulty, back pain, housework, IADL
Physical Performance	arm aid, strength, step, tandem, turn
Quality of Life	life quality
Visioin	eyes vision
Vital Signs	blood pressure

formance of the neural network models on the osteoporosis dataset, and identify salient integrated RFs that are important to the disease. Although we are not dealing with image dataset which has a clear spatial message, the features of EHR data can be visualized based on their importance score. As discussed in Section 3.3, the importance score of a RF is the gradient of the latent representation in a hidden unit with respect to this RF.

In this section, we demonstrate the gradients of several hidden units in terms of the input features, in order to understand the information of latent features in each unit. The gradients are calculated following the approach described in Section 3.3. For nodes in the first hidden layer, either weights or gradients can be used to visualize the importance of inputs. For nodes in the second layer, gradients connecting each node and inputs are calculated via the chain rule, combining the non-linear relations from the two layers. Since the RFs may have positive or negative effects on the disease, we rank the absolute value of gradients to measure the importance of the corresponding RFs.

Figure 7 shows the visualization of feature importance to one node. The y-axis shows the names of different subcategories, and x-axis is the feature index in each subcategory. The scores are normalized to [0,1]. Since the numbers of features in different subcategories are not the same, vacant positions are filled with 0. In the figure, each grid stands for the score of gradient connecting with one input feature. For example, *weight* subcategory contains 6 RFs which correspond to the first 6 grids in the x-direction with y='*weight*', and the remaining grids in this line are directly filled by 0; *Drink* subcategory contains 22 RFs so that the first 22 grids with y='*drink*' are not directly filled by 0. The grey level of each grid indicates the importance of an RF to a given node: the darker grid means larger gradient which is more important, and the whiter one is less important. Therefore, we can see that risk factors related to *exercise* contribute much to this hidden node. In other words, this node collects information from the *exercise* subcategory and reduces the impact from other subcategories. From Figure 7, RFs

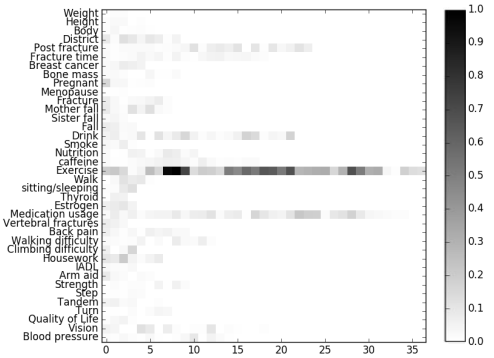


Figure 7: Visualization of the importance scores in one internal node in the first layer.

Table 2: Informative RFs identified by a hidden node.

Index	Variable	Description
7	PACT50	participate physical activity at age 50?
8	PACT30	participate physical activity at 30?
9	PACTTA	participate physical activity in teenager?
18	L30INT	times of low intensity activity at 30
19	M30INT	times of medium intensity activity at 30
21	TTOT30	times of any intensity activity at 30
28	30TMWT	activity times weighted by intensity at 30

in *exercise* indexed 7, 8, 9, 18, 19, 21, 28 contribute much to this hidden node. The important risk factors identified by this node are shown in Table 2 with their indexes in the category, variable names and semantics descriptions. From variable descriptions in Table 2, we see that these variables are closely related in semantics, and they are all important to the hidden node. Therefore, we conclude that this node identifies the informative risk factors related to *excercise*.

Figure 8 shows the visualization of selected hidden nodes in the two hidden layers of DBN. Each subfigure contains the gradient values of the input RFs to one node. The y-

Table 3: Coherence measurement.

	DBN	SDA	random
entropy	12.935	11.018	26.193

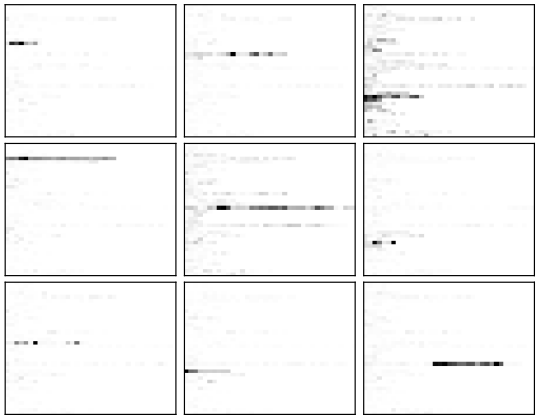
axis of these subfigures is the same as Figure 7, without the semantics for simplicity; the x-axis is also the same, which contains the indexes of risk factors in one subcategory. In Figure 8(a) which visualizes the first layer, each node seems to be "dominated" by a subcategory: features that are important to a node come out to fall into the same subcategory. For example, the important subcategories aggregated by each hidden node are: mother fall (mother ever break or fracture a bone), drink, walking difficulty, post fracture (any fracture post current visit), exercise, housework (difficulty of doing housework), caffeine, back pain, and medication usage. For the second layer in Figure 8(b), we calculate the gradient value using Equation 13. After visualizing each feature's score (the gradient value)in different subcategories, we also check which category they belong to according to Table 1. Therefore, we can find out that the important categories to the second layer are: physical function, lifestyle, endpoints/outcomes, and fracture and fall history.

