The Title of Your Book

Your Name
Department of Computer Science
Brown University

August 15, 2022

Contents

Preface

Which labor market institutions worked better in containing job losses during the Great Recession of 2008–2009? Is it good for employment to increase the progressiveness of taxation? Does it make sense to contrast “active” and “passive” labor market policies? Who actually gains and who loses from employment protection legislation? Why are minimum wages generally diversified by age? Is it better to have decentralized or centralized bargaining systems in monetary unions? Should migrants have access to welfare benefits? Should governments regulate working hours? And can equal opportunity legislation reduce discrimination against women or minority groups in the labor market?

Current labor economics textbooks neglect these relevant policy issues. In spite of significant progress in analyzing the costs and benefits of labor market institutions, these textbooks have a setup that relegates institutions to the last paragraph of chapters or to a final institutional chapter. Typically a book begins by characterizing labor supply (including human capital theory), labor demand, and the competitive equilibrium at the intersection of the two curves; it subsequently addresses such topics as wage formation and unions, compensating wage differentials, and unemployment without a proper institutional framework. There is little information concerning labor market institutions and labor market policies. Usually labor market policies are mentioned only every now and then, and labor market institutions are often not treated in a systematic way. When attention is given to these institutions, reference is generally made to the U.S. institutional landscape and to competitive labor markets in which, by definition, any type of policy measure is distortionary.

Acknowledgments

Your acknowledgments are included in the Preface as the final section.

Your Name
March 2014

1 Artificial Intelligence for S-mart Healthcare

Zhenjie Yao, Yixin Chen

1.1 Introduction

In the big data era, data has become more important than ever before. Meanwhile, rapid information technology improvement facilitates data processing. This trend is particularly obvious in the field of healthcare. Conventional data in heahthcare are all collected, standardized and stored in information system like HIS (Hospital Information System), LIS (laboratory information system), PACS (picture archiving and communication systems), etc., including demographic information, medical records, medical image, lab tests, medications, procedures and diagnosis, etc. Novel technologies such as wearable device, medical website, drug discovery and genetic testing enrich the categorization, scope and amount of the healthcare data. Intelligent analysis of these data is helpful for diagnosis, treatment, decision making support, prescription, disease prediction, curative effect evaluation, etc. It is an urgent task to extract valuable knowledge from the massive data in healthcare.

Deep learning (DL) can be traced back to around 2000. In recent years, deep learning outperforms traditional machine learning methods with a significant margin in image recognition, speech recognition and natural language processing [LeCun et al. 2015]. The great success demonstrated its strong ability of modeling sophisticated data. Nowadays, deep learning is applied to more and more fields, and has obtained many encouraging results. It has been shown that healthcare is one of the most promising directions.

Applying deep learning in healthcare results in many surprising works. The success of deep learning in the field of computer vision and image processing can be extended to the field of medical image processing directly. It is used for imaging, image segmentation, image recognition, lesion detection etc. This inspired lots of works along the direction of medical image processing; hundreds of papers were published in the last 3-5 years. Litjens et al. [Litjens et al. 2017] and Greenspan et al. [Greenspan et al. 2016] already gave substantial reviews on deep learning in medical image analysis. As mentioned above, health-care is more than medical image processing with a number of other aspects, therefore many researchers applied deep learning to analyze other types of data, and obtained significant improvements. This chapter focuses on health-care areas other than medical image processing. Yao et al. give a comprehensive survey of early deep learning applications on healthcare [Yao et al. 2018].

We would introduce more latest works, such as applications of some novel neural networks, it is expected that this article could help the readers understand the general situation of this field.

The remainder of this paper is organized as follows. Section 1.2 overviews commonly used deep learning algorithms, including their benefits and drawbacks. Section 1.3 reviews the relevant applications in healthcare in detail. Section 1.4 introduce our work, which apply deep learning for liver disease screening. Section 1.5 shows the experimental results. Section 1.6 analyzes the limitations and challenges in this area. Finally, the conclusion of this study is drawn and the prospect of further development is discussed in Section 1.7.

1.2 Deep learning

Neural Networks (NN) can be traced back to 1940s, and was first implemented as a perceptron in 1950s [Rosenblatt 1957], which is a bionic inspired linear classifier. Perceptron started the first wave on NN research. However, early neural networks did not achieve very good performance, and it was found that perceptron has its own limitations. Research works on NN were revived when the Multi-Layer Perceptron (MLP) was designed and trained with the Back Propagation (BP) algorithm in the 80s of the last century [Rumelhart et al. 1988, Werbos 1974]. These works gave rise to the second wave of NN research. Then, a NN usually had 3 layers, which were input layer, output layer and 1 hidden layer, it was referred to as the shallow model. Although these works did improve the model performance, NN has been shadowed by other machine learning methods represented by Support Vector Machine (SVM) in 1990s. Many factors kept NN from becoming deep architectures as we have today.

In fact, restrictions on the development of the neural network is not only the model itself, but also the limitations of hardware, such as memory capacity and computing power, and the volume of the training dataset. With rapid development of hardware, things changed gradually. Lecun et al. developed the famous convolutional neural networks with multiple hidden layers [LeCun et al. 1998]. Hinton et al. [Hinton et al. 2006] formally proposed the concept and method of deep learning. It was found that training a deep neural network is feasible. Meanwhile, rapid development of internet and mobile internet facilitated the collection of large dataset. The third wave of NN research has been set up, and its development has been rapid and enormous. Deep learning, which refers to NN with more than two hidden layers, got great success in computer vision and speech applications [Hinton et al. 2012, Krizhevsky et al. 2012, LeCun et al. 2015]. In the past 6 years, a variety of deep NN were developed and applied in various domains. In this paper, we group them into 5 categories, and introduce them one by one. Details about these deep NN could be found in [LeCun et al. 2015, Ravì et al. 2017].

Deep Neural Networks (DNN). Generally, all the 5 kinds of deep learning models can be called deep neural networks. For clarity, in this paper, deep NN refers to all kinds of deep learning models. DNN specifically refers to the basic structure, which is the conventional NN

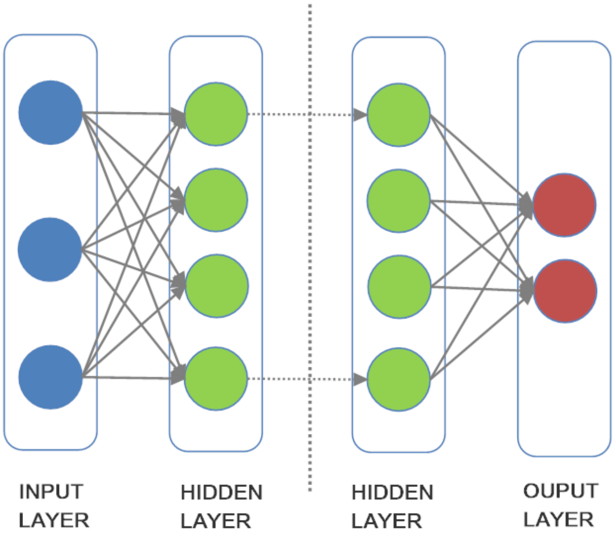


Figure 1.1: Architecture of DNN.

with more than two hidden layers, as shown in Fig.1. Training of DNN is difficult. Moreover, DNN is prone to suffer the problem of vanishing of the gradient, which means that in the BP training process, gradients (or errors) become negligible when it reaches the first several layers through many backward layers. Solving these problems led to the booming of the deep learning, and various kinds of deep NN.

