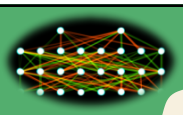


DEEP ANALYSES OF MESSY PERIOPERATIVE DATA

Parisa Rashidi, PhD
UF

HOW TO MAKE PERIOPERATIVE DATA SCIENCE MORE AFFECTIVE

THIS MORNING

Assembling a Deep EHR → Deep Learning Mechanisms → Applying Deep Insights

Tighe | Rashidi | Price

DISCLOSURES

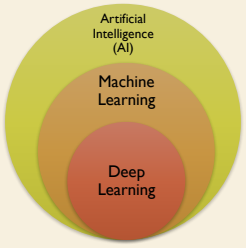
- I have *no* financial conflicts of interest to report
- Research support from NIH and NSF

UF Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

OBJECTIVES

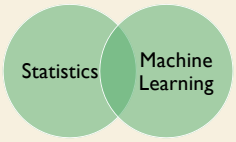
1. Learn about limitations of conventional models
2. What do deep learning models look like?
3. Deep learning in action

AI, MACHINE LEARNING, DEEP LEARNING

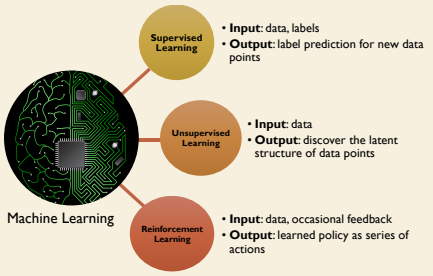


MACHINE LEARNING VS. STATISTICS

- Machine learning deals with high-dimensional data
  - Data can be unstructured
    - E.g. video, images, natural language data

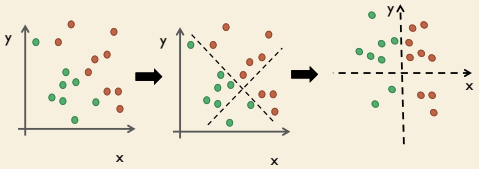


MACHINE LEARNING TASKS



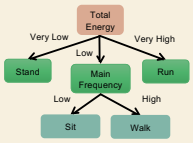
CENTRAL PROBLEM

- To transform data to learn useful representations of the input data (encode), i.e. **representation learning**

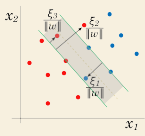


CONVENTIONAL MACHINE LEARNING MODELS

- There are already many different machine learning (ML) and statistical learning models
  - Logistic regression
  - Support Vector Machines (SVM)
  - Random forests



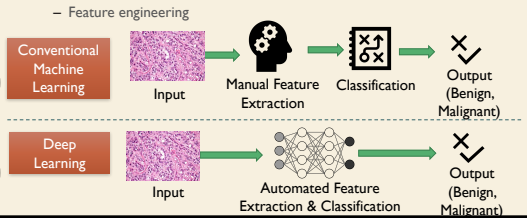
Decision Tree

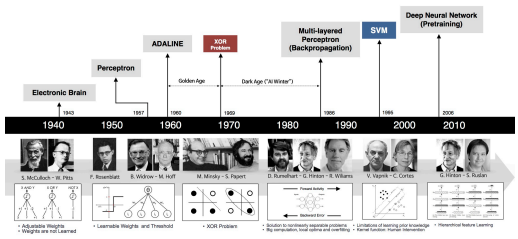


Support Vector Machine

LIMITATIONS OF CONVENTIONAL ML MODELS

- Shallow models that are unable to model highly complex data
  - E.g. video or image analysis
- Features need to be provided to the model, but what features for each problem?
  - Feature engineering

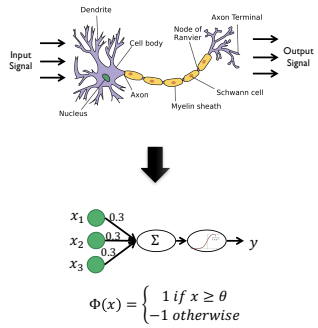




NEURAL NETWORKS HISTORY

GOES BACK TO 1940, WITH SEVERAL DARK AI WINTERS

Image from https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html



$$\Phi(x) = \begin{cases} 1 \; if \; x \geq \theta \\ -1 \; otherwise \end{cases}$$

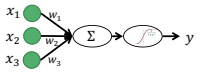
ELECTRONIC BRAIN - 1943

- Warren McCulloch and Walter Pitts published the first concept of a simplified brain cell (1943).
- A simple logic gate with binary outputs
  - Multiple signals arrive at the dendrites,
  - Signals then integrated into the cell body,
  - If the accumulated signal exceeds a certain threshold, an output signal is generated that will be passed on by the axon

