Distilling Knowledge from Deep Networks with Applications to Healthcare Domain

Zhengping Che, Sanjay Purushotham, Yan Liu
University of Southern California
{zche, spurusho, yanliu.cs}@usc.edu

USC UNIVERSITY OF SOUTHERN CALIFORNIA

Motivations

- Electronic Healthcare Records (EHR) data is under exponential growth.
  - New opportunities and urgent needs for discovery of meaningful data-driven representations and patterns in *Computational Phenotyping* research
- Deep Learning models have shown superior performance for robust prediction in computational phenotyping tasks [1].
  - Limited attempts on interpreting of features learned by deep learning models
  - Difficulties for clinicians in understanding and applying these models.
- Model interpretability is not only important but also **necessary** in healthcare.
  - A good interpretable model is shown to result in faster adoptability among the clinical staff and results in better quality of patient care.
  - Decision tree methods are widely employed in healthcare domain with easy interpretability but they do not achieve good performance.
- **Question**: How can we learn interpretable models from well trained deep network models?
  - Employ **mimicking ideas** suggested in recent deep learning papers, e.g., dark knowledge [2] and mimic learning [3].

Our Contributions

- **Interpretable Mimic Learning** – A simple yet effective knowledge distillation method
  - Mimic the performance of state-of-the-art deep learning models using well-known Gradient Boosting Trees (GBT).
- Extensive experiments on several deep learning architectures
  - Include state-of-the-art deep networks: Stacked denoising autoencoders (**SDA**) and Long Short Term Memory (**LSTM**).
  - Show Interpretable Mimic Learning models achieve comparable or even better performance than these deep learning models.
- Interpretable features and decision rules, learned by our Interpretable Mimic Learning models, validated by expert clinicians

Notations

- Assume each EHR data sample has static records with Q variables and temporal data of length T and P variables.
- By flattening the time series and concatenating static variables, we get an input vector $\boldsymbol{X} \in \mathbb{R}^D$ for each sample, where $D = TP + Q$.
- We can also only focus on the temporal variables, with input $\boldsymbol{X}_{ts} = (x_1, x_2, \cdots, x_T)^{\mathrm{T}} \in \mathbb{R}^{T \times P}$, where $x_t \in \mathbb{R}^P$ represents the variables at time t.
- A binary label $y \in \{0, 1\}$ which represents the patient's health state, e.g., mortality

References

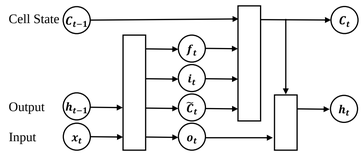
1. Che Z*, Kale D*, Li W, et al. *Deep computational phenotyping.* KDD 2015.
2. Hinton G, Vinyals O, and Dean J. *Distilling the knowledge in a neural network.* arXiv preprint 2015.
3. Ba J, and Caruana R. *Do deep nets really need to be deep?* NIPS 2014.
4. Vincent P, Larochelle H, Lajoie I, et al. *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.* JMLR 2010.
5. Hochreiter S, Schmidhuber J. *Long short-term memory.* Neural computation 1997.
6. Friedman JH. *Greedy function approximation: a gradient boosting machine.* Annals of statistics 2001.
7. Khemani RG, Conti D, Alonzo TA, et al. *Effect of tidal volume in children with acute hypoxemic respiratory failure.* Intensive care medicine 35.8 (2009).

Methodology

- Feedforward Network (DNN) and Stacked Denoising Autoencoder (SDA) [4]
  - Transformation $\boldsymbol{X}^{(l+1)}$ and reconstruction $\boldsymbol{Z}^{(l)}$ of each layer l:

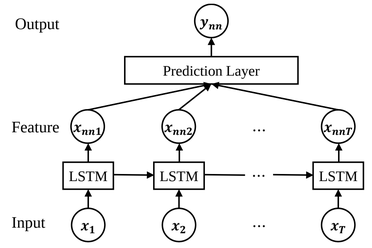
$$\boldsymbol{X}^{(l+1)} = f^{(l)}(\boldsymbol{X}^{(l)}) = s^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{X}^{(l)} + \boldsymbol{b}^{(l)}\right) \qquad \boldsymbol{Z}^{(l)} = s^{(l)}\left(\boldsymbol{W}^{(l)^{\mathrm{T}}}\boldsymbol{Z}^{(l+1)} + \boldsymbol{b}_d^{(l)}\right)$$

- Long Short-Term Memory (LSTM) [5]
  - A popular recurrent neural network model for sequential data and tasks



$$f_t = \sigma\left(\boldsymbol{W}_{fh}h_{t-1} + \boldsymbol{W}_{fx}x_t + b_f\right)$$
$$i_t = \sigma\left(\boldsymbol{W}_{ih}h_{t-1} + \boldsymbol{W}_{ix}x_t + b_i\right)$$
$$\tilde{C}_t = tanh\left(\boldsymbol{W}_{Ch}h_{t-1} + \boldsymbol{W}_{Cx}x_t + b_C\right)$$
$$o_t = \sigma\left(\boldsymbol{W}_{oh}h_{t-1} + \boldsymbol{W}_{ox}x_t + b_o\right)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$h_t = o_t * tanh(C_t) \qquad \text{(cell output, i.e. } x_{nnt}\text{)}$$

Standard LSTM Block



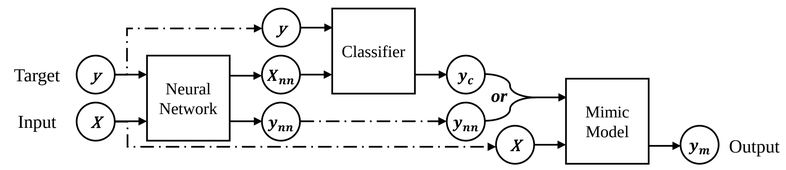
- Flatten the output from each block.
- Put a prediction layer on top of the outputs from each block.