From the above results, we see that the units in the neural network do select some informative RFs for osteoporosis. Based on the universal rule used by RFAX [2] which is a fracture risk assessment tool developed by WHO, some risk factors such as previous fracture, alcohol intake, and family history have already been shown to link to bone fracture risk. Besides, some of the physical and lifestyle features such as exercise, walking difficulty, and caffeine are examined as risk factors for osteoporotic fractures [9, 8]. As can be seen, these risk factors are selected by the proposed models.

Visualization for nodes in SDA is shown in Figure 9. From Figure 9(a), we find that the important features to nodes in the first layer are: drink, mother fall, walking/climbing difficulty/IADL (total difficulty), exercise, post fracture, back pain, fracture and post fracture, medication usage, and caffeine. Using the same way for DBN, we can see that Figure 9(b) indicates the important categories for the second layer, including endpoints/outcomes, lifestyle, fracture and fall history, and physical function.

Figure 8 and 9 give the visualization of RF contribution for selected hidden nodes. There are totally 100 nodes in the first hidden layer and 10 nodes in the second hidden layer. Other nodes not shown in the figures either have similar patterns or identify the same important categories as those listed in the figures, and also, there are some nodes which do not show this sparse distribution of importance scores. Although the two deep learning models, DBN and SDA give different visualization patterns, the informative feature categories they select are almost the same, and match some previous studies.

The quantitatively measurement of our models are shown in Table 3. We use information entropy as defined in Equation 14 to calculate the coherence of the latent features extracted by neural networks. Comparing to randomly scattering RFs into nodes in a network, deep learning models have a smaller entropy, which means that deep learning models tend to aggregate features with some similar characteristics rather than distributing features randomly among nodes.



(a) Nodes in the first layer

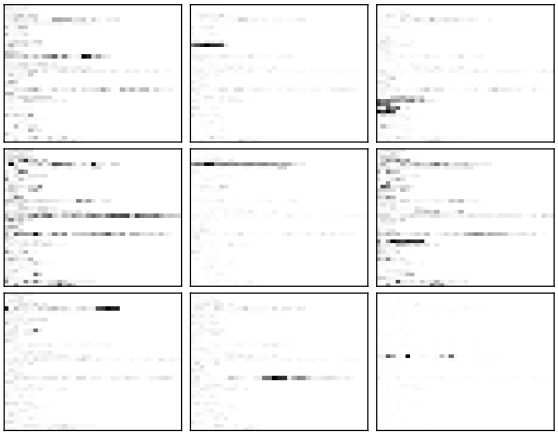
(b) Nodes in the second layer

Figure 8: Visualization of importance scores of some nodes in the first and second layer of DBN.

4.3 Osteoporosis prediction

The results for different data representations on the osteoporosis dataset are shown in Table 4. The performance of applying SVM on raw features and pretrained neural networks with different layer numbers are compared. As introduced in Section 3.4, we first use a shallow RBM/autoencoder and two-layer DBN/SDA for unsupervised training, then add labeled information to the four models for supervised training. The SVM classifier is implemented using Python scikit–learn library [19]. Area under ROC and PR curve are used to measure the prediction performance of models.

From Table 4, we can draw the following conclusions: (1) RBM-based and SDA-based methods give pretty good performances. This is due to the fact that these methods can automatically learn high-level feature representations from raw risk factors by leveraging their deep structure. (2) Both the 2-layer RBM and SDA methods improve the performance of corresponding single-layer methods. This benefit is brought by the ability of these methods to learn high-level relations layer-by-layer among raw risk factors. These experimental results also confirm the analysis of latent features in



(a) Nodes in the first layer

(b) Nodes in the second layer

Figure 9: Visualization of importance scores of some nodes in the first and second layer of SDA.

the previous section. (3) The two-layer neural network models outperform SVM which is applied on the raw input RFs. This shows that a pretrained neural network can better understand the hidden relations among input RFs, and better represent the information embedded in the EHR dataset.

