DeepCare: A Deep Dynamic Memory Model for Predictive Medicine

Trang Pham, Truyen Tran, Dinh Phung and Svetha Venkatesh

Center for Pattern Recognition and Data Analytics
School of Information Technology, Deakin University, Geelong, Australia
Email: {phtra,truyen.tran,dinh.phung,svetha.venkatesh}@deakin.edu.au

Abstract. Personalized predictive medicine necessitates modeling of patient illness and care processes, which inherently have long-term temporal dependencies. Healthcare observations, recorded in electronic medical records, are episodic and irregular in time. We introduce DeepCare, a deep dynamic neural network that reads medical records and predicts future medical outcomes. At the data level, DeepCare models patient health state trajectories with explicit memory of illness. Built on Long Short-Term Memory (LSTM), DeepCare introduces time parameterizations to handle irregular timing by moderating the forgetting and consolidation of illness memory. DeepCare also incorporates medical interventions that change the course of illness and shape future medical risk. Moving up to the health state level, historical and present health states are then aggregated through multiscale temporal pooling, before passing through a neural network that estimates future outcomes. We demonstrate the efficacy of DeepCare for disease progression modeling and readmission prediction in diabetes, a chronic disease with large economic burden. The results show improved modeling and risk prediction accuracy.

1 Introduction

Health care costs are escalating. To deliver cost effective quality care, modern health systems are turning to data to predict risk and adverse events. For example, identifying patients with high risk of readmission can help hospitals to tailor suitable care packages.

Modern electronic medical records (EMRs) offer the base on which to build prognostic systems [11,15,19]. Such inquiry necessitates modeling patient-level temporal healthcare processes. But this is challenging. The records are a mixture of the illness trajectory, and the interventions and complications. Thus medical records vary in length, are inherently episodic and irregular over time. There are long-term dependencies in the data - future illness and care may depend critically on past illness and interventions. Existing methods either ignore long-term dependencies or do not adequately capture variable length [1,15,19]. Neither are they able to model temporal irregularity [14,20,22].

Addressing these open problems, we introduce DeepCare, a deep, dynamic neural network that reads medical records, infers illness states and predicts future

outcomes. DeepCare has several layers. At the bottom, we start by modeling illness-state trajectories and healthcare processes [2,7] based on Long Short-Term Memory (*LSTM*) [9,5]. LSTM is a recurrent neural network equipped with memory cells, which store previous experiences. The current medical risk states are modeled as a combination of *illness memory* and the current medical conditions and are moderated by past and current interventions. The illness memory is partly forgotten or consolidated through a mechanism known as forget gate. The LSTM can handle variable lengths with long dependencies making it an ideal model for diverse sequential domains [6,18,17]. Interestingly, LSTM has never been used in healthcare. This may be because one major difficulty is the handling irregular time and interventions.

We augment LSTM with several new mechanisms to handle the forgetting and consolidation of illness through the memory. First, the forgetting and consolidation mechanisms are time moderated. Second, interventions are modeled as a moderating factor of the current risk states and of the memory carried into the future. The resulting model is sparse and efficient where only observed records are incorporated, regardless of the irregular time spacing. At the second layer of DeepCare, episodic risk states are aggregated through a new time-decayed multi-scale pooling strategy. This allows further handling of time-modulated memory. Finally at the top layer, pooled risk states are passed through a neural network for estimating future prognosis. In short, computation steps in DeepCare can be summarized as:

$$P\left(y \mid \boldsymbol{x}_{1:n}\right) = P\left(\text{nnet}_y\left(\text{pool}\left\{\text{LSTM}(\boldsymbol{x}_{1:n})\right\}\right)\right) \tag{1}$$

where $\boldsymbol{x}_{1:n}$ is the input sequence of admission observations, y is the outcome of interest (e.g., readmission), nnet_y denotes estimate of the neural network with respect to outcome y, and P is probabilistic model of outcomes.

We demonstrate our DeepCare in answering a crucial component of the holy grail question "what happens next?". In particular, we predict the next stage of *disease progression* and the risk of *unplanned readmission* for diabetic patients after a discharge from hospital. Our cohort consists of more than 12,000 patients whose data were collected from a large regional hospital in the period of 2002 to 2013. The forecasting of future events may be considerably harder than the classical classification of objects into categories due to inherent uncertainty in unseen interleaved events. We show that DeepCare is well-suited for modeling disease progression, as well as predicting future risk.