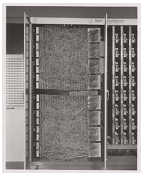
$$y = \sum_{j=1}^{d} w_j x_j + w_0 = \mathbf{w}^T \mathbf{x}$$

$$\mathbf{w} = [w_0, w_1, \ldots, w_d]^T$$

$$\mathbf{x} = [1, x_1, \ldots, x_d]^T$$

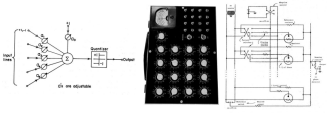


The Mark I Perceptron machine was the first implementation of the perceptron algorithm. The machine was connected to a camera that used 20×20 cadmium sulfide photocells to produce a 400-pixel image. The main visible feature is a patchboard that allowed experimentation with different combinations of input features. To the right of that are arrays of potentiometers that implemented the adaptive weights.
Wiki - Perceptron



PERCEPTRON - 1957

- The basic processing element (Rosenblat, 1957)
- Associated with each input xi is a connection weight wi
- Rosenblatt proposed an algorithm to automatically learn the optimal weights



Adaline

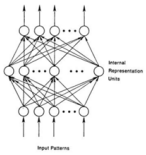
$$E(w) = \frac{1}{2}\sum_i \left(y_i - \Phi(z_i)\right)^2$$

ADALINE - 1960

- Proposed by Widrow & Hoff
- Illustrates the key concept of defining and minimizing a cost function
- The groundwork for understanding more advanced techniques
- Key difference from perceptron
  - Compared to perceptron, we can use a continuous function to compute the error
  - Differentiable

AI WINTER (1974 – 1980)

Picture & quote from HBO series, Game of Thrones



To be more specific, then, let

$$E_p = \frac{1}{2}\sum_j (y_j - o_j)^2 \qquad (1)$$

be our measure of the error on input/output pattern p, and let $E = \sum_p E_p$ be our overall measure of the error. We wish to show that the delta rule implements a gradient descent in E when the units are linear. We will proceed by simply plugging into

$$-\frac{\partial E_p}{\partial w_{ij}} = \delta_{pj} i_{pi},$$

which is proportional to $\Delta_p w_{ij}$, as prescribed by the delta rule. When there are no hidden units it is straightforward to compute the relevant derivative. For this purpose we use the chain rule to write the partial derivative of the error as the product of the rate of the error with respect to the output of the unit times the derivative of the output with respect to the weight.

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} \qquad (2)$$

The first part tells how the error changes with the output of the jth unit and the second part tells how much changing w_{ij} changes that output. Now, the derivatives are easy to compute. First, from Equation 1

$$\frac{\partial E_p}{\partial o_j} = -(y_j - o_j) = -\delta_{pj}. \qquad (3)$$

Not surprisingly, the contribution of unit o_j to the error is simply proportional to δ_{pj}. Moreover, since we have linear units,

$$o_j = \sum_i w_{ij} i_{pi}.$$

From which we conclude that

$$\frac{\partial o_j}{\partial w_{ij}} = i_{pi}.$$

Thus, substituting back into Equation 2, we see that

$$-\frac{\partial E_p}{\partial w_{ij}} = \delta_{pj} i_{pi}. \qquad (4)$$

BACKPROPAGATION - 1986

- Proposed by Rumelhart, Hinton, and Williams
- Paved the way for training deep networks

Rumelhart, Hinton, Williams (1986)

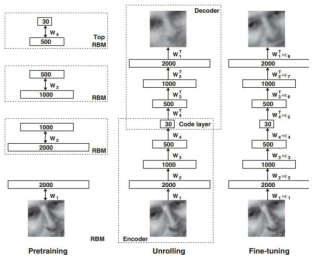
AI WINTER (1987 – 1993)

Picture & quote from HBO series, Game of Thrones

DEEP LEARNING IS REBORN

REIGNITED DEEP LEARNING IN 2006

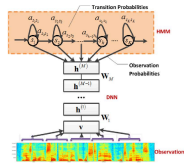
Ruslan Salakhutdinov and Geoffrey Hinton, 2006



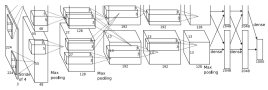
An Efficient Learning Procedure for Deep Boltzmann Machines. Ruslan Salakhutdinov and Geoffrey Hinton. Neural Computation August 2012, Vol. 24, No. 8: 1967 – 2006.

