The Effects of Deep Network Topology on Mortality Prediction

Hao Du*, Mohammad M. Ghassemi*, Mengling Feng

Abstract— **Deep learning has achieved remarkable results in the areas of computer vision, speech recognition, natural language processing and most recently, even playing Go. The application of deep-learning to problems in healthcare, however, has gained attention only in recent years, and it's ultimate place at the bedside remains a topic of skeptical discussion. While there is a growing academic interest in the application of Machine Learning (ML) techniques to clinical problems, many in the clinical community see little incentive to upgrade from simpler methods, such as logistic regression, to deep learning. Logistic regression, after all, provides odds ratios, p-values and confidence intervals that allow for ease of interpretation, while deep nets are often seen as 'black-boxes' which are difficult to understand and, as of yet, have not demonstrated performance levels far exceeding their simpler counterparts. If deep learning is to ever take a place at the bedside, it will require studies which (1) showcase the performance of deep-learning methods relative to other approaches and (2) interpret the relationships between network structure, model performance, features and outcomes. We have chosen these two requirements as the goal of this study. In our investigation, we utilized a publicly available EMR dataset of over 32,000 intensive care unit patients and trained a Deep Belief Network (DBN) to predict patient mortality at discharge. Utilizing an evolutionary algorithm, we demonstrate automated topology selection for DBNs. We demonstrate that with the correct topology selection, DBNs can achieve better prediction performance compared to several bench-marking methods.**

I. INTRODUCTION

While the term 'Big Data' became popular only a couple years ago, the use of big data in many industries has been ongoing for decades. Over the course of the last several years industries including insurance, banking and telecommunications have grown steadily reliant on data for decision making and process control. Unlike the steady growth in reliance seen in other industries, the medical community's relationship with data, and the retrospective approach to quality improvement, has been more of a sudden onset than a gradual process. This reality comes as no surprise when one considers that the electronic collection, storage and analysis of health-care data is still a relatively new phenomenon.

Low data volumes are not the only issue that make Health-care data challenging. Many publicly available datasets are artifact ridden, heterogeneous, high dimensional, and temporally dependent. This reality poses challenges for machine learning practitioners that are interested in utilizing the data to provide decision support.

Most practitioners of applied machine learning acknowledge that feature extraction and data pre-processing account for a substantial component of the time and effort of any standard research project, and the health-care domain is no exception. As volumes of data grew in non-healthcare domains, we have observed the rise of a field which attempts to automate the feature extraction process itself - representation learning. 'Deep learning' describes a class of representation learning methods inspired by the muti-layered neural networks of the human brain. In recent years, deep learning methods have been applied to various AI problems in the areas of computer vision, speech recognition and natural language processing where they achieved impressive results [1].

The speech community was the first to leverage the power of representation learning for practical purposes, and demonstrated tremendous increases in the efficacy of speech recognition when using 'deep' techniques compared to the then state-of-the-art (Gaussian Mixture Models). In some cases, the community saw as much as a 50% reduction in error on essential benchmark data sets. These advances made end-user technologies, such as Apples Siri, a reality. Since then, 'deep' techniques have infiltrated practically every application area imaginable, with clinical decision support being a notable exception.

Deep learning has also been applied to several health-related problems, including the diagnosis of liver cancer, diabetes, heart-failure [2], tumor segmentation [3] and transplant acceptance [4]. Despite this work, there is clinical resistance to adopting deep learning for decision support due to a perceived lack of evidence that the 'deep' techniques actually outperform their simpler counterparts, such as logistic regression.

If it is not utilized with care, this critique of deep learning is valid. Indeed, many out-of-the-box 'deep' networks require (1) datasets with large degrees of freedom, which is almost never the case in clinical data (the human genome alone has more features than all the financial indicators of the world combined) and (2) an optimal network topology to ensure optimal network performance [5].

While the first issue may be reasonably addressed via manual feature selection, the second issues is all too often left undressed. Important features of deep networks, such as the number of layers, and the number of nodes within each layer and the connectivity structure are selected using investigator intuition as opposed to principled optimization.

*MMG and HD contributed equally to this work and are to be considered shared first authors.