To summarize, our main contributions are: (i) Introducing DeepCare, a deep dynamic neural network for medical prognosis. DeepCare models irregular timing and interventions within LSTM – a powerful recurrent neural networks for sequences and (ii) Demonstrating the effectiveness of DeepCare for disease progression modeling and medical risk prediction, and showing that it outperforms baselines.

2 Long Short-Term Memory

This section briefly reviews Long Short-Term Memory (LSTM), a recurrent neural network (RNN) for sequences. A LSTM is a sequence of units that share the same set of parameters. Each LSTM unit has a memory cell that has state $\boldsymbol{c}_t \in \mathbb{R}^K$ at time t. The memory is updated through reading a new input $\boldsymbol{x}_t \in \mathbb{R}^M$ and the previous output $\boldsymbol{h}_{t-1} \in \mathbb{R}^K$. Then an output states $\boldsymbol{h}_t$ is written based on the memory $\boldsymbol{c}_t$. There are 3 sigmoid gates that control the reading, writing and memory updating: input gate $\boldsymbol{i}_t$, output gate $\boldsymbol{o}_t$ and forget gates $\boldsymbol{f}_t$, respectively. The gates and states are computed as follows:

$$\boldsymbol{i}_t = \sigma\left(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{2}$$

$$\boldsymbol{f}_t = \sigma\left(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{3}$$

$$\boldsymbol{o}_t = \sigma\left(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t * \boldsymbol{c}_{t-1} + \boldsymbol{i}_t * \tanh\left(W_c \boldsymbol{x}_t + U_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{c}_t) \tag{6}$$

where σ denotes sigmoid function, $*$ denotes element-wise product, and $W_{i,f,o,c}$, $U_{i,f,o,c}$, $\boldsymbol{b}_{i,f,o,c}$ are parameters. The gates have the values in $(0, 1)$.

The memory cell plays a crucial role in memorizing past experiences. The key is the additive memory updating in Eq. (5): if $\boldsymbol{f}_t \to \mathbf{1}$ then all the past memory is preserved. Thus memory can potentially grow overtime since new experience is stilled added through the gate $\boldsymbol{i}_t$. If $\boldsymbol{f}_t \to \mathbf{0}$ then only new experience is updated (memoryless). An important property of additivity is that it helps to avoid a classic problem in standard recurrent neural networks known as vanishing/exploding gradients when t is large (says, greater than 10).

LSTM for sequence labeling. The output states $\boldsymbol{h}_t$ can be used to generate labels at time t as follows:

$$P\left(y_t = l \mid \boldsymbol{x}_{1:t}\right) = \text{softmax}\left(\boldsymbol{v}_l^\top \boldsymbol{h}_t\right) \tag{7}$$

for label specific parameters $\boldsymbol{v}_l$.

LSTM for sequence classification. LSTM can be used for classification using a simple mean-pooling strategy over all output states coupled with a differentiable loss function. For example, in the case of binary outcome $y \in \{0, 1\}$, we have:

$$P\left(y = 1 \mid \boldsymbol{x}_{1:n}\right) = \text{LR}\left(\text{pool}\left\{\text{LSTM}(\boldsymbol{x}_{1:n})\right\}\right) \tag{8}$$

where LR denotes probability estimate of the logistic regression, and $\text{pool}\left\{\boldsymbol{h}_{1:n}\right\} = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{h}_t$.