FIRST IMPRESSIVE RESULTS

STARTING IN 2010-2012



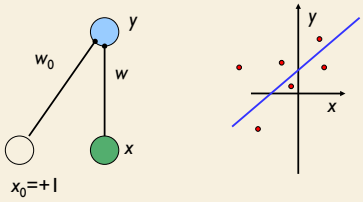
Dahl, George E., Dong Yu, Li Deng, and Alex Acero. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." IEEE Transactions on audio, speech, and language processing 20, no. 1 (2012): 30-42.



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

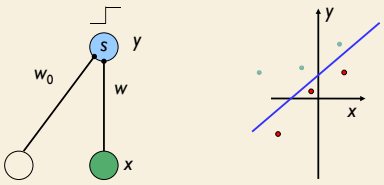
REGRESSION

$$y = \sum_{j=1}^{d} w_j x_j + w_0 = \mathbf{w}^T \mathbf{x}$$

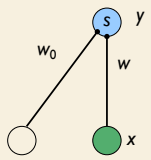


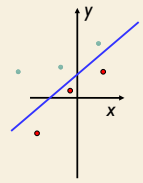
CLASSIFICATION

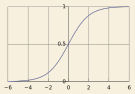
$$\Phi(x) = \begin{cases} 1 \; if \; x \geq \theta \\ -1 \; otherwise \end{cases}$$



CLASSIFICATION + POSTERIOR PROBABILITY







$$y = \text{sigmoid}\left(\mathbf{w}^T\mathbf{x}\right) = \frac{1}{1+\exp\left[-\mathbf{w}^T\mathbf{x}\right]}$$

PERCEPTRON ONLINE LEARNING

- We do not write the error function over the whole sample at once
  - But on individual instances at each step

1. Start from random weights
2. At each iteration, adjust parameters a little bit to minimize error based on current input

Learning Rate

Input Feature Value

$$\Delta w_j = \eta(y_i - \tilde{y}_i)x_i^j$$

Weight update

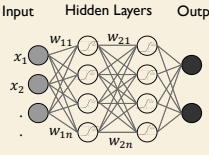
Actual value

Predicted value

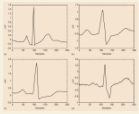
FULLY CONNECTED NETWORKS (FCN)

- Composed of multiple hidden layers
  - Every neuron in layer i is connected to every other neuron in layer i + 1

Input Hidden Layers Output



Rumelhart, Hinton, Williams (1986)



- Even simple NN can be helpful for classification:
  - Normal beat,
  - Congestive heart failure beat
  - Ventricular tachyarrhythmia beat
  - Atrial fibrillation beat

Güler, İnan. "ECG beat classifier designed by combined neural network model." *Pattern recognition* 38.2 (2005): 199-208.

GRADIENT DESCENT

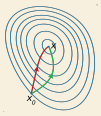
- A simple, yet powerful optimization algorithm
- Think of it as climbing down a hill until a local or global minimum is reached.
- In each iteration, we take a step away from the gradient where the step size is determined by the value of the learning rate as well as the slope of the gradient.

$\boxed{\Delta w = -\eta \nabla E(w)}$ Update

$$\frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{2} \sum_i \left(y_i - \Phi(z_i)\right)^2$$

$$= -\sum_i \left(y_i - \Phi(z_i)\right) x_i^j$$

Gradient

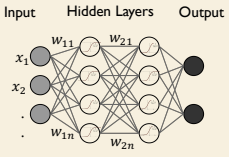


FULLY CONNECTED NETWORKS (FCN)

- Backpropagation algorithm is key

$$h_i = \sigma\left(\sum_{j=1}^{d} x_j w_{ij} + b_{ij}\right)$$

1. **Sample** a batch of data
2. **Forward** through network, compute loss
3. **Backpropagate** to calculate the gradients
4. **Update** the weights using the gradients



Rumelhart, Hinton, Williams (1986)

DEEP LEARNING

- Learning successive layers of increasingly meaningful representations
  - Mostly learned via models known as neural networks

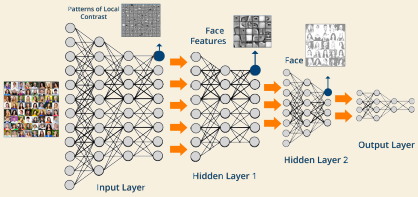
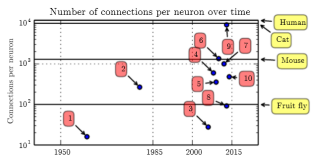