M.M. Ghassemi is the corresponding author for this work. He is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology,Boston, MA 02139 USA. Phone: 617-599-6010. Email:ghassemi@mit.edu

H. Du is with the School of Electrical and Electronic Engineering, Nanyang Technology University,Singapore. Email:duha0001@e.ntu.edu.sg

M. Feng is with (1) the Institute for Infocomm Research, Singapore and (2) The Institute for Medical Engineering and Science, Massachusetts Institute of Technology,Boston, MA 02139 USA. Email:mfeng@mit.edu

978-1-4577-0220-4/16/$31.00 ©2016 IEEE

In this study, we will (1) explore the effects of network topology on the performance of a Deep Learning algorithm to predict critical care patients' 28-day mortality and (2) demonstrate how techniques in global optimization may be utilized to automatically determine network topology.

II. METHODS

A. Areas of Exploration

Data As deep learning methods represent data with multiple layers of abstractions, they require large amount of training data to obtain stable and reliable features. We will explore whether the Electronic Medical Record (EMR) data routinely collected for care purpose are sufficient for deep learning methods to achieve reasonable performance. A real-world EMR dataset on critical care patients will be used.

Topology Neural networks are the most common deep learning architecture. It has been reported that the topology, or *structure*, of the neural networks greatly affects their functionality [5]. We aim to discover the optimum topology through an evolution algorithm.

Computational Complexity Learning the connection weights of the deep neural networks are known to be computationally expensive. Experiments will be conducted on computational clusters to measure the computational time and memory needed for deep neural network training.

Learning Task Deep learning methods can be applied to many learning tasks, such as detection, forecasting, prediction, etc. We will focus on the application of deep learning for a predictive model in this study. To be specific, we apply deep learning to predict critical care patients' 28-day mortality based on their continuously collected EMR data.

B. Data & Problem Statement

Data for this study was extracted from the publicly available Multiparamter Intelligent Monitoring in Intensive Care database (MIMIX-II) [6]. MIMIC-II is a publicly available clinical database developed by the Massachusetts Institute of Technology (MIT), Phillips Healthcare, and Beth Israel Deaconess Medical Center (BIDMC). The database contains de-identified EMR data from over 32,000 critically ill patients treated in the ICUs at the BIDMC from 2001 to 2008.

To explore the feasibility and effectiveness of deep learning methods on health analytics problems, we applied a deep learning model to predict the 28-day mortality of critical care patients in the MIMIC-II database. Mortality prediction [7] is a very common problem in healthcare analytics, and was chosen for it's simplicity. although this study focuses on mortality predictions, most of our findings can be easily generalized to other predictive problems in health-care analtyics.

We extracted a total of 73 clinical variables from the MIMIC-II dataset:

- **Demographic and Severity:** age, gender, weight, critical service type (medical/surgical/cardiac/recovery), Simplified Acute Physiology Score (SAPS) [8], Sequential Organ Failure Assessment (SOFA) Score [9] and Glasgow Coma Score (GCS)

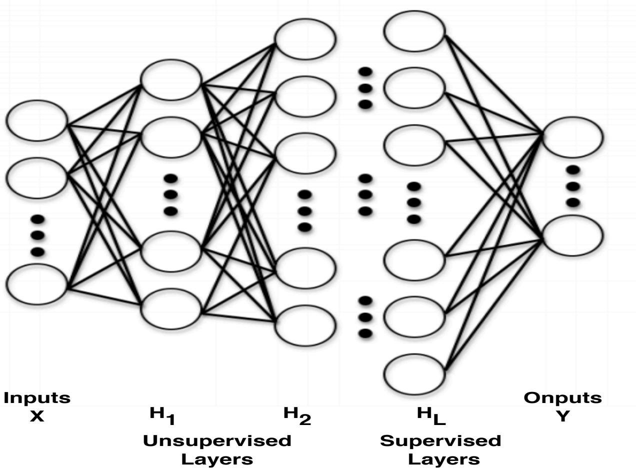

Fig. 1. Deep Belief Network with both unsupervised and supervised layers.

