What is Machine Learning and How Can It Be Leveraged in Anesthesiology?

Christine Lee, MS
PhD Candidate, Department of Biomedical Engineering, UC Irvine
Senior Engineer, Algorithms-Applied ML Group, Edwards Lifesciences
What is Machine Learning?

- Sub-field of computer science
- Creating algorithms that can learn from data and make predictions on new data
- Supervised vs Unsupervised Learning
- “Machine learning is the science of getting computers to act without being explicitly programmed.” - Andrew Ng, VP & Chief Scientist of Baidu; Co-Chairman and Co-Founder of Coursera; and an Adjunct Professor at Stanford University
ML in Healthcare

Machine-learning system could aid critical decisions in sepsis care
Model predicts whether ER patients suffering from sepsis urgently need a change in therapy.

Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks

Today’s decisions deserve tomorrow’s insights.
At Verily, we’re on a mission to help patients and doctors make informed decisions about their health.

FDA Nod for Edwards’ Acumen HI Software
Wildens Portfolio

Verily Projects
We are developing tools and platforms to enable more continuous health data collection for timely decision-making and effective interventions. We are running longitudinal studies to better understand ways to predict and prevent disease onset and progression. And, we are undertaking significant joint efforts with partners to actually transform the way healthcare is delivered.
Handwritten Digit Recognition with a Back-Propagation Network

AT&T Bell Laboratories, Holmdel, N. J. 07733

Figure 1: Examples of original zipcodes from the testing set.
Why now?
1. The Hardware Exists

Figure 2. The rapidly falling ILSVRC winning entry classification error rate (red line) and the growing number of entries using GPUs each year (green bars).

2. The Data Exists

> 30 million patients visit the hospital in the US alone \(^1\)

“Medical data is estimated to grow over 50-fold this decade, to 25,000 petabytes worldwide by 2020”\(^3\)
What is Medical Data?

**Images**
- MRI
- CT Scans
- X-rays

**Sparse Manual Data**
- Clinical Notes
- Labs
- Test results
- Diagnosis
- Patient history
- Interventions

**Higher Resolution Data**
- Physiological Waveforms
- Intermittent Vitals
- Genomics

**Personal Health Data**
- Sociological/Behavioral
- Fitness trackers
- Health apps
Demographic and Chart Events
(Labs, Notes, Diagnosis etc.)

Vitals

Invasive Monitoring
Non Invasive monitoring

PreOp
OR
Postop/ICU

Interv.

Hour

1 6 12 18 24 30 ...

< 1 MB

~ 500 MB

~ 1 GB

~500 MB

~ 1 GB

1 MB

~ 1 GB
ML Basics
Developing a machine learning model

1. Define the classification problem
   - Predict postoperative reintubation using intraoperative data

2. Label data [0, 1]
   - Reintubation event annotated in the EMR

3. Data preparation/preprocessing
   - Feature extraction and selection (if necessary)
   - Statistical, existing, or domain expert features
   - Feature selection like PCA or sequential forward

4. Train the model

5. Finalize the model and test performance
Developing a machine learning model

1. Define the classification problem
   - Predict postoperative reintubation using intraoperative data

2. Label data [0, 1]
   - Reintubation event annotated in the EMR

3. Data preparation/preprocessing
   - Feature extraction and selection (if necessary)
   - Statistical, existing, or domain expert features
   - Reduce dimensionality (PCA, Fourier, etc.)