AutoEncoder networks (AE). AutoEncoder is an unsupervised model. The output layer of AE is not the label, but the input itself or its noisy version. Training of AE is to reconstruct the original input data. From the viewpoint of information theory, reconstructing original data means no loss of information. In other words, training of AE is searching the best coder to minimize information loss. The coder is the compressed data with least information loss. It is an optimal representation of original data. Because the coder is with low dimension and discriminative, AE is well suited for feature extraction. Some researchers used AE to extract features from various kinds of healthcare data [Miotto et al. 2016][Danaee et al. 2016][Al Rahhal et al. 2016][Xu et al. 2017].

Restricted Boltzmann Machine (RBM) related Networks (DBN and DBM). Restricted Boltzmann machine [Hinton and Sejnowski 1986] is a variant of Boltzmann machine, which mimics the conception of statistical thermodynamics, the neural variables range from 0 to 1. Restriction on Boltzmann machine leads to a bipartite graph structure. Restriction simplifies the network structure, which makes the training procedure of RBM more tractable than Boltzmann machine. Two kinds of deep NN frameworks can be derived from RBM, Deep

Belief Networks (DBN) [Hinton et al. 2006] and Deep Boltzmann Machines [Salakhutdinov and Hinton 2009]. Both DBN and DBM are initialized by layer-wise greedy training of RBM, and fine-tuned by target labels.

Convolutional Neural Networks (CNN). CNN consists of interleaved convolution layer and pooling (subsample) layer, which is robust to shifting, rotation and scaling [LeCun et al. 1998]. CNN is the most commonly used deep learning model. Its success on ImageNet [Krizhevsky et al. 2012] is the tipping point of deep learning research boom. Although CNN is designed for image recognition [LeCun et al. 1998], it can be used on any fixed, ordered, and location related data. For example, 1-d time sequence with fixed length, it is also convenient to extend CNN to 3-d video.

Recurrent Neural Networks (RNN). Generally speaking, all the networks mentioned above are used as static models with fixed inputs. RNN [Williams and Zipser 1989] is a dynamic model, whose output is determined not only by the current states; but also by the previous state. RNN suffers the same problem as other NNs, which is vanishing of the gradients. Furthermore, besides vanishing layer by layer, gradients of RNN also vanish along time. To overcome this problem, Hochreiter and Schmidhuber [Hochreiter and Schmidhuber 1997] replaced the node of RNN by Long short-term memory (LSTM). Chung et al. [Chung et al. 2014] proposed a simpler Gated Recurrent Unit (GRU), which get similar performance as LSTM. RNN got success in many tasks for sequential data processing, such as language modeling for speech and text [Hinton et al. 2012].

Attention. Attention in neural network is to pick important information from the features, which help the neural network to pay more attention to important parts of data. The core of attention mechanism is weighting, and there are many ways to calculate the weights. Attention mechanism can form a complete network, such as transformer in machine translation [Vaswani et al. 2017]. More often, attention serves as a part of other neural network to improve its performance.

Generative Adversarial Network (GAN). GAN mainly includes two components: a generative models named generator, and a discriminant model called discriminator. The two parts are trained in an adversarial way, that is, the goal of the generator is to confuse the fake with the real, while the goal of the discriminator is to find the fakes. Specifically, GAN learns the generative model of data distribution through adversarial training [Goodfellow et al. 2014].

Graph Neural Networks (GNN). GNN are used for graph data modeling, which focused on finding node representation by information aggregation from neighborhoods in an iterative manner [Bruna et al. 2013][Hamilton et al. 2017]. GNN is good at modeling networks with complex relationships. GNN has gained a great success on tasks such as graph classification, node classification and link prediction etc., and has been applied to recommend system [Ying et al. 2018], computer vision [Yan et al. 2018], and internet traffic forecasting [Yao et al. 2021].

1.3 Healthcare Applications

In this section, we introduce various applications of deep learning in healthcare.

Electronic Health Records (EHR), is the systematized collection of patient-centered health information in a digital format, which may include a range of health related data. Mehrabi et al. [Mehrabi et al. 2015] attempted to discover the temporal pattern and association rule of the diagnosis codes. The authors modeled each patient's records as a matrix of temporal clinical events with International Classification of Diseases - 9th Version (ICD-9) diagnosis codes as rows and years of diagnosis as columns. A deep Boltzmann machine network with three hidden layers was constructed with each patient's diagnosis matrix values as visible nodes. Miotto et al. [Miotto et al. 2016] proposed an unsupervised representation of patient from the EHRs, named Deep Patient, which is a three-layer SDAE. This representation was evaluated by assessing the probability of patients to develop various diseases. Test results on a dataset composed of 76,214 test patients validated the effectiveness of this representation. The Area Under Curve (AUC) of Deep Patient is 0.773, compared to that of the best conventional representation is 0.695. Rajkomar et al. [Rajkomar et al. 2018] analyzed the EHR extensively. First, the EHR data is converted to events in temporal order, which served as the input of deep learning model for 4 tasks. The 4 tasks are inpatient mortality, 30-day unplanned readmission, length of Stay and discharge diagnosis. Three different network architectures were designed, and the final model is an ensemble of predictions. The baselines are enhanced version of conventional scores, which are commonly used in hospitals. AUC of all the 4 tasks are improved consistently. Yao et al. [Yao et al. 2020] used densely connected neural network for liver disease screening, which takes the demographics and lab test results from EHR as input, and assess the likelihood of liver disease. Choi et al. [Choi et al. 2020] designed a Graph Convolutional Transformer to learn the hidden structure of EHR while performing supervised prediction tasks on EHR data. The proposed model consistently outperformed previous approaches for various prediction tasks, indicating that it can serve as an effective general-purpose representation learning algorithm for EHR data. Dernoncourt et al. [Dernoncourt et al. 2017] designed the first de-identification system based on a deep artificial neural network with LSTM as element. The model was tested on two largest public available de-identification datasets. This model outperforms the state-of-the-art systems. It yielded an F1-score of 99.23% on the MIMIC de-identification dataset, with a recall of 99.25% and a precision of 99.06%. Torfi and Fox [Torfi and Fox 2020] built a correlation-capturing Generative Adversarial Network (corGAN) to generate synthetic healthcare records, which capture the correlations between adjacent medical features in the data representation space by combining Convolutional Generative Adversarial Networks and Convolutional Autoencoders. For various Machine Learning settings such as classification and prediction, the synthetic data show similar performance with real data.