- **Co-morbidity:** all 23 co-morbidities used to calculate the Elixhauser score [10]. All these chronic conditions are judged based on the patients' assigned International Classification of Diseases (ICD-9) codes.
- **Lab results at Admission:** Albumin, ALP (Alkaline phosphatase), ALT (Alanine transaminase, AST (Aspartate transaminase), Bilirubin, BUN (Blood urea nitrogen), Cholesterol, Creatinine, Glucose, HCO3 (Serum bicarbonate, HCT (Hematocrit), K (Serum potassium, Lactate, Mg (Serum magnesium), Na (Serum sodium), PaCO2 (partial pressure of arterial CO2, PaO2 (Partial pressure of arterial O2, pH, Platelets, Troponin, WBC (White Blood Cell count),
- **Vital Signs (Median Values of first 24 hours):** Heart Rate, Temperature, Mean Arterial Blood Pressure, Oxygen Saturation and Spontaneous Respiration Rate
- **Interventions:** Usage of mechanical Ventilation, Vasopressor and sedative medications

We only utilized data from the first ICU admissions of every patient in the MIMIC data. We also removed patients missing any of our selected clinical variables. Following this exclusion criteria, our final data-set included 15,647 unique patients. All data was partitioned into training (60%), validation (20%) and testing sets (20%). All experiments were conducted on a computational cluster with 16 nodes, each node has 2*8 cores of 2.0GHz CPUs and 64 GB of memory. All BDN neural networks were implemented in Matlab using the toolbox described in [11].

C. Deep Belief Network

Deep learning methods aim to automatically learn feature relationships at multiple levels of abstraction, where features at higher levels in network are composed of features from lower levels [12]. Deep Belief Networks (DBN) [13] are a variety of unsupervised probabilistic generative models. As shown in Figure 1, DBNs are composed of multiple layers of hidden units/nodes that form a hierarchical abstraction of the input data. The output of DBNs are often connected to

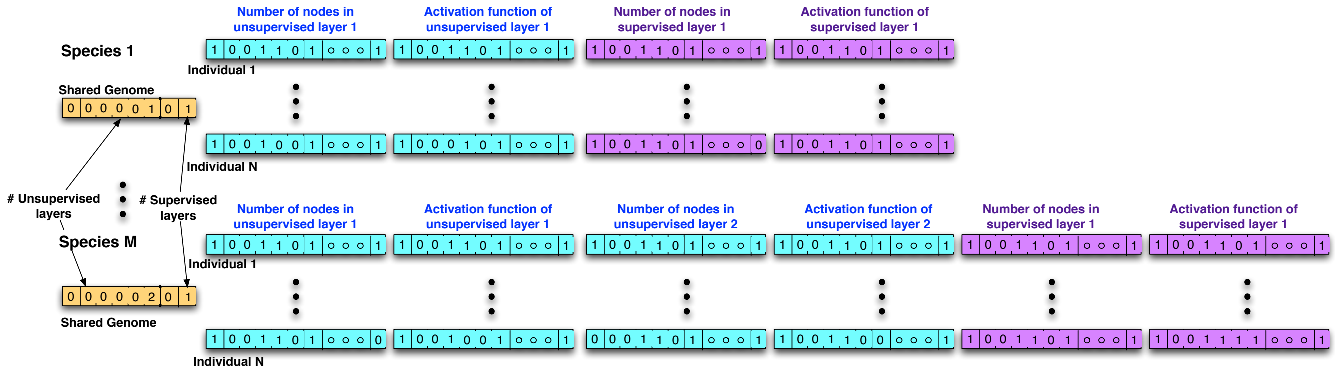

Fig. 2. Encoding various Deep Belief Network topologies with species shared genomes and individual genomes.

one or more supervised layers for classification.

Training DBNs can be computationally expensive as the number of nodes and hidden layers grow. Hinton et al [13] discovered that DBNs can be viewed as a compositions of Restricted Boltzmann Machines (RBMs), where each unsupervised hidden layer can be trained independently using the previous layer as observed input data. The process of the greedy layer-wise unsupervised training is performed as follows:

I. Train the first layer of DBN as an RBM that models the raw input x.

II. Use that first layer to obtain a representation of the input that will be used as data for the second layer.

III. Train the second layer as an RBM, taking the transformed data (samples or mean activation) as training examples (for the visible layer of that RBM).