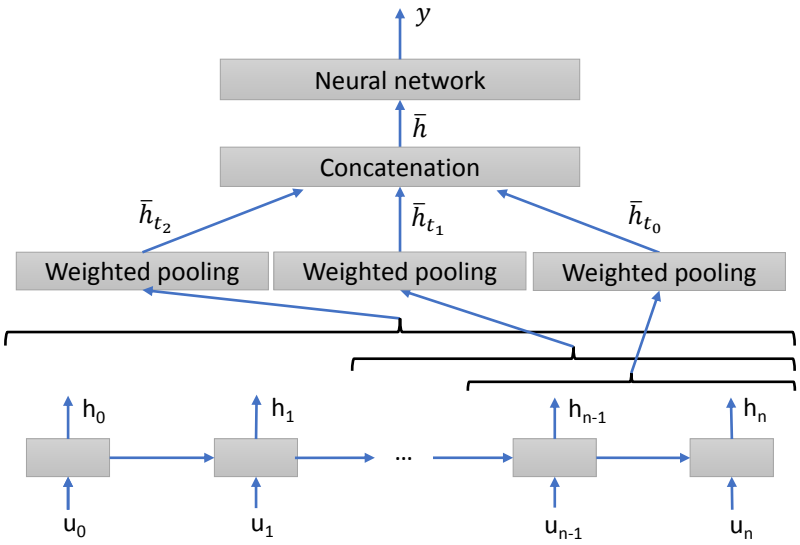

Fig. 1. DeepCare architecture. The bottom layer is Long Short-Term Memory [9] with irregular timing and interventions (see also Fig. 2b)

3 DeepCare: A Deep Dynamic Memory Model

In this section we present our contribution named DeepCare for modeling illness trajectories and predicting future outcomes. As illustrated in Fig. 1, DeepCare is a deep dynamic neural network that has three main layers. The bottom layer is built on LSTM whose memory cells are modified to handle irregular timing and interventions. More specifically, the input is a sequence of admissions. Each admission t contains a set of diagnosis codes (which is then formulated as a feature vector $\boldsymbol{x}_t \in \mathbb{R}^M$), a set of intervention codes (which is then formulated as a feature vector $\boldsymbol{p}_t$), the admission method m_t and the elapsed time $\Delta t \in \mathbb{R}^+$ between the two admission t and $t-1$. Denote by $\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_n$ the input sequence, where $\boldsymbol{u}_t = [\boldsymbol{x}_t, \boldsymbol{p}_t, m_t, \Delta t]$, the LSTM computes the corresponding sequence of distributed illness states $\boldsymbol{h}_0, \boldsymbol{h}_1, ..., \boldsymbol{h}_n$, where $\boldsymbol{h}_t \in \mathbb{R}^K$. The middle layer aggregates illness states through multiscale weighted pooling $\boldsymbol{z} = \text{pool}\{\boldsymbol{h}_0, \boldsymbol{h}_1, ..., \boldsymbol{h}_n\}$, where $\boldsymbol{z} \in \mathbb{R}^{K \times s}$ for s scales.

The top layer is a neural network that takes pooled states and other statistics to estimate the final outcome probability, as summarized in Eq. (1) as $P(y \mid \boldsymbol{x}_{1:n}) = P(\text{nnet}_y(\text{pool}\{\text{LSTM}(\boldsymbol{x}_{1:n})\}))$. The probability $P(y \mid \boldsymbol{x}_{1:n})$ depends on the nature of outputs and the choice of statistical structure. For example, for binary outcome, $P(y = 1 \mid \boldsymbol{x}_{1:n})$ is a logistic function; for multiclass outcome, $P(y \mid \boldsymbol{x}_{1:n})$ is a softmax function; and for continuous outcome, $P(y \mid \boldsymbol{x}_{1:n})$ is Gaussian. In what follows, we describe the first two layers in more detail.

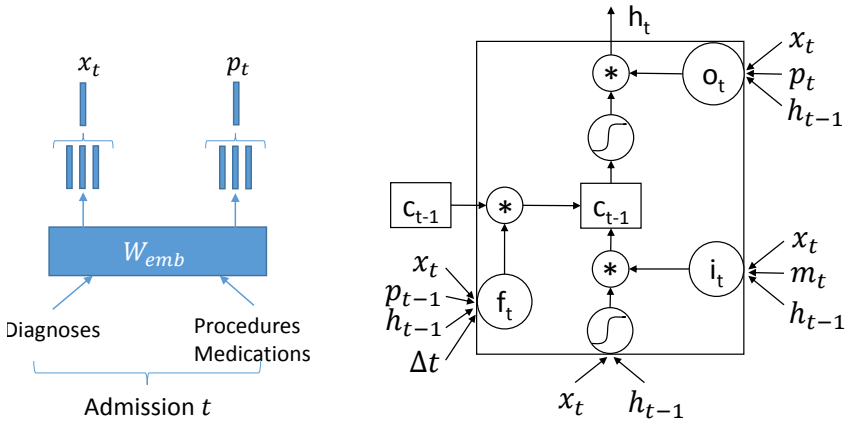

Fig. 2. (a) Admission embedding. Discrete diagnoses and interventions are embedded into 2 vectors $\boldsymbol{x}_t$ and $\boldsymbol{p}_t$. (b) Modified LSTM unit as a carrier of illness history. Compared to the original LSTM unit (Sec. 2), the modified unit models times, admission methods, diagnoses and intervention