Time series such as Electrocardiography (ECG) and Electroencephalogram (EEG) are noninvasive recording of the electrical activity of the body, they are time series. Yao et al. [Yao et al. 2017] designed a Multi-scale Convolutional Neural Networks (MCNN), which is the first deep learning model for atrial fibrillation detection. First, R wave was detected from ECG, and RR (R wave to R wave) interval sequence was the input of the MCNN for atrial fibrillation detection. The algorithm was tested on both public and private datasets. They got the best detection performance on the public dataset. Rajpurkar et al. [Rajpurkar et al. 2017] collected the largest ECG dataset. 64,121 ECGs were recorded by a single lead wearable monitor from 29,163 patients. A ResNet with 34 layers was built. 14 kinds of rhythms (13 arrhythmia and 1 normal) were recognized. They got cardiologist-level arrhythmia detection performance. Wulan et al. [Wulan et al. 2020] introduced three deep learning models for the generation of ECG signals. They also provide an quantitative approach to evaluate the performance of ECG generative models. Schirrmeister et al. [Schirrmeister et al. 2017] studied deep convolutional neural networks with a range of different architectures, designed for decoding imagined or executed movements from raw EEG. The test results on recognizing 4 movements showed that the CNN methods surpass that of the widely-used filter bank common spatial patterns (FBCSP) decoding algorithm. Liu et al. [Liu et al. 2021] adopt three-dimension convolution attention neural network (3DCANN) for EEG emotion recognition, which is composed of spatio-temporal feature extraction module and EEG channel attention weight learning module. 3DCANN can extract the dynamic relation well among multi-channel EEG signals and the internal spatial relation of multi-channel EEG signals during continuous time period.

Social media data record the activity on internet, which is helpful for monitoring of the mental health status and Adverse Drug Reactions (ADRs). Benton et al. [Benton et al. 2017] developed a deep neural Multi-Task Learning (MTL) model for 10 prediction tasks (suicide, seven mental health conditions, neurotypicality, and gender), which map the Twitter data to each kind of problem. The MTL model is compared with Single Task Learning (STL) models. Results showed that an MTL model performs significantly better than other models for all the tasks. For example, the AUC for suicide is 0.848. Dong et al. [Dong et al. 2021] developed an end-to-end graph neural network algorithm called semi-supervised Graph Instance Transformer (SS-GIT) based on multiple instance learning and contrastive self-supervised learning to predict early signs of generalized anxiety disorder and depression under the weak supervisions. Empirical results on a mobile sensing dataset demonstrate that the proposed model outperforms the best baseline model by 6.7% on ROC-AUC. Zhang et al. [Zhang et al. 2021] presented an Adversarial Neural Network with Sentiment-aware Attention (ANNSA) model for detecting ADRs. A sentiment-aware attention mechanism is proposed to extract the word-level sentiment features associated with sentiment words and learn task-related information by optimizing a task-specific loss. It is found that the word-level sentiment features are helpful in detecting ADRs from social media data. Social media is a novel field for healthcare, which is a powerful supplement of traditional healthcare data.

Wearable devices are smart electronic devices worn on the body, which can capture data consecutively. Ravi et al. [Ravi et al. 2016] presented a human activity recognition technique based on deep learning methodology, which is designed to enable accurate and real-time classification for low-power wearable devices. To reduce the computation demands, a CNN with constraints was adopted to analyze the spectrum for activity recognition. Aliamiri et al. [Aliamiri and Shen 2018] utilized the wearable device with build-in photo-plethysmography (PPG) sensor to provide a portable, non-intrusive and low-cost solution for AF monitoring and detection. An end-to-end deep learning AF detection system was built based on CNN, which can filter out poor quality signals and make reliable AF detection. Zhang et al. [Zhang et al. 2017] proposed a DL framework adopting sparse auto-encoder (SAE) to extract emotion-related features, and logistic regression for emotion recognition. Only respiration data collected from wearable devices was used for recognizing human emotions. The accuracy was about 80%. A key limitation is that DL model is computation-intensive, which can hardly work for wearable devices with limited battery, memory and computational capacity. Tradeoff between resources and performance is necessary.

Drug & compound can be inspected by deep learning based molecular structure analysis. Unterthiner et al. [Unterthiner et al. 2015] built a system named DeepTox, which normalized the chemical representations of the compounds. A large number of chemical descriptors are used as input to machine learning methods. DNN with 5 layers was used to predict the toxicity of the compounds. Test results on a dataset of 12,000 environmental chemicals and drugs show that Deeptox outperformed conventional methods on 12 different toxic effects inspection. Cheung et al. [Cheung and Moura 2020] predicting desirable molecular properties for drugs that can inhibit SARS-CoV-2. GNN is adopt to model the topologies of the molecules. Preliminary results on two COVID-19 related datasets are encouraging. Li et al. [Li et al. 2021] proposed Drug3D-Net, a grid-based 3D convolutional neural network with spatial-temporal gated attention module to model the spatial geometric structure of molecules for predicting molecular properties, which extracts the geometric features for molecular prediction tasks in the process of convolution.

Genomics analysis is employed for cancer detection and survival prediction. Kumardeep et al. [Chaudhary et al. 2017] studied survival expectations among different subgroups of hepatocellular carcinoma (HCC) by integrating multi-omics data of various patient cohorts. They built a deep learning model (autoencoder) by training 360 HCC patients' data gathered from TCGA (The Cancer Genomic Atlas) database. They found that mutations of $TP53$, $KRT19$, $EPCAM$ and $BIRC5$ genes, and activated Wnt and Akt signaling pathways are strongly associated with patient's survival expectation, which was validated by five external datasets containing various cohorts and omics types. They had expected their workflow to be practical for HCC prognosis prediction, as it is the first study employing deep learning to identify multi-omics features in order to predict differential survival of HCC patients. Deep learning methods can be employed to pursue precision and personalized medical care

Table 1.1: Insights of DL in healthcare

Architecture	Applications	Scope	Comments
DNN	EHR,EEG,Community,Drug,Genomics	General	Conventional, various tasks
AE	EHR,ECG,Community,Wearable,Drug,Genomics	General	Feature extraction, usually combined with other models
DBN&DBM	EHR,ECG,EEG,Wearable	General	Various tasks, sometimes as feature extractor
CNN	EHR,ECG,EEG,Wearable,Drug	Specific	Uniformly sampled data, such as image & time series
RNN	EHR,ECG,Wearable	Specific	Sequences
Attention	EHR,ECG,EEG,Community,Drug	–	Performance improvement, usually combined with other models
GAN	EHR,ECG,Drug	General	Data augmentation & anonymization
GNN	EHR,Community,Drug	Specific	Graph structure data

by finding comprehensive and reasonable biomarkers of certain diseases. Deep learning architectures have great potential to integrate and analyze various data from different sources, for example, DNA sequence data, gene expression data, protein structure data and so on.

After reviewing all the works, we give some insights of DL models in healthcare in Table 1.1. The DL models are divided into general models and specific models. General models refer to those models that are capable to be used in various tasks, and compatible with input data in various formats. Specific models refer to those models that are designed to accomplish certain task, and the input data is strictly restricted.

DNN is conventionally a general model, which has been used in nearly all the tasks. As mentioned in Section 1.2, AE is generally used for feature extraction, which is an unsupervised learning model. However, it is usually combined with other supervised learning models to accomplish certain tasks. AE is the most general deep learning model, it could be used in almost all kinds of tasks. DBN\DBM is general model, too. GAN is general, which is used for data augmentation and anonymization for all kinds of data.