IV. Iterate (II and III) for the desired number of layers, each time propagating upward either samples or mean values.

V. Append additional layers to the trained DBN for supervised predictions and fine-tune all the parameters of this deep architecture with respect to the supervised training criterion performance.

D. Evolutionary Algorithm for Network Topology Optimization

An optimal network topology is crucial to ensure optimal performance of DBNs[5]. The search space for DBN's network topology is, in theory, infinite. In this study, we employed an evolutionary algorithm (Genetic Algorithm) to discover an optimal network topology. In [14], researchers successfully demonstrated how an evolutionary algorithm could be applied to identify the topology of a neural network capable of completing a level of Super Mario on its own. We apply a similar approach in determining our network topology in this study.

An evolutionary algorithm is a type of optimization method that emulates nature's evolution process, i.e. generating and selecting the best solution based on the 'survival of the fittest' principle. As illustrated in Figure 2, we use binary codes to encode the parameters of DBN topoloy into a genome. Candidates in the populations are grouped into species, where individuals from the same species share a portion of genome that encodes the number of hidden unsupervised and supervised layers. An individual genome then contain codes for the number of nodes in the corresponding hidden layers and their activation functions. As a result, one can observe from Figure 2 that different species will have different length of genomes. The evolutionary process for the optimum topology is as follows:

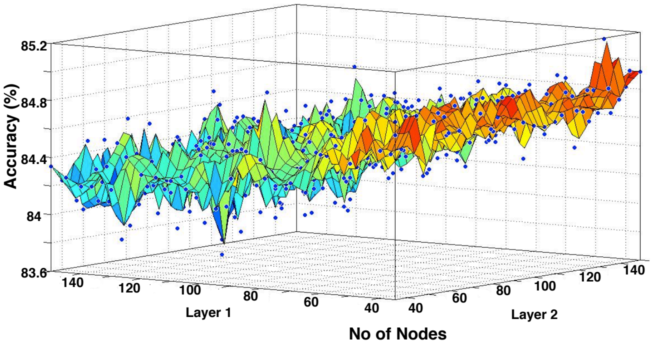

Fig. 3. Exhaustive search for the optimum network topology of neural network with two hidden layers.

I. Initialize the population by randomly generating M species and N individual candidates among each species.

II. Evaluate the prediction performance of all DBN topologies coded with candidate genomes.

III. Select the top $p\%$ candidates based on their performance as the survivors and discard all the rest. Species with small population become extinct.

IV. Evolve the surviving candidates through: (1) *mutations* on both the species shared and individual genomes and (2) *breeding* (exchange, merge or replace genomes) within and across species.

V. Iterate steps II to IV until the performance of candidates start to converge and report the best performance genome.

III. EXPERIMENTS AND RESULTS

A grid search that exhausts every possible topology of a two layer fully connected network was conducted to illustrate how variations in network topology can impact the performance of neural networks. Figure 3 summarizes the

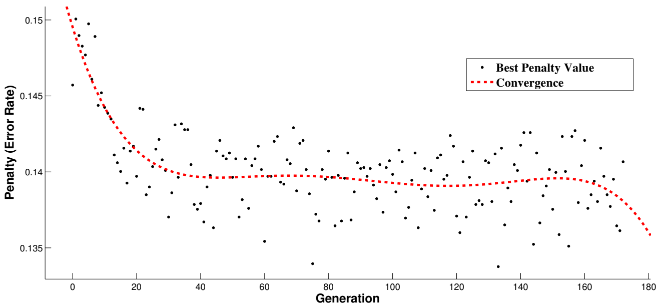

Fig. 4. The Evolutionary algorithm's iterative performance for optimum network topology identification. The red dotted line in the figure indicates convergence of the best performance over the generations.

results, where the maximum number of nodes was set to 150 for each layer. We observe that although the nature of the surface is rough, there are some general trends. The accuracy of the network is more strongly related to the number of hidden nodes in the first layer, compared to the second.

