PART 2: Learning Representations for Sequences, Images, EHR, and Text

Learned representations: topic models, CNNs, RNNs
Tricks of the trade: Data augmentation, dropout
Models that generate data: Deep Generative Models, VAE, GAN

Slides / Resources / Bibliography:
https://michaelchughes.com/mlhc2018_tutorial.html

How to represent this structured data for prediction/classification?

“21yo woman who presented with pain on the left side of her thorax”
Part 2 outline

2-stage hand-engineered representations of data
  • Bag of words for images, text, EHR codes

Learnable representations of data
  • Images
  • Time series
  • Text

• Tricks of the trade
• Models that generate data
Bag-of-words representation

Text

Friendly staff, good tacos, fresh ingredients, and fast service. What more can you look for at taco bell?

Images

Credit: Fei-Fei Li

ICD-9/ CPT Codes

09/01 emphysema
09/01 biopsy
09/01 emphysema
... 09/15 radiotherapy

count vector
over large (fixed-size) vocabulary

09/01 emphysema
09/01 biopsy
09/01 emphysema
Topic models for clinical bag-of-codes

Explain all data via set of shared clinical “topics”

Each “topic” is a distribution over all 5126 possible billing codewords

How to interpret?
• Might be a subtype of target disease (bipolar, postpartum, etc.)
• Might be related conditions
Flexible Patient Representation

Each patient’s history is a “mixture” of topics

5126 codewords

emphysema
mammography screening
radiotherapy
biopsy
tobacco use disorder
emphysema
...

≈

45 %

+ 55 %

breast cancer
0 %

45 %
cancer
0 %

asthma

55 %
tobacco use
0 %

pregnancy

emphysema
mammography screening
radiotherapy
biopsy
tobacco use disorder
emphysema
...

76
Topic Model

1) Pick number of topics $K$
2) Train to reconstruct data $\mathbf{x}$

\[\mathbf{x} \approx \pi \]

$\pi_n$ (n-th patient)

\[
\begin{bmatrix}
0 & 0.5 & 0 & 0.5 & 0
\end{bmatrix}
\]

patient-topic vector

size $K$

codeword vector

size 5126

e.g. Latent Dirichlet Allocation “LDA” (Blei, Ng & Jordan 03)
Predictions from bag-of-words

INPUT:
High-dim. codewords

\[ x \]

\[ \Rightarrow \]

\[ y \]

classify

INPUT:
Low-dim.
patient-topic vector

\[ \pi \]

\[ \Rightarrow \]

\[ y \]

classify

+ more interpretable!

Will discovered topic features make good predictions?
Why not bag-of-words?

- Tractable, but lose information (order matters)
- Cool models for low-dim. representations, but hard to integrate into predictive task

What might be better?

- Avoid two stage represent-then-predict
- Learn representations end-to-end
  - one trainable pipeline
Prediction tasks with IMAGES

• Image classification

• Object detection

• Object segmentation

For much more, see survey on Deep Learning for Medical Images: Litjens et al. 2017
Convolutional Neural Networks (CNNs): Trainable features for images

Goal: learn feature representations that:
- Represent high-level information
  - “objects” and “parts”
- Invariant to translation
  - object could appear anywhere

Credit: L.P. Morency & T. Baltrusaitis, ACL 2017 Tutorial
Basic 2D Convolution Operation

Slide same “small window” with fixed weights across entire image

Each output value depends on small subset of input

Advantages
• Fewer parameters to learn
• Can detect same pattern in any position in the image
Example Convolution in 2D

Credit: L.P. Morency & T. Baltrusaitis, ACL 2017 Tutorial
Deep CNN Example: AlexNet

Deep CNN Example: ResNet

Error vs. Depth on the ImageNet benchmark

Credit: KDD Tutorial by Sun, Xiao, & Choi: http://dl4health.org/
Figure idea originally from He et. al., CVPR 2016
Prediction tasks for TIME SERIES

Predict one label per sequence

Given vital sign history, predict mortality risk

Predict one label per timestep

Given vital sign history, predict need for ventilator at each hour

Other tasks: “seq2seq”, where x has length T and y has length U
Assumptions for Time Series ML

1) Regular time intervals between observations

- **Generative Paradigm**
  - Hidden Markov Models
    - Rabiner ‘89
  - State Space Models
    - Kalman ‘60

- **Deep Learning Paradigm**
  - Recurrent Neural Nets (RNNs)
    - LSTM
    - GRU

2) Irregular intervals?