3.1 Admission Embedding

Fig. 2a illustrates the admission embedding. There are two main types of information recorded in a typical EMR: (i) diagnoses of current condition; and (ii) interventions. Diagnoses are represented using WHO's ICD (International Classification of Diseases) coding schemes[1]. Interventions include procedures and medications. The procedures are typically coded in CPT (Current Procedural Terminology) or ICHI (International Classification of Health Interventions) schemes. Medication names can be mapped into the ATC (Anatomical Therapeutic Chemical) scheme. These schemes are hierarchical and the vocabularies are of tens of thousands in size. Thus for a problem, a suitable coding level should be used for balancing between specificity and robustness.

Codes are first embedded into a vector space of size M and embedding is learnable. Since each admission typically consists of multiple diagnoses, we average all the present vectors to derive $\boldsymbol{x}_t \in \mathbb{R}^M$. Likewise, we derive the averaged intervention vector $\boldsymbol{p}_t \in \mathbb{R}^M$. Finally, an admission embedding is a $2M$-dim vector $[\boldsymbol{x}_t, \boldsymbol{p}_t]$.

3.2 Moderating Admission Method and Effect of Interventions

There are two main types of admission: planned and unplanned. Unplanned admission refers to transfer from emergency attendance, which typically indicate higher risk. Recall from Eqs. (2,5) that the input gate $\boldsymbol{i}$ control how much new

[1] http://apps.who.int/classifications/icd10/browse/2016/en

information is updated into memory $\boldsymbol{c}$. The gate can be modified to reflect the risk level of admission type as follows:

$$\boldsymbol{i}_t = \frac{1}{m_t}\sigma\left(W_i\boldsymbol{x}_t + U_i\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{9}$$

where $m_t = 1$ if emergency admission, $m_t = 2$ if routine admission.

Since interventions are designed to cure diseases or reduce patient's illness, the output gate is moderated by the *current* intervention as follows:

$$\boldsymbol{o}_t = \sigma\left(W_o\boldsymbol{x}_t + U_o\boldsymbol{h}_{t-1} + P_o\boldsymbol{p}_t + \boldsymbol{b}_o\right) \tag{10}$$

Interventions may have long-term impacts than just reducing the current illness. This suggests the illness forgetting is moderated by *previous* intervention

$$\boldsymbol{f}_t = \sigma\left(W_f\boldsymbol{x}_t + U_f\boldsymbol{h}_{t-1} + P_f\boldsymbol{p}_{t-1} + \boldsymbol{b}_f\right) \tag{11}$$

where $\boldsymbol{p}_{t-1}$is intervention at time step $t-1$.

3.3 Capturing time irregularity

We introduce two mechanisms of forgetting the memory by modified the forget gate $\boldsymbol{f}_t$ in Eq. 11:

Time Decay Recall that the memory cell holds the current illness states, and the illness memory can be carried on to the future time. There are acute conditions that naturally reduce their effect through time. This suggests a simple decay

$$\boldsymbol{f}_t \leftarrow d(\Delta_{t-1:t})\boldsymbol{f}_t \tag{12}$$

where$\Delta_{t-1:t}$ is the time passed between step $t-1$ and step t, and $d\left(\Delta_{t-1:t}\right) \in (0, 1]$ is a decay function, i.e., it is monotonically non-increasing in time. One function we found working well is $d(\Delta_{t-1:t}) = \left[\log(e + \Delta_{t-1:t})\right]^{-1}$, where $e \approx 2.718$ is the the base of the natural logarithm.