Attention could formulate transformer, can combine with other model like DNN, CNN, RNN and GNN, its scope depends on its partner.

CNN, RNN and GNN are specific models, which are designed for specific tasks. CNN is good at handling uniformly sampled data. RNN is designed to handle sequences, such as speech, language and text, which is easy to be extended to any event sequences like EHR. GNN is good at modeling data with graph structure, which means there are physical or abstract relationships between data.

1.4 Liver disease screening based on densely connected deep neural networks

In this section, we introduce our work [Yao et al. 2020] for example, which apply DNN for liver disease screening. The liver disease screening is formulated as a classification problem, where subjects with liver disease are positive samples, and subjects without liver disease are negative samples. The task is to find out whether the subject is positive or negative sample

Table 1.2: Summary of the selected LFT indicators

Indicators	Healthy range	Unit
Alanine aminotransferase(ALT)	1.00–40.00	U/L
Aspartate aminotransferase(AST)	1.00–45.00	U/L
AST/ALT	0.00–1.15	NULL
Total bilirubin(TBIL)	3.42–20.50	μmol/L
Direct bilirubin(DBIL)	0.10–6.84	μmol/L
Indirect Bilirubin(IB)	2.00–12.00	μmol/L
Alkaline phosphatase(ALP)	15.00–142.00	U/L
Glutamyl transpeptidase(r-GT)	11.00–50.00	U/L
Total protein(TP)	60.00–80.00	g/L
Albumin(ALB)	35.00–55.00	g/L
Globulin(GLOB)	20.00–30.00	g/L
Albumin/Globulin(A/G)	1.50–2.50	NULL
Cholinesterase(CHE)	4500–13000	U/L

based on LFT results. The diagnosis given by doctors is used to check whether the subject is positive or not.

1.4.1 LFT

Because of the diverse functions of the liver, there are many methods for liver function tests, and many kinds of indicators. For the purpose of large-scale screening, we only adopt the most commonly used LFT indicators. The selected LFT indicators are listed in Table 1.2, which contains the name, healthy range and unit of the LFT indicators. There are 11 LFT indicators (including ALT, AST, TBIL, etc.), and 2 derived indicators (AST/ALT, A/G). These LFT indicators are important parts of routine medical examinations, which ensure that the model proposed in these paper can be extended to large-scale screening.

1.4.2 Dataset and Preprocessing

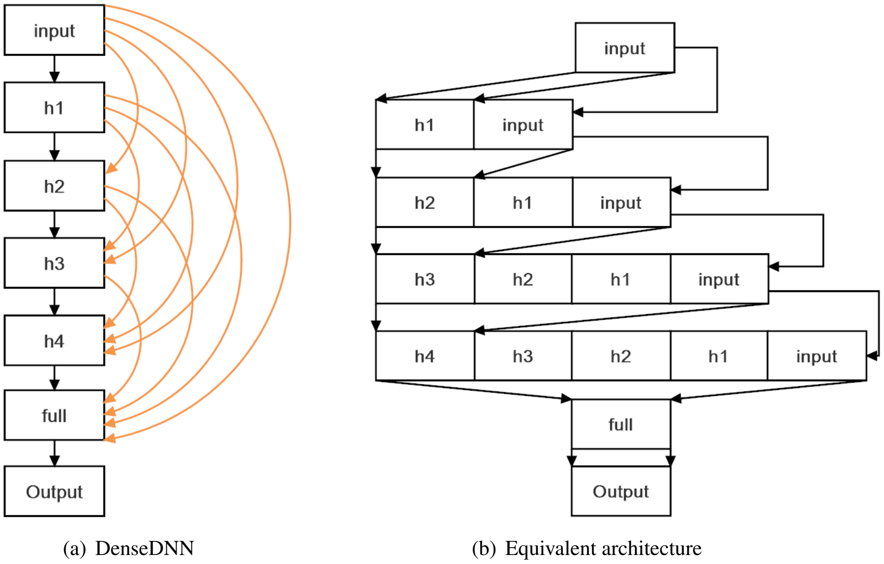
The dataset was collected from a hospital in northwestern China, which contains LFT data from 2015 to 2018. There are 84,685 samples in total. After removing samples with missing values, 76,914 LFT samples are retained and used in our study. Among these samples are 43,572 males and 33,329 females. 64,226 healthy samples and 12,688 samples was diagnosed as different degrees of liver disease. The average age of patients is 55.1(±14.2) years old, the youngest is 1 year old, and the oldest is 95 years old. It is worth noting that, to our knowledge, this is the largest LFT dataset ever studied, more than 100 times larger that the previous datasets. Therefore, this is a real large-scale liver disease screening application scenario. As a result, the experimental results should be considered more convincing.

We adopted 15 features for liver disease screening, include gender, age and 13 LFT indicators as shown in Table 1.2. All the features are numerical, except for the gender, which is transformed to numerical value 1 for male and 0 for female. All the features are normalized by Equation 1.1.

$$x_{ij} = \frac{x_{ij} - \bar{x}_i}{std(x_i)} \tag{1.1}$$

where x_{ij} is the value of the ith feature of the jth sample, x_i is the vector composed by ith feature of all the samples, $\bar{x}_i$ is the mean value of x_i, $std(x_i)$ is the standard deviation of x_i.

1.4.3 DenseDNN



(a) DenseDNN (b) Equivalent architecture

Figure 1.2: Overall architecture of densely connected DNN.

We present a novel architecture for deep neural networks, named densely connected deep neural networks (DenseDNN). As other deep neural networks, DenseDNN is an end-to-end deep neural network model.

Figure 1.2(a) is the architecture of DenseDNN. Since there are too many connections mixed together, Figure 1.2(b) illustrates an equivalent architecture. The features after preprocessing are fed into the DenseDNN as the input layer. As shown in the figure, the architecture is different from conventional DNN. There are 4 hidden layers (labeled as h1,h2,h3,h4) in the DenseDNN. All the neurons in the previous layers are concatenated to form a wider layer, which is the input of the subsequent layer. For example, the input of layer h3 is the input layer, h1 and h2, which establish direct connections between h3 and all its previous layers. Then the input layer, h1, h2, h3 are concatenated to be the input of h4. Similarly, the direct connections of h4 to all previous layers are established. The fully connected layer and output layer is similar as conventional DNN.

As shown in Figure 1.2(a), compared to the layer-to-layer connection in conventional DNNs, we densely add direct connections (shortcuts, marked in orange) across all the layers, including the input layer, the hidden layers, and the fully connected layer. These shortcuts ensure that all the layers are direct input of their subsequent layers, and all the subsequent layers get input from all their previous layers. Such direct connections shortened the distance between the output layer and all intermediate layers, which tends to alleviate the vanishing-gradient problem, enhancing information exchange in both feed forward and back propagation of the entire neural networks. Take h2 for example, if no shortcut, the only pathway from h2 to output is h2-h3-h4-full-output, which takes 4 steps. Shortcuts lead to 3 additional shorten pathways from h2 to output: h2-full-output; h2-h3-full-output, h2-h4-full-output, which takes 2, 3 and 3 steps, respectively.