In this section, we observe that deep learning methods improve the predication accuracy on osteoporosis data. In practice, healthcare datasets can contain more complicated and unstructured information in forms of text, image and voice. In the future, we plan to apply the proposed methods to these more complex datasets.

5. CONCLUSION AND FUTURE WORK

It is a challenging task to analyze the complicated and highly correlated relationships among numerous potential risk factors. Existing approaches on handcraft RFs usually focus on the prediction performance of different RFs, but ignore the interactions and synthetic roles of them. In this paper, we propose to use deep learning models to analyze risk factors. Two widely used deep architectures, DBN and SDA are studied in our experiments. The two models show

Table 4: Prediction results.

	AUC-ROC	AUC-PR
SVM	0.846	0.928
shallow RBM	0.841	0.921
2-layer DBN	**0.865**	**0.936**
shallow autoencoder	0.846	0.926
2-layer SDA	**0.864**	**0.935**

that different deep learning models share general characteristics on healthcare datasets.

There are many open questions when applying deep learning to EHR datasets. One interesting question related to risk factor analysis is how a deep architecture represents input RFs in each layer. Therefore, we investigate the internal performance of neural networks, in order to understand the information contained in latent features. The composition of a latent feature is visualized using the importance score which measures each RF's contribution to the latent node. Through visualizing hidden nodes in different layers, we find that both of the two deep learning models tend to aggregate correlated informative RFs while reducing the effect from other noisy RFs. The informative RF categories represented by hidden nodes are endorsed as important by previous medical studies. Both the hidden unit visualization and entropy measurement indicate that deep learning models do not learn healthcare features randomly, but have an ability to redistribute features and extract high-level representation for the inputs.

We have also trained models with one and two layers, and compared their performance in disease classification. Multilayer models are more expressive and high-level abstract features can better represent the essentials of raw inputs, yielding better performance. The performance of applying SVM on raw RFs is also compared with results from DBN and SDA. It shows that features extracted by a pretrained neural network can better represent the information embedded in the EHRs, and give more effective predictions of osteoporosis disease.

In the future work, we plan to further study and develop the neural network models with the application in healthcare area. One application is to investigate deep learning for multiple diseases prediction. Different diseases may share some joint reasons, and deep neural networks can often discover and disentangle these latent reasons. It is an encouraging and challenging task to build a disease phenotype base that can better represent multiple diseases. Moreover, deep learning models can be further developed for healthcare applications by incorporating the external knowledge from experts. It is expected that the existing medical knowledge can help deep learning models to better diagnose certain diseases.

6. ACKNOWLEDGMENTS

This work was sponsored in part by US National Science Foundation under grant IIS-1514204.

7. REFERENCES

[1] http://www.sof.ucsf.edu/interface/.
[2] http://www.shef.ac.uk/FRAX/tool.jsp/.
[3] Y. Anzai. *Pattern Recognition and Machine Learning.* 2012.

[4] R. Bender. *Cancer Epidemiology*. 2009.

[5] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 2007.

[6] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, 2010.

[7] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[8] C. Cooper, E. J. Atkinson, H. W. Wahner, W. M. O'Fallon, B. L. Riggs, H. L. Judd, and L. J. Melton. Is caffeine consumption a risk factor for osteoporosis? *Journal of Bone and Mineral Research*, 1992.

[9] S. R. Cummings, M. C. Nevitt, W. S. Browner, K. Stone, K. M. Fox, K. E. Ensrud, J. Cauley, D. Black, and T. M. Vogt. Risk factors for hip fracture in white women. *New England journal of medicine*, 1995.

[10] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 2009.

[11] G. E. Hinton. Deep belief networks. *Scholarpedia*, 2009.

[12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[13] T. A. Lasko, J. C. Denny, and M. A. Levy. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one*, 2013.

[14] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[15] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[16] H. Li, X. Li, M. Ramanathan, and A. Zhang. Identifying informative risk factors and predicting bone disease progression via deep belief networks. *Methods*, 2014.

[17] H. Z. Lo and W. Ding. Understanding deep networks with gradients. *neural networks*, page 5.

[18] L. Lusa and R. Blagus. The class-imbalance problem for high-dimensional class prediction. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, 2012.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 2011.

[20] J. Robbins, A. Schott, P. Garnero, P. Delmas, D. Hans, and P. Meunier. Risk factors for hip fracture in women with high bmd: Epidos study. *Osteoporosis international*, 2005.

[21] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 2008.

[23] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*. 2014.

[24] J. Zhou, J. Sun, Y. Liu, J. Hu, and J. Ye. Patient risk prediction model via top-k stability selection. In *In Proceedings of the 13th SIAM International Conference on Data Mining*, 2013.