Forgetting through Parametric Time Time decay may not capture all conditions, since some conditions can get worse, and others can be chronic. This suggests a more flexible parametric forgetting:

$$\boldsymbol{f}_t = \sigma\left(W_f\boldsymbol{x}_t + U_f\boldsymbol{h}_{t-1} + Q_f\boldsymbol{q}_{\Delta_{t-1:t}} + P_f\boldsymbol{p}_{t-1} + \boldsymbol{b}_f\right) \tag{13}$$

where $\boldsymbol{q}_{\Delta_{t-1:t}}$ is a vector derived from the time difference $\Delta_{t=1:t}$. For example, we may have: $\boldsymbol{q}_{\Delta_{t-1:t}} = \left(\Delta_{t-1:t}, \Delta^2_{t-1:t}, \Delta^3_{t-1:t}\right)$ to model the third-degree forgetting dynamics.

3.4 Recency Attention via Multiscale Pooling

Once the illness dynamics have been modeled using the memory LSTM, the next step is to aggregate the illness states to infer about the future prognosis. The simplest way is to use mean-pooling, where $\bar{\boldsymbol{h}} = \text{pool}\{\boldsymbol{h}_{1:n}\} = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{h}_t$. However, this does not reflect the attention to recency in healthcare. Here we introduce a simple attention scheme that weighs recent events more than old ones: $\bar{\boldsymbol{h}} = \left(\sum_{t=t_0}^{n} w_t \boldsymbol{h}_t\right) / \sum_{t=t_0}^{n} w_t$, where

$$w_t = [m_t + \log(1 + \Delta_{t:n})]^{-1}$$

and $\Delta_{t:n}$ is the elapsed time between the step t and the current step n, measured in months; $m_t = 1$ if emergency admission, $m_t = 2$ if routine admission. The starting time step t_0 is used to control the length of look-back in the pooling, for example, $\Delta_{t_0:n} \leq 12$ for one year look-back. Since diseases progress at different rates for different patients, we employ multiple look-backs: 12 months, 24 months, and all available history. Finally, the three pooled illness states are stacked into a vector: $\left[\bar{\boldsymbol{h}}_{12}, \bar{\boldsymbol{h}}_{24}, \bar{\boldsymbol{h}}_{all}\right]$ which is then fed to a neural network for inferring about future prognosis.

3.5 Learning

Learning is carried out through minimizing cross-entropy: $L = -\log P(y \mid \boldsymbol{x}_{1:n})$, where $P(y \mid \boldsymbol{x}_{1:n})$ is given in Eq. (1). For example, in the case of binary classification, $y \in \{0, 1\}$, we use logistic regression to represent $P(y \mid \boldsymbol{x}_{1:n})$, i.e.,

$$P(y = 1 \mid \boldsymbol{x}_{1:n}) = \sigma\left(b_y + \text{nnet}\left(\text{pool}\{\text{LSTM}(\boldsymbol{x}_{1:n})\}\right)\right)$$

where the structure inside the sigmoid is given in Eq. (1). The cross-entropy becomes: $L = -y \log \sigma - (1 - y)\log(1 - \sigma)$. Despite having a complex structure, DeepCare's loss function is fully differentiable, and thus can be minimized using standard back-propagation. The details are omitted due to space constraint.

4 Experiments

4.1 Data

The dataset is a diabetes cohort of more than 12,000 patients (55.5% males, median age 73) collected in a 12 year period 2002-2013 from a large regional Australian hospital. Data statistics are summarized in Fig. 3. The diagnoses are coded using ICD-10 scheme. For example, E10 is diabetes Type I, and E11 is diabetes Type II. Procedures are coded using the ACHI (Australian Classification of Health Interventions) scheme, and medications are mapped in ATC codes. We preprocessed by removing (i) admissions with missing key information; and (ii) patients with less than 2 admissions. This leaves 7,191 patients with 53,208 admissions. To reduce the vocabulary, we collapse diagnoses that share the first 2 characters into one diagnosis. Likewise, the first digits in the procedure block are used. In total, there are 243 diagnosis, 773 procedure and 353 medication codes.