  - EITHER Deliberately model irregularity

- **Generative Paradigm**
  - Continuous Time HMMs
    - Leiva-Murillo et al. NIPS ‘11
    - Liu et al. NIPS ‘15

- **Deep Learning Paradigm**
  - Extensions of RNNs

  - OR Align to a regular grid, then goto (1)
Assumptions for Time Series ML

1) Regular time intervals between observations

- **Generative Paradigm**
  - Hidden Markov Models
    - Rabiner ‘89
  - State Space Models
    - Kalman ‘60

- **Deep Learning Paradigm**
  - Recurrent Neural Nets (RNNs)
    - LSTM
    - GRU

Focus here on this tutorial
- Easier to integrate with prediction task
- Easier to train end-to-end

2) Irregular intervals?
- EITHER Deliberately model in

  - **Generative Paradigm**
    - Continuous Time HMMs
      - Leiva-Murillo et al. NIPS ‘11
      - Liu et al. NIPS ‘15

  - **Deep Learning Paradigm**
    - Extensions of RNNs

- OR Align to a regular grid, then goto (1)
Recurrent Neural Networks (RNNs)

Credit: Chris Olah  http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Simple RNN unit

Each “A” cell shares *same* weight parameters

*Credit: Chris Olah*  [http://colah.github.io/posts/2015-08-Understanding-LSTMs/](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
Long Short Term Memory unit (LSTM)

“cell state”
vector c

“hidden state”
vector h

Credit: Chris Olah  http://colah.github.io/posts/2015-08-Understanding-LSTMs/
LSTM: Captures long-range info.

Settings of $f$ and $i$ exist that could maintain same $c$ for any number of steps $t$

$$C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t$$

$f$ between 0 and 1
- 0 means “forget old cell value”
- 1 means “keep old cell value”

$i$ between 0 and 1
- 0 means “discard new cell value”
- 1 means “keep new cell value”

Credit: Chris Olah  [http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
Learning to Diagnose with LSTMs (Lipton et al ICLR 2016)

- 10k sequences from the Pediatric ICU
- Durations of 12 hrs to several months
- 13 vital signs (blood pressure, heart rate, etc.)
- Prediction task: label each sequence with 128 separate ICD diagnoses

Each example consists of irregularly sampled multivariate time series with both missing values and, occasionally, missing variables. We resample all time series to an hourly rate, taking the mean measurement within each one hour window.
## Performance: LSTM vs baseline

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Rate</td>
<td>0.7128</td>
<td>0.5</td>
</tr>
<tr>
<td>Log. Reg., First 6 + Last 6</td>
<td>0.8122</td>
<td>0.7404</td>
</tr>
<tr>
<td>Log. Reg., Expert features</td>
<td>0.8285</td>
<td>0.7644</td>
</tr>
<tr>
<td>MLP, First 6 + Last 6</td>
<td>0.8375</td>
<td>0.7770</td>
</tr>
<tr>
<td>MLP, Expert features</td>
<td>0.8551</td>
<td>0.8030</td>
</tr>
<tr>
<td>LSTM-DO-TR</td>
<td>0.8560</td>
<td>0.8075</td>
</tr>
<tr>
<td>Max of LSTM-DO-TR &amp; MLP</td>
<td>0.8643</td>
<td>0.8194</td>
</tr>
</tbody>
</table>

- MLP has 3 layers, layer size = 300
- LSTM has 2 layers, layer size = 128

143 “Expert features”: 11 stats for each of 13 vital signs
  - mean, min, max, median, slope, etc.
Prediction tasks with TEXT