4. Train the model

5. Finalize the model and test performance
Featurizing The Data
Simple 1 Patient Example

Raw ABP Signal

HR, SYS, DIA, MAP, TV, RR, ETCO2, SpO2

Hgb

Vasopressor Administration

Age, Height, Weight, Gender

Sampled at 100 Hz over 30 minutes
= 180,000 datapoints

Sampled every 1 minute over 30 minutes
= 30 datapoints * 8

Manual
2 datapoints

Manual
5 datapoints

1 Time
4 datapoints

> 180,000 datapoints TOTAL per patient!
Preprocessing and Data Limitations

Raw ABP Signal

needs to be filtered for noise
need to reduce dimensionality → overfitting

HR, SYS, DIA, MAP, TV, RR, ETCO2, SpO2

Remove outliers

Non uniformly sampled
Sparse
Timing and/or dosage may be unreliable

Hgb

Incorrect or empty entries

Vasopressor Administration

Age, Height, Weight, Gender
Preprocessing and Data Limitations

How to time align non-uniformly sampled data with uniformly sampled?

How to scale everything to N patients?

Needs to be filtered for noise
Need to reduce dimensionality → overfitting

Remove outliers

Non uniformly sampled
Sparse
Timing and/or dosage may be unreliable

Incorrect or empty entries

Raw ABP Signal

HR, SYS, DIA, MAP, TV, RR, ETCO2, SpO2

Hgb

Vasopressor Administration

Age, Height, Weight, Gender
Alignment and Scaling Data to N Patients
Take 1 representative value per feature per patient

<table>
<thead>
<tr>
<th>Minimum, Max, Mean, Std Values</th>
<th># Vasopressors Boluses</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>SYS</td>
<td>DIA</td>
<td>MAP</td>
<td>TV</td>
<td>RR</td>
</tr>
<tr>
<td>PT 1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PT 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PT 3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>...</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PT N</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Size N patients x 34 features

*Ideal for models like logistic regression*
Developing a machine learning model

1. Define the classification problem
   - Predict postoperative reintubation using intraoperative data

2. Label data [0, 1]
   - Reintubation event annotated in the EMR

3. Data preparation/preprocessing
   - Feature extraction and selection (if necessary)
   - Statistical, existing, or domain expert features
   - Feature selection like PCA or sequential forward

4. Train the model

5. Finalize the model and test performance
Training the model

N Patients

80% Training

Separate out a validation set
- Split into 80% Training : 20% Validation
- OR
- K-fold Cross Validation

Train a model
- Optimize model learning parameters with validation set

Finalize and “freeze” the model

20% Testing

Test the model on data not used in training or validation
What does it mean to train a model?

The goal is to tune the model and learn $\beta$ such that we minimize the error between predicted $y (0,1)$ and the true label $[0,1]$

$$Predicted\ y = f(x) = \frac{1}{1 + e^{- (\beta_0 + \sum_i \beta_i x)}}$$

Logistic Regression

Bias

Weight Coefficients

Values of each feature in the model

Learnable Parameters

# of features + 1

Probability between 0 and 1
What does it mean to train a model?

The goal is to tune the model and learn $\beta$ such that we minimize the error between predicted $y$ (0,1) and the true label [0,1]

$$\text{Predicted } y = f(x) = \frac{1}{1 + e^{-(\beta_0 + \sum_i \beta_i x)}}$$

Logistic Regression

Error = Predicted $y$ - True $y$

Iteratively update the values of $\beta$ in the direction of minimizing the error
Logistic Regression

Input: 

<table>
<thead>
<tr>
<th>PT 1</th>
<th>PT 2</th>
<th>PT 3</th>
<th>...</th>
<th>PT N</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

# learnable parameters = M features + 1 bias

N patients by M features

Apply logistic transformation

Output: probability of label between 0 and 1
Deep Learning
Self learns a complex representation of input data

Logistic regression is a neural network with NO hidden layers

\[ \sigma \]

\[ \sigma \]
Deep Learning-Fully Connected Feed Forward Neural Network

*Self learns a complex representation of input data*

- Model Hyperparameters
  - Number of neurons
  - Number of hidden layers
  - Batch size and epochs
  - Activation function
  - Learning rate
  - Regularization parameters

Neuron: $f(\sum_0 \beta x)$,

- $f$: activation function (ReLU or tanh)
- $\beta$: learnable parameters
- $x$: output of the previous layer