Image from https://www.edureka.co/blog/what-is-deep-learning



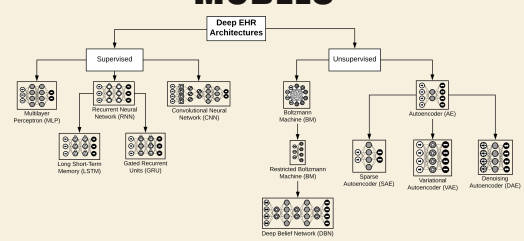
1. Adaptive linear element (Widrow and Hoff, 1960)
2. Neocognitron (Fukushima, 1980)
3. GPU-accelerated convolutional network (Chellapilla et al., 2006)
4. Deep Boltzmann machine (Salakhutdinov and Hinton, 2009a)
5. Unsupervised convolutional network (Jarrett et al., 2009)
6. GPU-accelerated multilayer perceptron (Ciresan et al., 2010)
7. Distributed autoencoder (Le et al., 2012)
8. Multi-GPU convolutional network (Krizhevsky et al., 2012)
9. COTS HPC unsupervised convolutional network (Coates et al., 2013)
10. GoogLeNet (Szegedy et al., 2014a)

INCREASING NETWORK SIZE

- It is not just about the number of neurons.
- Brain is much more complex

*Deep learning textbook, Goodfellow et al. 2015.

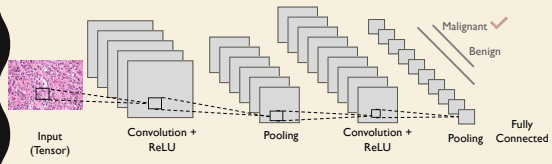
MANY DEEP LEARNING MODELS



B. Shickel, P. J. Tighe, A. Bihorac and P. Rashidi, "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," in IEEE Journal of Biomedical and Health Informatics, vol. PP, no. 99, pp. 1-1.

CONVOLUTIONAL NEURAL NETWORKS (CNN)

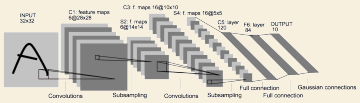
- CNN layers extract patches from input feature maps and apply the same transformation to all patches, producing an output feature map.
- Tool of choice in almost every vision tasks these days
- Many variants



CNNS HAVE BEEN AROUND FOR A LONG TIME

- Goes back to 1989, first successful application *LeNet*
- In recent years, it scaled up due to hardware improvements, algorithmic advances, and data availability

Yann LeCun

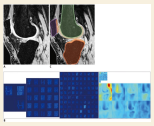


LeNet (1989)

IMPRESSIVE PROGRESS

- Error rates for ImageNet challenge have fallen from 28.5% to below 2.5% since 2010

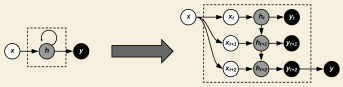
- Many biomedical imaging applications
  - Image segmentation & registration
  - Automatic labeling (e.g. lesion detection)
  - Computer-aided diagnosis



Lee, June-Goo, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Gaik Bae Kim, Joon Bom Seo, and Namkug Kim. "Deep Learning in Medical Imaging: General Overview." Korean Journal of Radiology 18, no. 4 (2017): 570-584.

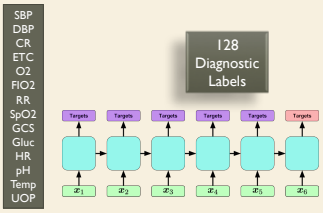
RECURRENT NEURAL NETWORKS (RNN)

- CNNs useful when data has spatial structure
- Recurrent neural networks (RNNs) useful for sequentially ordered data
  - Time series, natural language
- Popular RNN variants
  - Long short-term memory (LSTM)
  - Gated recurrent unit (GRU)



RECURRENT NEURAL NETWORKS (RNN)

- Physiological signals can be processed by LSTMs

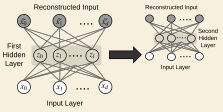


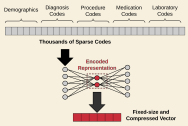
AUTOENCODERS

- Typically used for unsupervised representation learning
- To encode input into a lower dimensional space
  - Reconstruction and decoding