Predict one label per sentence

Given sentence, predict mortality risk

Patient has abnormal ... of cholesterol

$\text{Predict one label per word}$

Given sentence, predict which words are relevant for a sepsis diagnosis

Patient has abnormal ... of cholesterol

Unlike time-series forecasting prediction tasks, can use bidirectional representations
Word Embeddings (word2vec)

Goal: map each word in vocabulary to high-dimensional vector
• Preserve semantic meaning in this new vector space

\[ \text{vec(swimming)} - \text{vec(swim)} + \text{vec(walk)} = \text{vec(walking)} \]
Word Embeddings (word2vec)

Goal: map each word in vocabulary to high-dimensional vector
• Preserve semantic meaning in this new vector space
How to embed?

Goal: learn weights

\[ W = \begin{bmatrix}
7.1 & & & \\
& 3.2 & & \\
& & -4.1 & \\
& & & \\
\end{bmatrix} \]

embedding dimensions

fixed vocabulary
typical 100-1000

embedding dimensions

typical 100-1000

Training

Reward embeddings that predict nearby words in the sentence.

Credit:
https://www.tensorflow.org/tutorials/representation/word2vec
Embeddings for EHR: “med2vec”

Goal: Embed patient visits in way that predicts neighboring visits

Credit: Liu & Sun ICML 2017 Tutorial
Embeddings for EHR: “med2vec”

Credit: Liu & Sun ICML 2017 Tutorial
Embeddings for clinical notes

Radiology report annotation using intelligent word embeddings: Applied to multi-institutional chest CT cohort

Imon Banerjee\textsuperscript{a,\*}, Matthew C. Chen\textsuperscript{b}, Matthew P. Lungren\textsuperscript{b,\*1}, Daniel L. Rubin\textsuperscript{a,\*1}

\textsuperscript{a} Department of Biomedical Data Science, Stanford University, Stanford, CA, United States
\textsuperscript{b} Department of Radiology, Stanford University, Stanford, CA, United States

Table 2
Clustered explored from IWE space using K-means++.

<table>
<thead>
<tr>
<th>Clusters</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 6: Hemorrhage/infection in lungs</td>
<td>‘concern’, ‘suspic’, ‘worrysom’</td>
</tr>
<tr>
<td>Cluster 7: Suspicious</td>
<td></td>
</tr>
</tbody>
</table>
Bidirectional LSTM

Elderly patient has abnormal ... of cholesterol

Hidden representation at position $t$ uses information from BOTH left and right contexts

Image Credit: Cui, Ke, and Wang 2018
https://arxiv.org/abs/1801.02143

Bidir LSTM refs:
- Schuster and Paliwal 1997
- Graves & Schmidhuber 2005
Neural Net Parts are Composable
Bidirectional LSTM + Convolutions?

Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling

Peng Zhou¹, Zhenyu Qi¹, Suncong Zheng¹, Jiaming Xu¹, Hongyun Bao¹, Bo Xu¹,²

BLSTM Layer
- left context
- word embedding
- right context

Two-dimensional Convolution Layer

Two-dimensional Max Pooling Layer

Output Layer
Part 2 outline

2-stage hand-engineered representations of data
  • Bag of words for images, text, EHR codes

Learnable representations of data
  • Images
  • Time series
  • Text

• Tricks of the trade
• Models that generate data
Trick: Early Stopping

What clinically relevant signal should we be using?

Big idea: stop training after your heldout set stops improving
  • Avoid overfitting
  • Save time / compute resources

Credit: https://deeplearning4j.org/docs/latest/deeplearning4j-nn-early-stopping
Tricks: Data Augmentation

*Data Augmentation*: Increase effective size of dataset by applying small, random perturbations to features during training.

Choose perturbations which do not change label.

<table>
<thead>
<tr>
<th>Images</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flip left-to-right</td>
<td>• Add slight misspellings</td>
</tr>
<tr>
<td>• Slight rotations or crops</td>
<td>• Replace word with similar word</td>
</tr>
<tr>
<td>• Recolor or brighten</td>
<td></td>
</tr>
</tbody>
</table>

This scheme approximately captures an important property of natural images, namely, that object identity is invariant to changes in the intensity and color of the illumination. This scheme reduces the top-1 error rate by over 1%.

from AlexNet paper (Krizhevsky et al. NIPS 2012)
Data Augmentation for Melanoma Classification

What clinically relevant process should we be using?