Output layer = logistic function

Recall...Logistic Regression

- $f$: logistic function
- $\beta$: learnable parameters
- $x$: input features (input layer only)

Probability of label between 0 and 1
Deep Learning-Fully Connected Feed Forward Neural Network

Self learns a complex representation of input data

- Model Hyperparameters
  - Number of neurons
  - Number of hidden layers
  - Batch size and epochs
  - Activation function
  - Learning rate
  - Regularization parameters

Neuron:
$$ f(\sum_{0}^{i} \beta x), $$

Logistic Output Layer

Probability of label between 0 and 1

Iteratively update all the values of $\beta$ in each neuron in the direction of minimizing the error
\[ \text{the complexity} = \text{number of learnable parameters (}\beta\text{).} \]
Adding Complexity
High Resolution Data over Time over N Patients

**How to time align non-uniformly sampled data with uniformly sampled?**

<table>
<thead>
<tr>
<th>Time</th>
<th>HR</th>
<th>SYS</th>
<th>DIA</th>
<th>MAP</th>
<th>TV</th>
<th>RR</th>
<th>SpO2</th>
<th>Hgb</th>
<th>Vasopressor Administered [0 or 1]</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>9</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>9</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>9</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>…</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>10</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>30 minutes</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>10</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**PT 1**

**Backfilling/Forwardfilling**  
**Binary**  
**Repeated values**

<table>
<thead>
<tr>
<th>Time</th>
<th>HR</th>
<th>SYS</th>
<th>DIA</th>
<th>MAP</th>
<th>TV</th>
<th>RR</th>
<th>SpO2</th>
<th>Hgb</th>
<th>Vasopressor Administered [0 or 1]</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>8</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>8</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>8</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>…</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>8</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>30 minutes</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>10</td>
<td>[0 or 1]</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**PT N**

**N patients x 30 timepoints x 13 features**

**Additional dimension of time**
High Resolution Waveforms

Beat Detection
Beat Quality
Extract Features

Adverse event onset
Adverse event end

Raw ABP Signal
Beat Detection
Beat Quality
Extract Features

Label

1. Take the features at each timepoint and treat as a static input (i.e. N patients by M features) → Lose time series relationship
2. Take the features as a time series as an input (i.e. N patients by T time by M channels)
High Resolution Waveforms

1. Take the features at each timepoint and treat as a static input (i.e. N patients by M features)
2. Take the features as a time series as an input (i.e. N patients by T time by M channels)
Convolutional Neural Networks
Deep Neural Networks with a Convolutional Layer

Input

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

HR Timeseries

Size 10 timepoints

Hypotensive Event

Filter 1
Weights
(Size 4 x 1)

Filter 2
Weights
(Size 4 x 1)

Convolutional Layer Output
Activation Map
Size 4 x 2

β₁ β₂ β₃ β₄

β₅ β₆ β₇ β₈

f₁,1 f₁,2 f₁,3 f₁,4
f₂,1 f₂,2 f₂,3 f₂,4
Deep Neural Networks with a Convolutional Layer

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

“Convolving” the time series by filters...multiplying by $\beta$

Filter 1
Weights
(Size 4 x 1)

Filter 2
Weights
(Size 4 x 1)

Apply Activation Function

$\beta_1 \times t_1 + \beta_2 \times t_2 + \beta_3 \times t_3 + \beta_4 \times t_4$

$\beta_5 \times t_1 + \beta_6 \times t_2 + \beta_7 \times t_3 + \beta_8 \times t_4$
Deep Neural Networks with a Convolutional Layer

Input

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

Filter 1
Weights
(Size 4 x 1)

Filter 2
Weights
(Size 4 x 1)

Apply Activation Function

Convolutional Layer Output
Activation Map
Size 4 x 2

HR Timeseries

Hypotensive Event

HR
Time
Deep Neural Networks with a Convolutional Layer

Input

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

Filter 1
Weights (Size 4 x 1)
\[ \beta_1 \, \beta_2 \, \beta_3 \, \beta_4 \]

Filter 2
Weights (Size 4 x 1)
\[ \beta_5 \, \beta_6 \, \beta_7 \, \beta_8 \]

Apply Activation Function

Convolotional Layer Output
Activation Map
Size 4 x 2

Keep moving forward in time...
Deep Neural Networks with a Convolutional Layer

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

HR Timeseries

Input

Keep moving forward in time...