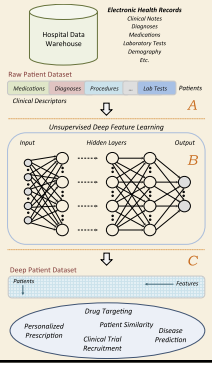
$$\tilde{x} = \sigma(W'z + b')$$

$$z = \sigma(Wx + b)$$





AUTOENCODERS

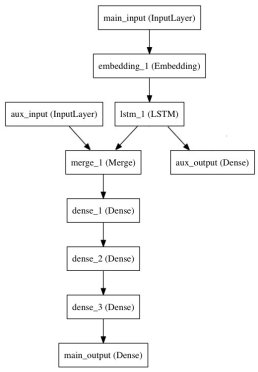


Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records

Riccardo Miotto[1,2], Li Li[1,2], Brian A. Kidd[1,2], Joel T. Dudley[1,2]

COMBINING DATA TYPES

WE CAN CREATE MULTI-INPUT AND MULTI-OUTPUT MODELS



Chollet, François. "Keras." (2015).

REINFORCEMENT LEARNING

- Which actions to take to maximize cumulative reward
- Tradeoff between exploration & exploitation

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert et al. "Mastering the game of go without human knowledge." Nature 550, no. 7676 (2017): 354.

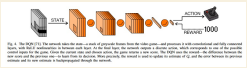


Fig. 1 The DQN CNN. The network takes the state, a stack of grayscale frames from the video game, and processes it with convolutional and fully connected layers, with ReLU nonlinearities in between each layer. At the final layer, the network outputs a discrete action, which corresponds to one of the possible control inputs for the game. Given the current state and chosen action, the game returns a new score. The DQN uses the reward—the difference between the previous score and the new score—to update its estimate of Q, and the error between its previous estimate and its new estimate is backpropagated through the network.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. "Human-level control through deep reinforcement learning." Nature 518, no. 7540 (2015): 529-533.

$$a = \pi(s)$$

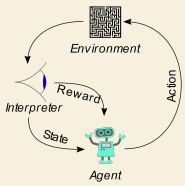


Image via Wiki – Reinforcement Learning

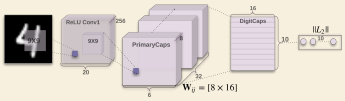
SUCCESS STORIES

- Near-human-level
  - Image classification, speech recognition, handwriting transcription
- Improved
  - Machine translation, text-to-speech, autonomous driving
- Superhuman
  - Go playing

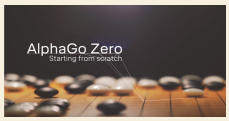
RECENT PROGRESS - 2017

- Capsule net
- One model for all
- Alpha-zero

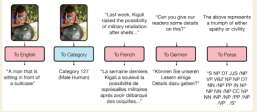
Capsule Net



Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." In Advances in Neural Information Processing Systems, pp. 3859-3869. 2017.



Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert et al. "Mastering the game of go without human knowledge." *Nature* 550, no. 7676 (2017): 354.

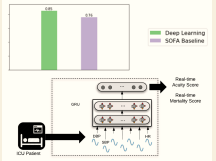


Kaiser, Lukasz, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. "One model to learn them all." arXiv preprint arXiv:1706.05137 (2017).

OUR USE CASES

- Time series: LSTM
  - SOFA Score Prediction
- Video feed: CNN
  - Functional status detection
- Text: LSTM
  - Mental health, clinical notes

AUC @ 24 hours (early detection)

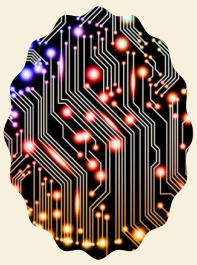


Shickel, Benjamin, Tyler Loftus, Tezcan Ozrazgat Baslanti, Azra Bihorac, and Parisa Rashidi. "1619: Increasing Sofa Score Granularity With Deep Learning." Critical Care Medicine 46, no. 1 (2018): 794.

Shickel, Benjamin, Martin Heesacker, Sherry Benton, and Parisa Rashidi. "Hashtag Healthcare: From Tweets to Mental Health Journals Using Deep Transfer Learning." *arXiv preprint arXiv:1708.01372* (2017).

AI FUTURE

- Human-level general intelligence: still far far away
- Many opportunities for AI in medicine
- Best to moderate our expectations, there is a short-term hype, but there is also a long-term vision



Keras

arXiv.org
open access
to the world's science

CNTK

TensorFlow

ANACONDA
Powered by Continuum Analytics

jupyter

mxnet

WHERE TO GO FROM HERE?

- Our recent survey paper on use of deep learning with EHR data link
- Look for the latest developments on ArXiv
- Many open software tools
  - Keras
  - TensorFlow
  - CNTK
  - Theano
  - MXNet