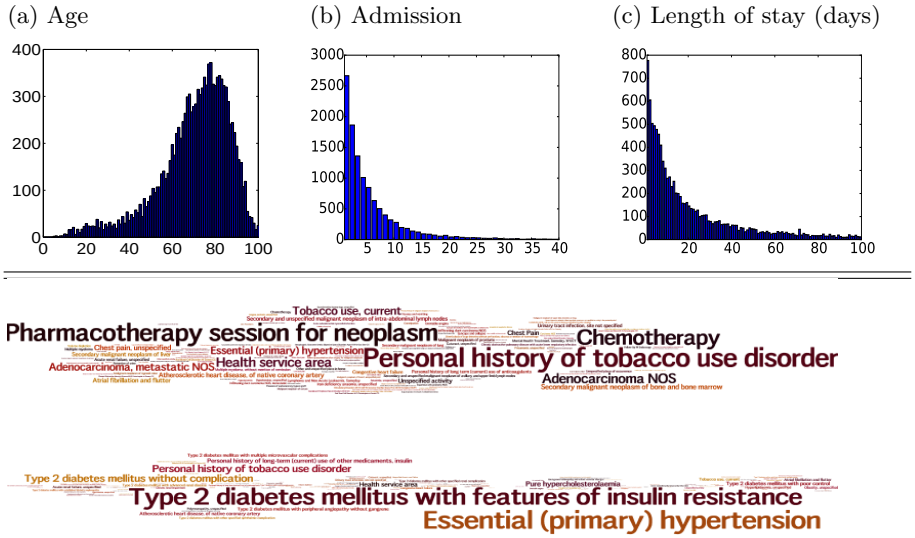

Fig. 3. Top row: Data statistics (y axis: number of patients; x axis: (a) age, (b) number of admissions, (c) number of days); **Bottom row**: Progression from pre-diabetes (upper diag. cloud) to post-diabetes (lower diag. cloud).

4.2 Implementation

The training, validation and test sets are created by randomly picking 2/3, 1/6, 1/6 data points, respectively. We vary the embedding and hidden dimensions from 5 to 50 but the results are rather robust. We report results for $M = 30$ embedding dimensions and $K = 40$ hidden units. Learning is by SGD with mini-batch of 16. Learning rate starts at 0.01. After $n_{waiting}$ epochs, if the model cannot find smaller training cost since the epoch with smallest training cost, the learning rate is divided by 2. At first, $n_{waiting} = 5$, then updated as $n_{waiting} = \min\{15, n_{waiting} + 2\}$ for each halving. Learning is terminated after $n_{epoch} = 200$ or after learning rate smaller than $\epsilon = 0.0001$.

4.3 Modeling Disease Progression

We first verify that the recurrent memory embedded in DeepCare is a realistic model of *disease progression*. We use the bottom layer of DeepCare (Secs. 3.1–3.3) to predict the next n_{pred} diagnoses at each discharge using Eq. (7).

Table 1 reports the Precision@n_{pred}. The Markov model has memoryless disease transition probabilities $P\left(d_t^i \mid d_{t+1}^j\right)$ from disease d^j to d^i at time t. Given an admission with disease subset D_t, the next disease probability is estimated as $Q\left(d^i; t\right) = \frac{1}{|D_t|} \sum_{j \in D_t} P\left(d_t^i \mid d_{t+1}^j\right)$. Using plain RNN improves over memoryless Markov model by 8.8% with $n_{pred} = 1$ and by 27.7% with $n_{pred} = 3$. Modeling irregular timing and interventions in DeepCare gains a further 2% improvement.

Model	$n_{pred} = 1$	$n_{pred} = 2$	$n_{pred} = 3$
Markov	55.1	34.1	24.3
Plain RNN	63.9	58.0	52.0
DeepCare (interven. + param. time)	**66.0**	**59.7**	**54.1**

Table 1. Precision@n_{pred} diagnoses prediction.

4.4 Predicting Unplanned Readmission

Next we demonstrate DeepCare on risk prediction. For each patient, a discharge is randomly chosen as prediction point, from which *unplanned readmission* after 12 months will be predicted. **Baselines** are SVM and Random Forests running on standard non-temporal features engineering using one-hop representation of diagnoses and intervention codes. Then pooling is applied to aggregate over all existing admissions for each patient. Two pooling strategies are tested: *max* and *sum*. Max-pooling is equivalent to the presence-only strategy in [1], and sum-pooling is akin to an uniform convolutional kernel in [20]. This feature engineering strategy is equivalent to zeros-forgetting – any risk factor occurring in the past is memorized.