Filter 1
Weights
(Size 4 x 1)

Filter 2
Weights
(Size 4 x 1)

Apply Activation Function

Convolutional Layer Output
Activation Map
Size 4 x 2
Deep Neural Networks with a Convolutional Layer

Input

1 Convolutional Layer
2 filters (kernels) of length 4 timepoints
And stride of 2 timepoints

Filter 1
Weights
(Size 4 x 1)

Filter 2
Weights
(Size 4 x 1)

Keep moving forward in time...
Deep Neural Networks with a Convolutional Layer

Convolutional Layer

- Creates Multi-dimensional and “activation maps”
- Size 4x2

Max Pooling Layer

- Halves the output
- Size 2x2

Flatten

- Into 1 Dimension
- Size 4

Deep Neural Network

Logistic function

Probability of Event

Hypotensive Event

HR

Time
Deep Neural Networks with a Convolutional Layer

Convolutional Layer
Creates Multi-dimensional and “activation maps”
Size 4x2

$\begin{bmatrix}
f_{1,1} & f_{1,2} & f_{1,3} & f_{1,4} \\
f_{2,1} & f_{2,2} & f_{2,3} & f_{2,4}
\end{bmatrix}$

Max Pooling Layer
Halves the output
Size 2x2

$\begin{bmatrix}
f_{1,2} & f_{1,3} \\
f_{2,2} & f_{2,3}
\end{bmatrix}$

Flatten Into 1 Dimension
Size 4

Hypotensive Event

Time

HR

Update ALL the $\beta$s in the direction of minimizing the error

Deep Neural Network

Logistic function

Probability of Event

Error = Predicted Probability of Event - True Label
Convolutional Neural Networks in Use Today

Facial Recognition

1 Input Image
Size 7x7 with 3 color channels

1 Convolutional Layer
2 filters of length 3x3 and a **stride** of 2

![Convolutional Neural Network Diagram](image-url)
1 Input Image
Size 7x7x3 color channels

Convolutional Layer

Max Pooling layer

Flatten

Deep Neural Network

Logistic Output layer

Label Probability of image
ML Starter Kit

- Pretty much whatever OS you want + ready to use Images with packages pre-downloaded
- **Keras** - Python API for deep learning (runs on top of TensorFlow from Google)
- **Scikit Learn** - statistical models and model analysis
- **Pandas** - pretty data structures
- **Anaconda and Jupyter notebooks** - program to keep all packages easily updated and web app that makes everything pretty
ML Starter Kit

Other Tools:
- Matlab
  - Statistics and Machine Learning Toolbox
  - Neural Network Toolbox
  - Parallel Computing Toolbox
- R

Tutorials:
- Machinelearningmastery.com
- neuralnetworksanddeeplearning.com
- CS231n: Convolutional Neural Networks for Visual Recognition
- Coursera Machine Learning by Andrew Ng
Conclusion

- Medical data can be sparse, high resolution, variable and complex
- Medical data can get BIG...the data is there for the application of ML models
- ML methods are not new
- Clinical application of “traditional methods” like logistic regression tends to focus on “low resolution” data
- Adding the extra dimension of time and also more high resolution data makes the data more complex
- Deep learning can capture that complexity
- As we move forward, the goal is to create models that merge more high resolution data with the low resolution to fully capture relationships in all patient data
Thank you!

linkedin.com/in/christineklee1
chriskl2@uci.edu
Christine_lee@Edwards.com

YOUNG RESEARCHERS WORKSHOP
3:30 = 5:30 PM, IRONWOOD B