In the training phase, the shortcuts could enhance the gradient back propagation. In the test phase, the hidden layers contains information of all the forward layers with different depths (input, h1, h2, h3, h4). For example, h4 received direct information from input, h1, h2, h3, while h4 in conventional DNN received direct information from h3 only. That is why the architecture adjustment is capable of improving the performance of conventional DNN.

We introduce the dense connections to the network architecture in order to improve the performance of conventional DNN. The subsequent experimental results will validate the expected performance improvements.

1.4.4 Implementation Details of DenseDNN

The input layer includes 15 neurons, each corresponding to one feature. The 4 hidden layers are with dense connections, where each layer includes 512 neurons. Each hidden layer is composed of dense connections, 'tanh' activation, and dropouts. The dropout rate was set to 0.5. The fully connected layers includes 1024 neurons with a drop rate of 0.5, but the activation is 'sigmoid'. The output layer of DenseDNN is a softmax layer with two neurons, corresponding to liver disease or not, respectively.

There are 2 important parameters in DenseDNN, the first one is the width of the hidden layers (number of neurons per hidden layer), the second one is dropout rate. We have tried different values for the 2 parameters, which are listed in the Section 1.5.3. The values mentioned above give the best result.

1.5 Results and Discussions

1.5.1 Measurements

To measure the performance of the proposed method for liver disease screening, the sensitivity (Sen), specificity (Spec), Detection Rate (DR), False Alarm Rate (FAR) and accuracy (Acc) are calculated, which are defined as:

$$Sen = TP/(TP+FN), \tag{1.2}$$

$$Spec = TN/(TN+FP), \tag{1.3}$$

$$DR = Sen, \tag{1.4}$$

$$FAR = 1 - Spec, \tag{1.5}$$

$$Acc = (TP+TN)/(TP+FN+TN+FP), \tag{1.6}$$

in which TP (True Positive) is the number of positive samples that are recognized as positive; FN (False Negative) is the number of positive samples that are recognized as negative; TN (True Negative) is number of negative samples that are recognized as negative; and FP (False Positive) is number of negative samples that are recognized as positive. In general, the goal of a good detection method is to minimize the FAR while maximizing all the other 4 performance metrics.

The test results are achieved by a 5-fold cross-validation. We divide the whole dataset of 76,914 samples randomly into 5 subsets. Each subset takes its turn as the test set while the remaining four subsets are combined to be used for training and validation. The cross validation strategy ensures that there are no overlap between training and testing datasets. During the training procedure, for each round, the dataset is randomly divided into training and validation sets with a ratio of 9:1. As a result, in one round of the cross validation, $1/5$ of the whole dataset (15,383 samples) is used for testing; the rest $4/5$ of the whole dataset (61,531 samples) is used for training (55,377 samples) and validation (6,154 samples).

The network is trained using Adam, an efficient stochastic gradient-based optimizer [**?**]. The batch size is set to 1024.

1.5.2 Reference models

For comparison, we also test Logistic Regression (LR), Random Forests (RF), conventional Deep Neural Networks (DNN) on the same dataset. LR is the most commonly used statistical analysis tools in clinical data modeling. RF is one of the most commonly used machine learning models. DNN is adopted as a reference model to show the effectiveness of the proposed DenseDNN architecture.

Table 1.3: Comparison of different widths.

width	64	128	256	512	1024	2048
AUCs	0.8851	0.8881	0.8904	0.8919	0.8922	0.8911

Table 1.4: Comparison of different dropout rates.

Dropout rate	0.3	0.4	0.5	0.6	0.7
AUCs	0.8812	0.8891	0.8919	0.8904	0.8856

In the DNN, the connections across layers are removed. Its architecture is composed by input layer, hidden layers (h1 to h4), fully connected layers, and output layer. The input of each layer includes only one adjacent previous layer.

1.5.3 Parameter setting

In this paragraph, we test two important parameters in DenseDNN, the first one is width (number of neurons per hidden layer) of the hidden layers, the second one is dropout rate. We have tried different values for the two parameters.

A quantitative measurement of detection model is Area Under Curve (AUC), which is defined as ratio of area under the Receiver Operating Characteristic (ROC) curve. Generally, the larger the AUC is, the better performance the detector gets.

Table 1.3 lists the AUCs of DenseDNN with different width, where dropout rate is fixed to 0.5. As shown in the table, DenseDNN with 512 neurons and 1024 neurons per hidden layer perform the best. The AUC is 0.8919 when width is 512, and 0.8922 when width is 1024. Since the performance is similar, we select 512 as the final width to reduce the complexity of DenseDNN.

Table 1.4 lists AUCs of DenseDNN with different dropout rates, with width fixed to 512. We can find that DenseDNN with a dropout rate of 0.5 gives the highest AUC. In the following tests, 0.5 is selected as the dropout rate.

1.5.4 Learning curves

Figure 1.3 shows the learning curves of the two kinds of DNNs, for both training and validation sets. Horizontal and vertical axes are, respectively, the number of epochs and the loss (cross entropy), which is defined as

$$C = \frac{1}{n}\sum_{i=1}^{n}[y_t \ln y_p + (1-y_t)\ln(1-y_p)], \tag{1.7}$$

where y_t is the true label, and y_p is the predicted label.

Table 1.5: Comparison of AUCs.

Model \ Epochs	100	200	300	400	500	600
DNN	0.8831	0.8867	0.8884	0.8896	0.8897	0.8897
DenseDNN	0.8881	0.8919	0.8929	0.8930	0.8935	0.8933

For the purpose of comprehensive evaluation, we trained the model for more epochs to get a complete learning curve.

As shown in the figure, on the training set, the loss drops consistently as the number of epochs increases. The loss of DenseDNN drops more sharply than that of DNN. There is a large gap between the 2 curves. The gap widens with the progression of epochs. The loss of DenseDNN drops to about 0.24 in 600 epochs, while that of DNN drops to about 0.265.

On the validation set, the losses of both models drop to about 0.28, and no longer decline significantly. However, the loss of DenseDNN drops more sharply than that of DNN. The DenseDNN takes about 90 epochs to reach the loss of 0.28, while the DNN takes about 160 epochs. After about 350 epochs, the loss of DenseDNN increases slightly, which is a sign of overfitting. Despite the overfitting phenomenon on the training set, we can find that the loss on the validation set remains stable. Such overfitting is moderate, may be not harmful, and may even improve performance.

Generally, we select an appropriate fitting that minimizes the loss on the validation set. As shown in Figure 1.3, for DenseDNN, the training loss drops to 0.28 after about 90 epochs and keep steady, and raises from 350 epochs. The loss of DNN drops to 0.28 after about 160 epochs and keep steady. Form the learning curves, we infer that for DenseDNN, 90 to 350 epochs are appropriate, and that for DNN is 160 to 600.

Since its loss drops more quickly than DNN, we infer that DenseDNN is easier to train than conventional DNN. This is because the shortcut connections in DenseDNN shortens the distance between all layers, making the errors propagation faster and speeding up the training process.