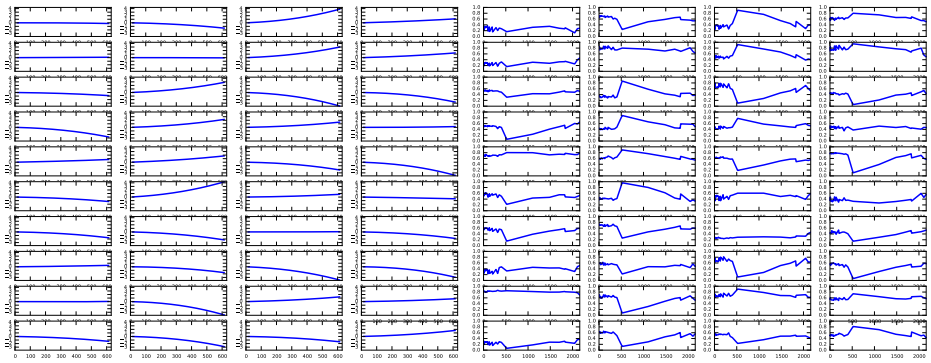

Fig. 4. (Left) 40 channels of forgetting due to time elapsed. (Right) The forget gates of a patient in the course of their illness.

Dynamics of forgetting. Fig. 4(left) plots the contribution of time into the forget gate. The contributions for all 40 states are computed using $Q_f \boldsymbol{q}_{\Delta_t}$ as in Eq. (13). There are two distinct patterns: decay and growing. This suggests that the time-based forgetting has a very small dimensionality, and we will under-parameterize time using decay only as in Eq. (12), and over-parameterize time using full parameterization as in Eq. (13). A right balance is interesting to warrant a further investigation. Fig. 4(right) shows the evolution of the forget gates through the course of illness (2000 days) for a patient.

	Model	F-score (%)
1	SVM (*max-pooling*)	64.0
2	SVM (*sum-pooling*)	66.7
3	Random Forests (*max-pooling*)	68.3
4	Random Forests (*sum-pooling*)	71.4
5	LSTM (*mean-pooling+logit. regress.*)	75.9
6	DeepCare (*mean-pooling+nnets*)	76.5
7	DeepCare ([*interven.+time decay*]*+recent.multi.pool.+nnets*)	77.1
8	**DeepCare ([*interven.+param. time*]*+recent.multi.pool.+nnets*)**	**79.1**

Table 2. Results of unplanned readmission prediction within 12 months.

Prediction results. Table 2 reports the F-scores. The best baseline (non-temporal) is Random Forests with *sum pooling* has a F-score of 71.4% [Row 4]. Using LSTM with simple mean-pooling and logistic regression already improves over best non-temporal methods by a 4.5% difference in 12-months prediction [Row 5, ref: Sec. 2]. Moving to deep models by using a neural network as classifier helps with a gain of 5.1% improvement [Row 6, ref: Eq. (1)]. By carefully modelling the irregular timing, interventions and recency+multiscale pooling, we gain 5.7% improvement [Row 7, ref: Secs. (3.2,3.3)]. Finally, with parametric time we arrive at 79.1% F-score, a 7.7% improvement over the best baselines [Row 8, ref: Secs. (3.2,3.3)].

5 Related Work and Discussion

Electronic medical records (EMRs) are the results of interleaving between the illness processes and care processes. Using EMRs for prediction has attracted a significant interest in recent year [11,19]. However, most existing methods are either based on manual feature engineering [15], simplistic extraction [20], or assuming regular timing as in dynamic Bayesian networks [16]. Irregular timing and interventions have not been adequately modeled. Nursing illness trajectory model was popularized by Strauss and Corbin [2,4], but the model is qualitative but imprecise in time [7]. Thus its predictive power is very limited. Capturing disease progression has been of great interest [10,14], and much effort has been spent on Markov models [8,22]. However, healthcare is inherently non-Markovian due to the long-term dependencies. For example, a routine admission with irrelevant medical information would destroy the illness memory [1], especially for chronic conditions.

Deep learning is currently at the center of a new revolution in making sense of a large volume of data. It has achieved great successes in cognitive domains such as vision and NLP [12]. To date, deep learning approach to healthcare has been an unrealized promise, except for several very recent work [13,3,21], where irregular timing is not property modeled. We observe that there is a considerable similarity between NLP and EMR, where diagnoses and interventions play the

role of nouns and modifiers, and an EMR is akin to a sentence. A major difference is the presence of precise timing in EMR, as well as the episodic nature. Our DeepCare contributes along that line.