As the search space grows quadraticly with the number of nodes in each layer, it is not practical to apply an exhaustive search for topology optimization in large topological domains. The grid search for just the two layer network took approximately 4 hours. Hence, we applied the evolutionary algorithm for topology test on a larger domain (with maximum number of nodes set to 1000). As shown in Figure 4, the evolutionary algorithm converged after 172 generation, (4.75 hours). The mortality prediction performance of the learned DBN was compared against a number of benchmarking methods:the Support Vector Machine (SVM) and the Gradient Boosting Model (GBM).

We observe in Table I that the optimized DBN achieved a modest improvement in classification accuracy compared to the SVM and Gradient Boosting baselines. Importantly, our results in Figure 3 show that without topological optimization, the DBN may actually under-perform both baselines. These results highlight the importance of global optimization approaches when utilizing deep learning procedures. Indeed, with further optimization of the network topology, we expect even greater improvements in model performance. For example, In this study we assumed that the network was fully connected, and that each layer share the same activation function. In future studies, we may expand the evolutionary optimization algorithm to identify the connectivity and activation functions of the network as well.

IV. Conclusions and Future Explorations

DBN (or deep learning networks in general) can be a promising method to model and extract the implicit correlations among clinical variables when utilized correctly. In this work, we demonstrated how to use an evolutionary algorithm to search for an optimum network topology. This work is meant to serve as a simple illustration of how techniques in global optimization can be used to arrive at a correct network topology. The same technique may be used to derive optimal topological structures far nuanced than the number of nodes in a two layer system. Indeed, the number of layers, their connectivity structure, number of nodes and activation functions for each node can and should all be selected via global optimization procedures. We hope that our work here inspires other users of deep learning in the clinical informatics community to more robustly select their network topology, generating better performance in important clinical tasks, and bringing an era of deep learning in medicine one step closer.

TABLE I

Comparison of prediction accuracy and required run time for the DBN with GA topology optimization, SVM, and Gradient Boosting Model.

Model	Accuracy(%)	RunTime (hrs)
DBN (with GA)	86.0	4.75
SVM	84.0	1.04
Gradient Boosting	85.5	0.57

References

[1] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[2] R. Miotto, L. Li, and J. T. Dudley, "Deep learning to predict patient future diseases from the electronic health records," in *Advances in Information Retrieval*. Springer, 2016, pp. 768–774.

[3] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *arXiv preprint arXiv:1505.03540*, 2015.

[4] C. Esteban, O. Staeck, Y. Yang, and V. Tresp, "Predicting clinical events by combining static and dynamic information using recurrent neural networks," *arXiv preprint arXiv:1602.02685*, 2016.

[5] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[6] D. J. Hsu, M. Feng, R. Kothari, H. Zhou, K. P. Chen, and L. A. Celi, "The association between indwelling arterial catheters and mortality in hemodynamically stable patients with respiratory failure: A propensity score analysis," *CHEST*, vol. 18, no. 6, pp. 1470–1476, 2015.

[7] L. A. Celi, S. Galvin, G. Davidzon, J. Lee, D. Scott, and R. Mark, "A database-driven decision support system: Customized mortality prediction," *Journal of Personalized Medicine*, vol. 2, no. 4, p. 138, 2012.

[8] J. Le Gall, S. Lemeshow, and F. Saulnier, "A new simplified acute physiology score (saps ii) based on a european/north american multicenter study," *JAMA*, vol. 270, no. 24, pp. 2957–2963, 1993.

[9] J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, and L. Thijs, "The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure," *Intensive care medicine*, vol. 22, no. 7, pp. 707–710, 1996.

[10] C. van Walraven, P. C. Austin, A. Jennings, H. Quan, and A. J. Forster, "A modification of the Elixhauser comorbidity measures into a point system for hospital death using administrative data." *Medical care*, vol. 47, no. 6, pp. 626–33, Jun. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19433995

[11] M. A. Keyvanrad and M. M. Homayounpour, "A brief survey on deep belief networks and introducing a new object oriented MATLAB toolbox (deebnet)," *CoRR*, vol. abs/1408.3264, 2014.

[12] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, no. 7, pp. 1527–1554.

[14] SethBling, "Mari/o - machine learning for video games," https://www.youtube.com/watch?v=qv6UVOQ0F44, 2015, [Online; accessed 14-Mar-2016].