To find the best fit, we trained the two models for 600 epochs and checked the AUC of models with different training rounds on the test sets. As shown in Table 1.5, the AUC increases as the numbers of training epochs increase for both models. For example, the AUC of DNN increases from 0.8831 at 100 epochs to 0.8897 at 500 epochs, does not improve as the number of training epochs increases to 600. As for DenseDNN, the AUC increases from 0.8881 at 100 epochs to 0.8935 at 500 epochs, and drops slightly when the number of training epochs increases to 600. From these data, we can infer that 500 epochs is the best fit for both models. However, from the learning curves in Figure 1.3, we have found that 160–350 epochs is the best fitting for validation set. Such difference is due to the difference between the validation set and the testing set. According to the results on the test sets, we have found that 500 epoches is best fitted model. However, in practical situation, we can not

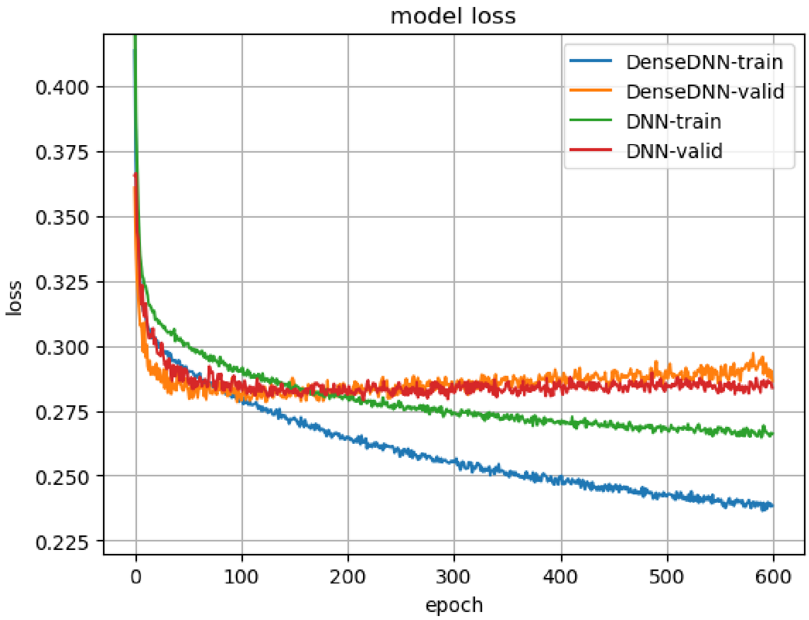


Figure 1.3: Learning curves of the neural networks

get the test sets at the stage of training models. The model should be determined according to the performance on validation set, which suggests 160–350 epochs is the best fitting. In the following experimental setting, both models are trained for 200 epochs.

1.5.5 Comparison with conventional methods

The ROC curves of DenseDNN, DNN, random forests, and logistic regression for liver disease detection are given in Figure 1.4. As shown in Figure 1.4, for each curve, with the increasing false alarm rate, the detection rate also rises up. Generally, The higher the curve is, the better performance the detector gets. The AUCs are presented in Table 1.6.

We find that LR is the worst detector, there is a very large margin between the ROC curve of LR and the other ROC curves. All the other ROCs are consistently much higher than that of LR. The AUC of LR is 0.7977, while that of RF, DNN, and DenseDNN are 0.8790, 0.8867, and 0.8919, respectively. The machine learning models, including RF, DNN and DenseDNN, get much better detection performance than LR.

Furthermore, we compare the detection performance of all the three machine learning models. The ROC curves of deep learning (including DenseDNN and DNN) are always higher than that of the RF with a significant margin. The AUC of the 2 deep learning model are 0.8867 and 0.8919, respectively, and that of RF is 0.8790.

Table 1.6: Prediction performance of different models for liver disease screening

Model	Sen with different Specs					AUC
	0.95	0.90	0.85	0.80	0.75	
LR	0.3412	0.4794	0.5690	0.6416	0.7003	0.7977
RF	0.5724	0.6857	0.7517	0.7986	0.8353	0.8790
DNN	0.5643	0.6884	0.7604	0.8116	0.8517	0.8867
DenseDNN	**0.5840**	**0.7068**	**0.7737**	**0.8204**	**0.8565**	**0.8919**

As for the two deep learning models, The ROC curve of DenseDNN is always higher than that of DNN with a visible margin. The AUC are 0.8919 vs. 0.8867.

According to the ROC curves and their AUCs, we conclude that, for the liver disease screening task, the DenseDNN architecture improves the performance significantly than conventional machine learning methods, and conventional deep neural networks. The proposed model improves the screening performance.

The Spec, Sen, and AUCs of all the models for liver disease screening are listed in Table 1.6. Five specificity values are selected as reference, which are 0.95, 0.90,0.85, 0.80, and 0.75. We can figure out that DenseDNN gets largest sensitivity on all the different values of specificity, being low or high. The AUC of DenseDNN is 0.8919, which is greater than that of all the other models. The comparison shows that the proposed DenseDNN is capable of detecting liver diseases more effectively.

From Figure 1.4 and Table 1.6, we can figure out that, not only the proposed model, but also all the DNN models outperform conventional machine learning models. Deep learning is more effective for nonlinear modeling than other conventional models.

As demonstrated in Section 1.4.3, the shortcuts across the hidden layers strengthen information exchange in both feed forward and back propagation procedures of the entire neural network, which make the proposed DensDNN learn from information across all the layers. The shortcuts make the training and predicting procedure more effective.

From another point of view, deeper neural network usually means more complex nonlinear modeling. Compared with traditional DNN, the shortcuts added by DenseDNN connect hidden layers with different complexity, which enhance the modeling ability of each layer, and lead to better performance.

1.6 Challenges and Limitations

Although deep learning models have shown their power in so many healthcare applications, there are still a few major challenges. We summarize them as follows:

Data. The deep NN models are data driven. The number of model parameters is much higher than that of conventional models. Huge volume of data is needed to train the models. However, in healthcare applications, data collection is not easy, a dataset with 10,000 samples

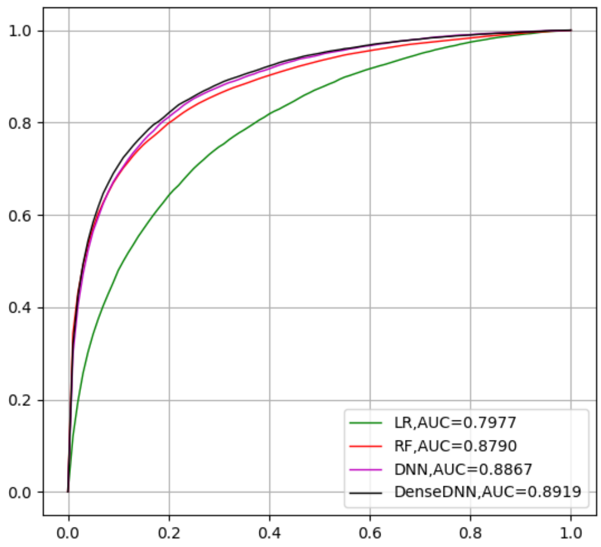


Figure 1.4: ROC curves of different models for liver disease screening

is often considered large and is hard to get large. This scale is small if compared to Imagenet dataset, which has 14,197,122 images. Furthermore, data in healthcare is often unbalanced. For disease screening tasks, patients with target outcome are typically scarce compared to healthy cohorts. Building big, representative dataset is an important and time-consuming task. GAN could alleviate this problem, but it can generate new data according to the distribution of existing data, and cannot increase the overall representation ability of the data set.