LSTM Prediction Model

- Gradient Boosting Trees (GBT) [6]
  - An ensemble of weak learners (decision trees)
  - Find a linear combination of several functions $h(x)$ using gradient descent approaches to approximate the prediction function $F(x)$.
  - Final model with M weak learners (stages): $F_M(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const$

Interpretable Mimic Learning

- Main idea: Use GBT to mimic the performance of deep network models.



- Training Pipeline 1 (GBTmimic-LR-*)
  1. Train a deep neural network (e.g., DNN, SDA, or LSTM) on $\boldsymbol{X}$ and target y; Take the activations $\boldsymbol{X}_{nn}$ from the highest hidden layer;
  2. Train a classifier (e.g., LR) on $\boldsymbol{X}_{nn}$ and y; Take the soft prediction scores y_c;
  3. Train a mimic model (i.e., GBT) on $\boldsymbol{X}$ and y_c; Get the final output y_m.
- Training Pipeline 2 (GBTmimic-*)
  1. Train a deep neural network (e.g., DNN, SDA, or LSTM) on $\boldsymbol{X}$ and target y; Take the soft prediction scores y_{nn};
  2. Train a mimic model (i.e., GBT) on $\boldsymbol{X}$ and y_{nn}; Get the final output y_m.
- After training: Apply the mimic model in the final step to the original classification task.
- Advantages:
  - **Maintain good performance** of the original complex deep networks.
  - Avoid overfitting and **provide better generalizations** than standard decision tree methods.
  - **Provide better interpretability** than original models, from its decision rules and tree structures.

Quantitative Results

- Dataset: Acute hypoxemic respiratory failure data of 398 child patients [7]
- Two tasks:
  - MOR: Predict whether the patient dies within 60 days after admission.
  - VFD: Predict whether the patient survives and is on a ventilator for more than 14 days
    - *Ventilator free Days* is a surrogate outcome of morbidity and mortality.
- Classification Results: AUC(mean / std): Mean / Standard deviation of Area Under ROC

Other baseline methods: SVM: Support Vector Machine; LR: Logistic Regression; DT: Decision Trees

Method		Task			
		MOR		VFD	
		AUC(mean)	AUC(std)	AUC(mean)	AUC(std)
Baseline	SVM	0.6431	0.059	0.7248	0.056
	LR	0.6888	0.068	0.7602	0.053
	DT	0.5965	0.081	0.6024	0.044
	GBT	0.7233	0.065	0.7630	0.051
NN-based	DNN	0.7288	0.084	0.7756	0.053
	SDA	0.7313	0.083	0.7211	0.051
	LSTM	**0.7726**	0.062	0.7720	0.061
	LR-DNN	0.7300	0.084	0.7759	0.052
	LR-SDA	0.7459	0.068	**0.7818**	0.051
	LR-LSTM	0.7658	0.063	0.7665	0.063
Mimic	GBTmimic-DNN	0.7574	0.064	**0.7835**	0.054
	GBTmimic-SDA	0.7382	0.084	0.7194	0.049
	GBTmimic-LSTM	**0.7668**	0.059	0.7357	0.054
	GBTmimic-LR-DNN	**0.7673**	0.070	**0.7862**	0.058
	GBTmimic-LR-SDA	**0.7793**	0.066	**0.7818**	0.049
	GBTmimic-LR-LSTM	0.7555	0.067	0.7524	0.060

Interpretations

- Top useful features and corresponding importance scores

Model	Features (Importance Scores) for MOR task			
GBT	MAP-D1(0.052)	PaO2-D2(0.052)	FiO2-D3(0.037)	PH-D3(0.027)
GBT-DNN	MAP-D1(0.031)	δPF-D1(0.031)	PH-D1(0.029)	PIM2S(0.027)
GBT-SDA	OI-D1(0.036)	MAP-D1(0.032)	OI-D0(0.028)	LIS-D0(0.028)
GBT-LSTM	δPF-D1(0.058)	MAP-D1(0.053)	BE-D0(0.043)	PH-D1(0.042)
GBT-LR-DNN	δPF-D1(0.032)	PRISM12ROM(0.031)	PIM2S(0.031)	Unplanned(0.030)
GBT-LR-SDA	**PF-D0(0.036)**	**δPF-D1(0.036)**	**BE-D0(0.032)**	**MAP-D1(0.031)**
GBT-LR-LSTM	δPF-D1(0.066)	PH-D1(0.044)	MAP-D1(0.044)	PH-D3(0.041)

Model	Features (Importance Scores) for VFD task			
GBT	MAP-D1(0.035)	MAP-D3(0.033)	PRISM12ROM(0.030)	VT-D1(0.029)
GBT-DNN	MAP-D1(0.042)	PaO2-D0(0.033)	PRISM12ROM(0.032)	PIM2S(0.030)
GBT-SDA	LIS-D0(0.049)	LIS-D1(0.039)	OI-D1(0.036)	PF-D3(0.032)
GBT-LSTM	δPF-D1(0.054)	MAP-D1(0.049)	PH-D1(0.046)	BE-D0(0.040)
GBT-LR-DNN	**PaO2-D0(0.047)**	**PIM2S(0.038)**	**MAP-D1(0.038)**	**VE-D0(0.034)**
GBT-LR-SDA	PaO2-D0(0.038)	VE-D0(0.034)	PH-D3(0.030)	MAP-D1(0.030)
GBT-LR-LSTM	PH-D3(0.062)	PaO2-D0(0.055)	δPF-D1(0.043)	MAP-D1(0.037)

- Important trees from GBT (up) and GBTmimic-LR-SDA (bottom) on MOR task