INCREASING DEEP LEARNING MELANOMA CLASSIFICATION BY CLASSICAL AND EXPERT KNOWLEDGE BASED IMAGE TRANSFORMS

Cristina Nader Vasconcelos*  
Departamento de Ciência da Computação  
Instituto de Computação  
Universidade Federal Fluminense, Brazil

Barbara Nader Vasconcelos  
Servico de Dermatologia  
Hospital Universitário Pedro Ernesto (Hupe)  
Universidade Estadual do Rio de Janeiro, Brazil

Fig. 2. Distortion of the original image by lesion axis analysis
Tricks: Dropout

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava
Geoffrey Hinton
Alex Krizhevsky
Ilya Sutskever
Ruslan Salakhutdinov

NITISH@CS.TORONTO.EDU
HINTON@CS.TORONTO.EDU
KRIZ@CS.TORONTO.EDU
ILYA@CS.TORONTO.EDU
RSALAKHU@CS.TORONTO.EDU

Credit: Srivastava et al. JMLR 2014
Sample at train, downweight at test

In practice, often set dropout probabilities to 50% for hidden units 20% for input units

*Credit: Srivastava et al. JMLR 2014*
Dropout Benefits

Decent gains on many tasks (images, genes, sequences)
• over other regularization (L1/L2) and other models

• MNIST images

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Classification error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
<td>1.62</td>
</tr>
<tr>
<td>L2 + L1 applied towards the end of training</td>
<td>1.60</td>
</tr>
<tr>
<td>L2 + KL-sparsity</td>
<td>1.55</td>
</tr>
<tr>
<td>Max-norm</td>
<td>1.35</td>
</tr>
<tr>
<td>Dropout + L2</td>
<td>1.25</td>
</tr>
<tr>
<td>Dropout + Max-norm</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Table 9: Comparison of different regularization methods on MNIST.

• Genetics

<table>
<thead>
<tr>
<th>Method</th>
<th>Code Quality (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network (early stopping) (Xiong et al., 2011)</td>
<td>440</td>
</tr>
<tr>
<td>Regression, PCA (Xiong et al., 2011)</td>
<td>463</td>
</tr>
<tr>
<td>SVM, PCA (Xiong et al., 2011)</td>
<td>487</td>
</tr>
<tr>
<td>Neural Network with dropout</td>
<td>567</td>
</tr>
<tr>
<td>Bayesian Neural Network (Xiong et al., 2011)</td>
<td>623</td>
</tr>
</tbody>
</table>

Table 8: Results on the Alternative Splicing Data Set.

Credit: Srivastava et al. JMLR 2014
Tricks: Label/Target Replication

Lipton et al. 2016: “Learning to Diagnose with LSTMs”

At test time, make one prediction per sequence

At train time, duplicate target label at every timestep

\[
\alpha \cdot \frac{1}{T} \sum_{t=1}^{T} \text{loss}(\hat{y}^{(t)}, y^{(t)}) + (1 - \alpha) \cdot \text{loss}(\hat{y}^{(T)}, y^{(T)})
\]
Tricks: Label/Target Replication

Lipton et al. 2016: “Learning to Diagnose with LSTMs”

AUC scores for 128 separate binary diagnostic predictions

<table>
<thead>
<tr>
<th>LSTM Models with Dropout (probability 0.5)</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-DO</td>
<td>0.8377</td>
<td>0.7741</td>
</tr>
<tr>
<td>LSTM-DO-TR</td>
<td><strong>0.8560</strong></td>
<td><strong>0.8075</strong></td>
</tr>
</tbody>
</table>

Adding TR leads to modest improvements
Part 2 outline

2-stage hand-engineered representations of data
  • Bag of words for images, text, EHR codes

Learnable representations of data
  • Images
  • Time series
  • Text

• Tricks of the trade

• Models that generate data
Autoencoder Neural Networks

Goal:  
Compress but retain information! 
- Encode each input feature vector into low-dim. vector 
- Decode back into feature vector with little information loss