Interpretability. In the tasks of image recognition and speech recognition, we care more about whether the model works accurately, and care less about why it works. Although some visualization of the feature maps may help us understand the intermediate results, most of the deep neural networks are end-to-end black-boxes and not interpretable. In healthcare task, interpretability is far more important than other applications. We need to analyze risk factors, and find out what is the best treatment for certain disease. Che et al. [Che et al. 2015] attempted to distill knowledge from deep NN by Interpretable Mimic Learning. Lundberg and Lee interpreted model by Shapley value, which measures marginal contribution of each feature. These ideas are helpful for model interpretation. Furthermore, data visualization is still the most powerful tool for model interpretation.

Data representation. In many traditional learning tasks, such as image or speech, the data is homogeneous and neat. In healthcare, some data (such as EHR) are irregular and poorly structured. Moreover, many data often have missing values. For example, vital signs manually collected at hospitals often have missing fields. At present, EHR data is represented as a temporal-events matrix and missing values are often handled by simple interpolation [Mehrabi et al. 2015][Cheng et al. 2016]. A better representation may help to improve the performance.

Generalization ability. In other applications, the difference between training dataset and testing dataset is not significant, because the training and testing datasets are often generated by the same or similar underlying distribution. Therefore, the learned models generalize well. In healthcare, the difference between training and testing data is prone to be significant. For example, model training on dataset from Americans would not suit for Chinese [Yao et al. 2017]. Transfer learning is a powerful method for such settings.

Computational complexity. The deep NNs are one of the most complex machine learning models. They are time, space, and memory consuming. Huge number of operations takes time for model running. This problem is especially severe for wearable devices with limited memory, computing power and battery. Simplified model may be helpful, Chen et al. [Chen et al. 2015] have tried to reduce complexity in deep NN, which shrinks the storage requirements of neural networks substantially while mostly preserving generalization performance. Another solution is to design energy-efficient hardware accelerators for deep NNs. For example, on a number of representative neural network layers, it is possible to achieve a speedup of 450.65x over a Graphics Processing Unit (GPU), and reduce the energy by 150.31x on average for a 64-chip DaDianNao system [Chen et al. 2016].

1.7 Conclusion and Further works

In this chapter, we have surveyed applications of deep learning in healthcare areas, including EHR, ECG, EEG, community healthcare, data from wearable devices, drug analysis and genomics analysis. We have shown that deep learning has achieved remarkable results in these areas. We introduce DenseDNN, a neural network architecture for liver disease screening. The task is to detect the occurrence of liver diseases according to the LFT data. The proposed DenseDNN jointly learns from information across all the layers. The shortcuts across all the hidden layers strengthen feature extraction in both feed forward and back propagation during training, which makes the training and predicting procedure more effective. Test results on a dataset with 76,914 samples show that the proposed DenseDNN model achieves better performance than conventional machine learning models and conventional deep learning models.

Deep learning has shown great potential in promoting the revolution in the healthcare industry. The unmatched learning ability of deep learning has made it an attractive and indispensable technology for analyzing clinical and healthcare data. One interesting future direction is using deep learning to learn from multi-dimensional, heterogeous, and non-structural personal data, such as demographics, diets, habits, sleeping, mental health, medical imaging, vital signs, medication, lab tests, etc. Such fusion of information can lead to new breakthrough in data-driven healthcare decision making.

1.8 Exercises and Projects

1. How might CBIR be applied so teachers with a computer and connected camera can be reminded of the names of students in their class?
2. Consider the two colourful images (Image A and Image B) showed below, represented in the RGB color space. Suppose that the intensity values of each pixel in all bands (R, G, and B) are the same. Furthermore, each (R,G,B) triplet is represented by a single intensity value. For example the triplet $(R,G,B) = (2,2,2)$ is represented by the intensity value 2.

   Suppose also that the colour space was quantized in five colors with intensity values 0, 1, 2, 3, and 4.
3. Compute the L_1 between the *Color Histograms* (5 bins) of the two images. The L_1 distance between two color histograms H_A and H_B is computed as follows: $L_1(H_A, H_B) = \sum_{i=1}^{K} |H_A[i] - H_B[i]|$, where K is the size of both histograms (5, in the case).
4. By considering both the feature vector extraction function and the distance function defined of the descriptor *Color Coherence Vector – CCV* [Skeen and Stonebraker 1983], compute the distance $\delta_{CCV}(A,B)$ between the two images.
5. Consider the existence of two classes (*class 1* and *class 2*) composed of five images each. Consider the existence of three different descriptors (*descriptor 1*, *descriptor 2*, and *descriptor 3*), whose feature vector extraction functions extract vectors belonging to the R^2 space. Table **??** shows the coordinate of each image of each class, considering the three descriptors.

2 This Is Another Chapter

3 And Yet Another Chapter

A This Is an Appendix

Of course, in specifying the particular communication system under investigation, we must know the important physical parameters, such as transmitted power, bandwidth, type(s) of noise present, and so on, and information theory allows these constraints to be incorporated. However, information theory does not provide a way for communication system complexity to be explicitly included. Although, this is something of a drawback, information theory itself provides a way around this difficulty, since it is generally true that as we approach the fundamental limit on the performance of a communication system, the system complexity increases, sometimes quite drastically. Therefore, for a simple communication system operating far from its performance bound, we may be able to improve the performance with a relatively modest increase in complexity. On the other hand, if we have a rather complicated communication system operating near its fundamental limit, any performance improvement may be possible only with an extremely complicated system.

In this chapter we are concerned with the rather general block diagram shown in Figure **??**. Most of the early work by Shannon and others ignored the source encoder/decoder blocks and concentrated on bounding the performance of the channel encoder/decoder pair. Subsequently, the source encoder/decoder blocks have attracted much research attention. In this chapter we consider both topics and expose the reader to the nomenclature used in the information theory literature. Quantitative definitions of information are presented in Sec. **??** that lay the foundation for the remaining sections. In Secs. **??** and 1.1 we present the fundamental source and channel coding theorems, give some examples, and state the implications of these theorems. Section **??** contains a brief development of rate distortion theory, which is the mathematical basis for data compression. A few applications of the theory in this chapter are presented in Sec. **??**, and a technique for variable-length source coding is given in Sec. **??**.

In this chapter we are concerned with the rather general block diagram shown in Figure **??**. Most of the early work by Shannon and others ignored the source encoder/decoder blocks and concentrated on bounding the performance of the channel encoder/decoder pair. Subsequently, the source encoder/decoder blocks have attracted much research attention. In this chapter we consider both topics and expose the reader to the nomenclature used in the information theory literature. Quantitative definitions of information are presented in Sec. **??** that lay the foundation for the remaining sections. In Secs. **??** and 1.1 we present the fundamental source and channel coding theorems, give some examples, and state the implications of these theorems.

A.1 What is Classification? What is Prediction?