DeepCare is generic and it can be implemented on existing EMR systems. For that more extensive evaluations on a variety of cohorts, sites and outcomes will be necessary. This offers opportunities for domain adaptations through parameter sharing among multiple cohorts and hospitals.

6 Conclusion

In this paper we have introduced DeepCare, a deep dynamic memory neural network for personalized healthcare. In particular, DeepCare supports prognosis from electronic medical records. DeepCare contributes to the healthcare model literature introducing the concept of *illness memory* into the nursing model of illness trajectories. To achieve precision and predictive power, DeepCare extends the classic Long Short-Term Memory by (i) parameterizing time to enable irregular timing, (ii) incorporating interventions to reflect their targeted influence in the course of illness and disease progression; (iii) using multiscale pooling over time; and finally (iv) augmenting a neural network to infer about future outcomes. We have demonstrated DeepCare on predicting next disease stages and unplanned readmission among diabetic patients. The results are competitive against current state-of-the-arts. DeepCare opens up a new principled approach to predictive medicine.

References

1. Ognjen Arandjelović. Discovering hospital admission patterns using models learnt from electronic hospital records. *Bioinformatics*, page btv508, 2015.
2. Juliet M Corbin and Anselm Strauss. A nursing model for chronic illness management based upon the trajectory framework. *Research and Theory for Nursing Practice*, 5(3):155–174, 1991.
3. Joseph Futoma, Jonathan Morris, and Joseph Lucas. A comparison of models for predicting early hospital readmissions. *Journal of biomedical informatics*, 56:229–238, 2015.
4. Bradi B Granger, Debra Moser, Barbara Germino, Joanne Harrell, and Inger Ekman. Caring for patients with chronic heart failure: The trajectory model. *European Journal of Cardiovascular Nursing*, 5(3):222–227, 2006.
5. Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
6. Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868, 2009.
7. Susan J Henly, Jean F Wyman, and Mary J Findorff. Health and illness over time: The trajectory perspective in nursing science. *Nursing research*, 60(3 Suppl):S5, 2011.

8. Rui Henriques, Cláudia Antunes, and Sara C Madeira. Generative modeling of repositories of health records for predictive tasks. *Data Mining and Knowledge Discovery*, pages 1–34, 2014.
9. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
10. Anders Boeck Jensen, Pope L Moseley, Tudor I Oprea, Sabrina Gade Ellesøe, Robert Eriksson, Henriette Schmock, Peter Bjødstrup Jensen, Lars Juhl Jensen, and Søren Brunak. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nature communications*, 5, 2014.
11. Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405, 2012.
12. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
13. Zhaohui Liang, Gang Zhang, Jimmy Xiangji Huang, and Qmming Vivian Hu. Deep learning for healthcare decision making with EMRs. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 556–559. IEEE, 2014.
14. Chuanren Liu, Fei Wang, Jianying Hu, and Hui Xiong. Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 705–714. ACM, 2015.
15. Jason Scott Mathias, Ankit Agrawal, Joe Feinglass, Andrew J Cooper, David William Baker, and Alok Choudhary. Development of a 5 year life expectancy index in older adults using predictive mining of electronic health record data. *Journal of the American Medical Informatics Association*, 20(e1):e118–e124, 2013.
16. Kalia Orphanou, Athena Stassopoulou, and Elpida Keravnou. Temporal abstraction and temporal Bayesian networks in clinical domains: A survey. *Artificial intelligence in medicine*, 60(3):133–149, 2014.
17. Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.
18. Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.
19. T. Tran, D. Phung, W. Luo, R. Harvey, M. Berk, and S. Venkatesh. An integrated framework for suicide risk prediction. In *KDD'13*, 2013.
20. Truyen Tran, Wei Luo, Dinh Phung, Sunil Gupta, Santu Rana, Richard L Kennedy, Ann Larkins, and Svetha Venkatesh. A framework for feature extraction from hospital medical data with applications in risk prediction. *BMC bioinformatics*, 15(1):6596, 2014.
21. Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM). *Journal of biomedical informatics*, 54:96–105, 2015.
22. Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.