Use cases: images, text, EHR, etc.  
- Use low-dim features for prediction 
- Inspect low-dim features for patterns 
- Easy storage / fast processing

Training: Optimize encoder & decoder weights to minimize reconstruction error
Denoising Autoencoders

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders, ICML’ 08

Goal: Improve robustness by adding noise to data when training

\[ x_n \rightarrow \tilde{x}_n \rightarrow z_n \rightarrow x'_n \]

Important: Noise process should match input domain

Training: Optimize encoder & decoder weights to minimize reconstruction error of clean input
Autoencoders for EHR

Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records

Riccardo Miotto\textsuperscript{1,2,3}, Li Li\textsuperscript{1,2,3}, Brian A. Kidd\textsuperscript{1,2,3}, Joel T. Dudley\textsuperscript{1,2,3}

Two stage training:
• Learn autoencoder
• Learn predictor from autoencoder

Noise
“Masking”
Set 5% random entries to 0

<table>
<thead>
<tr>
<th>Time Interval = 1 year (76,214 patients)</th>
<th>RawFeat</th>
<th>PCA</th>
<th>DeepPatient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diabetes mellitus with complications</td>
<td>0.794</td>
<td>0.861</td>
<td>0.907</td>
</tr>
<tr>
<td>Cancer of rectum and anus</td>
<td>0.863</td>
<td>0.821</td>
<td>0.887</td>
</tr>
<tr>
<td>Cancer of liver and intrahepatic bile duct</td>
<td>0.830</td>
<td>0.867</td>
<td>0.886</td>
</tr>
<tr>
<td>Regional enteritis and ulcerative colitis</td>
<td>0.814</td>
<td>0.843</td>
<td>0.870</td>
</tr>
<tr>
<td>Congestive heart failure (non-hypertensive)</td>
<td>0.808</td>
<td>0.808</td>
<td>0.865</td>
</tr>
</tbody>
</table>

Raw : size 41072
PCA : size 100*
DeepPatient : size 500

* Best size on validation set(?)
Generative Models

Raw Data  Gaussian Mixture  Gaussian Mixture
+ Neural Net likelihood

Credit: Johnson et al. NIPS 2016
# Generative Models

## Classic Generative Models

**e.g.** Gaussian mixtures & extensions

**PRO**
- Identify outliers/anomalies
- Estimate uncertainty
- Can use data with missing values

**CON**
- Bespoke inference (1+ months for algo. for each new model)
- Limited expressivity: using classic distribution building blocks like Gaussians

## Deep Generative Models

**e.g.** “deep” Gaussian mixtures

**PRO**
- Benefits of classic framework, plus
- Flexible data generation
- Black-box inference

**CON**
- *Is inference good enough?*
- Interpretation?
Variational Autoencoders (VAEs)

Goal: train deep generative models
- “AE”: Each data example \textit{sampled} from latent space
- “V”: Use variational inference to approximate posterior

Big idea: We can learn \textit{distributions} over possible embeddings
- Patient with long history has more certain embedding
- Patient with little history could take many possible values

Credit: David Blei
How to build high-quality generative models?

Training Data
(CelebA)
Generative Adversarial Net (GAN)

3.5 Years of Progress on Faces

2014 2015 2016 2017

(Brundage et al, 2018)
How do GANs work?

Two player “game”

**Discriminator:**
Identify if feature vector comes from training data or not

**Generator:**
Turn low-dim random noise into “plausible” data vectors

![Diagram](image)

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}}(x) \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z}(z) \left[ \log(1 - D(G(z))) \right]$$
medGAN for Health Data

Ask doc: How realistic is the record?

Scale 1-10

Outliers:
* medGAN sometimes generates records with both male and female gender-specific codes
End of Part 2: Best Practice Summary

Do: End-to-end training of representations if your goal is prediction quality

Do: Use clinical knowledge to improve tricks like dropout, early stopping, data augmentation

Do NOT: Blindly trust reproducibility of published methods