Of course, in specifying the particular communication system under investigation, we must know the important physical parameters, such as transmitted power, bandwidth, type(s) of noise present, and so on, and information theory allows these constraints to be incorporated. However, information theory does not provide a way for communication system complexity to be explicitly included. Although, this is something of a drawback, information theory itself provides a way around this difficulty, since it is generally true that as we approach the fundamental limit on the performance of a communication system, the system complexity increases, sometimes quite drastically. Therefore, for a simple communication system operating far from its performance bound, we may be able to improve the performance with a relatively modest increase in complexity. On the other hand, if we have a rather complicated communication system operating near its fundamental limit, any performance improvement may be possible only with an extremely complicated system.

1. This is a number list with a short item.
2. And another item that is much longer so that we can make sure it is formatted correctly and so forth and so on.
3. And a final short item.

Bibliography

M. M. Al Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, and R. R. Yager. 2016. Deep learning approach for active classification of electrocardiogram signals. *Information Sciences*, 345: 340–354.

A. Aliamiri and Y. Shen. 2018. Deep learning based atrial fibrillation detection using wearable photoplethysmography sensor. In *Biomedical & Health Informatics (BHI), 2018 IEEE EMBS International Conference on*, pp. 442–445. IEEE.

A. Benton, M. Mitchell, and D. Hovy. 2017. Multi-task learning for mental health using social media text. In *Proceedings of EACL*.

J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. 2013. Spectral networks and locally connected networks on graphs. *arXiv: Learning*.

K. Chaudhary, O. B. Poirion, L. Lu, and L. Garmire. 2017. Deep learning based multi-omics integration robustly predicts survival in liver cancer. *bioRxiv*, p. 114892.

Z. Che, S. Purushotham, R. Khemani, and Y. Liu. 2015. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*.

W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen. 2015. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294.

Y. Chen, T. Chen, Z. Xu, N. Sun, and O. Temam. 2016. Diannao family: energy-efficient hardware accelerators for machine learning. *Communications of the ACM*, 59(11): 105–112.

Y. Cheng, F. Wang, P. Zhang, and J. Hu. 2016. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 432–440. SIAM.

M. Cheung and J. M. Moura. 2020. Graph neural networks for covid-19 drug discovery. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5646–5648. IEEE.

E. Choi, Z. Xu, Y. Li, M. Dusenberry, G. Flores, E. Xue, and A. Dai. 2020. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 606–613.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

P. Danaee, R. Ghaeini, and D. A. Hendrix. 2016. A deep learning approach for cancer detection and relevant gene identification. In *Pacific Symposium on Biocomputing*, volume 22, p. 219. NIH Public Access.

F. Dernoncourt, J. Y. Lee, O. Uzuner, and P. Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3): 596–606.

G. Dong, M. Tang, L. Cai, L. E. Barnes, and M. Boukhechba. 2021. Semi-supervised graph instance transformer for mental health inference. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1221–1228. IEEE.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

H. Greenspan, B. van Ginneken, and R. M. Summers. 2016. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5): 1153–1159.

W. L. Hamilton, Z. Ying, and J. Leskovec. 2017. Inductive representation learning on large graphs. pp. 1024–1034.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6): 82–97.

G. E. Hinton and T. J. Sejnowski. 1986. Learning and relearning in boltzmann machines. *Parallel Distributed Processing*, 1.

G. E. Hinton, S. Osindero, and Y.-W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7): 1527–1554.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature*, 521(7553): 436–444.

C. Li, J. Wang, Z. Niu, J. Yao, and X. Zeng. 2021. A spatial-temporal gated attention module for molecular property prediction based on molecular geometry. *Briefings in Bioinformatics*, 22(5): bbab078.

G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. 2017. A survey on deep learning in medical image analysis. *arXiv preprint arXiv:1702.05747*.

S. Liu, X. Wang, L. Zhao, B. Li, W. Hu, J. Yu, and Y. Zhang. 2021. 3dcann: a spatio-temporal convolution attention neural network for eeg emotion recognition. *IEEE Journal of Biomedical and Health Informatics*.

S. Mehrabi, S. Sohn, D. Li, J. J. Pankratz, T. Therneau, J. L. S. Sauver, H. Liu, and M. Palakal. 2015. Temporal pattern and association discovery of diagnosis codes using deep learning. In *Healthcare Informatics (ICHI), 2015 International Conference on*, pp. 408–416. IEEE.

R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley. 2016. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6: 26094.

A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al. 2018. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1): 18.

P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng. 2017. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*.

D. Ravi, C. Wong, B. Lo, and G. Z. Yang. 2016. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *IEEE International Conference on Wearable and Implantable Body Sensor Networks*, pp. 71–76.

D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang. 2017. Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1): 4–21.

F. Rosenblatt. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3): 1.

R. Salakhutdinov and G. Hinton. 2009. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pp. 448–455.

R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. 2017. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11): 5391–5420.

D. Skeen and M. Stonebraker. 1983. A formal model of crash recovery in a distributed system. *IEEE Trans. Softw. Eng.*, SE-9(3): 219–228.

A. Torfi and E. A. Fox. 2020. Corgan: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. In *The Thirty-Third International Flairs Conference*.

T. Unterthiner, A. Mayr, G. Klambauer, and S. Hochreiter. 2015. Toxicity prediction using deep learning. *Computer Science*, 3(8).

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

P. J. Werbos. 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*.

R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2): 270–280.

N. Wulan, W. Wang, P. Sun, K. Wang, Y. Xia, and H. Zhang. 2020. Generating electrocardiogram signals by deep learning. *Neurocomputing*, 404: 122–136.

Z. Xu, S. Wang, F. Zhu, and J. Huang. 2017. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In *ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, pp. 285–294.

S. Yan, Y. Xiong, D. Lin, and X. Tang. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. pp. 7444–7452.

Z. Yao, Z. Zhu, and Y. Chen. 2017. Atrial fibrillation detection by multi-scale convolutional neural networks. In *Information Fusion (Fusion), 2017 20th International Conference on*, pp. 1–6. IEEE.

Z. Yao, J. Li, Z. Guan, Y. Ye, and Y. Chen. 2020. Liver disease screening based on densely connected deep neural networks. *Neural Networks*, 123: 299–304.

Z. Yao, Q. Xu, Y. Chen, Y. Tu, H. Zhang, and Y. Chen. 2021. Internet traffic forecasting using temporal-topological graph convolutional networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE.

Z.-J. Yao, J. Bi, and Y.-X. Chen. 2018. Applying deep learning to individual and community health monitoring data: A survey. *International Journal of Automation and Computing*, 15(6): 643–655.

R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. pp. 974–983.

Q. Zhang, X. Chen, Q. Zhan, T. Yang, and S. Xia. 2017. Respiration-based emotion recognition with deep learning. *Computers in Industry*, 92: 84–90.

T. Zhang, H. Lin, B. Xu, L. Yang, J. Wang, and X. Duan. 2021. Adversarial neural network with sentiment-aware attention for detecting adverse drug reactions. *Journal of Biomedical Informatics*, 123: 103896.

Author's Biography

Your Name

Your Name began life as a small child ...