KCC 2017 Summer

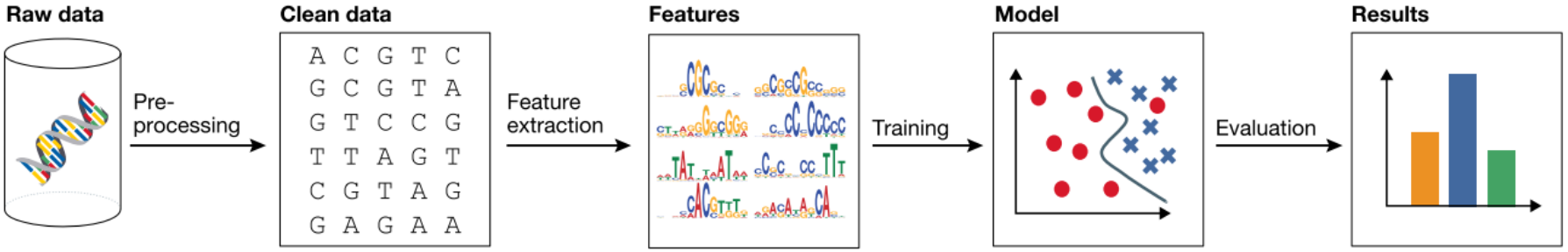
이슬 (한국뉴욕주립대)

Deep Learning in Bio-Healthcare
(딥러닝의 바이오헬스케어 응용)

1. Deep Learning Basics

Benefits of DNN Learning

Classical Machine Learning Pipeline in Comp Bio



Deep Learning in Comp Bio.

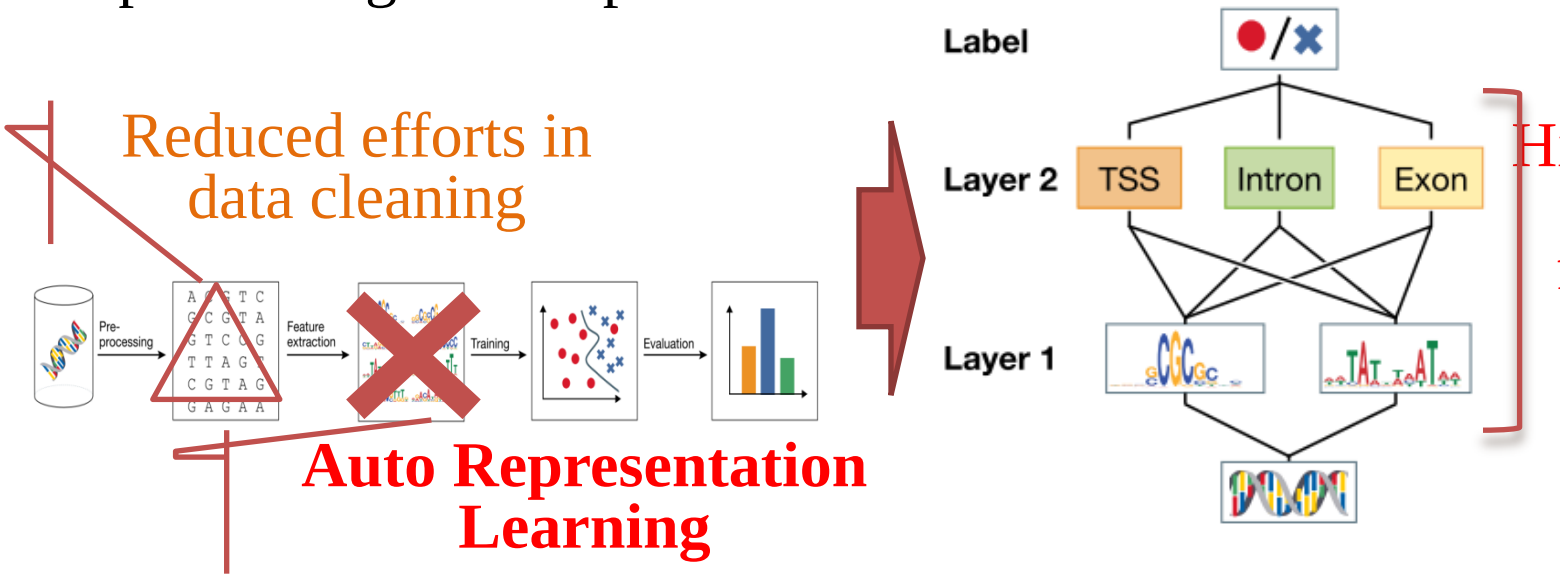


Fig 1A,D from Angermueller et al. (2016) *Molecular Systems Biology*, (12), 878.

Various Dimensions of Learning

- Computationalism vs Connectionism
- Feed Forward vs Recurrent Neural Network
- Deep Neural Network vs Deep Generative Model (Discriminative vs Generative Learning)
- Deep vs Shallow
- Supervised vs Unsupervised

Connectionism vs Computationalism

Two perspectives of cognition:

Computationalism

- World is abstracted by symbols forming specific structures and information is aggregated through this structure.
- Most traditional AI (logic; deductive)
- Focused on **search** and **representation** in a state space

Connectionism

- Aims at **massively parallel** models of consisting of large number of simple and uniform processing elements.
- Focused in **learning** (learning from data; inductive)
- Artificial neural networks

Feed Forward vs Recurrent Neural Nets

Feed Forward Neural Networks

- Connections only in one direction (directed acyclic graph)
- Implement functions, have no internal state

Recurrent Neural Networks

- Have directed cycles (feedback loops) with delays ⇒ have internal state (like flip-flops), can oscillate etc.
- Interesting models of the brain but more difficult to understand.

Discriminative vs Generative Learning

Two approaches of learning models:

Discriminative

- Directly model posterior probabilities $p(C_k|\mathbf{x})$

Generative

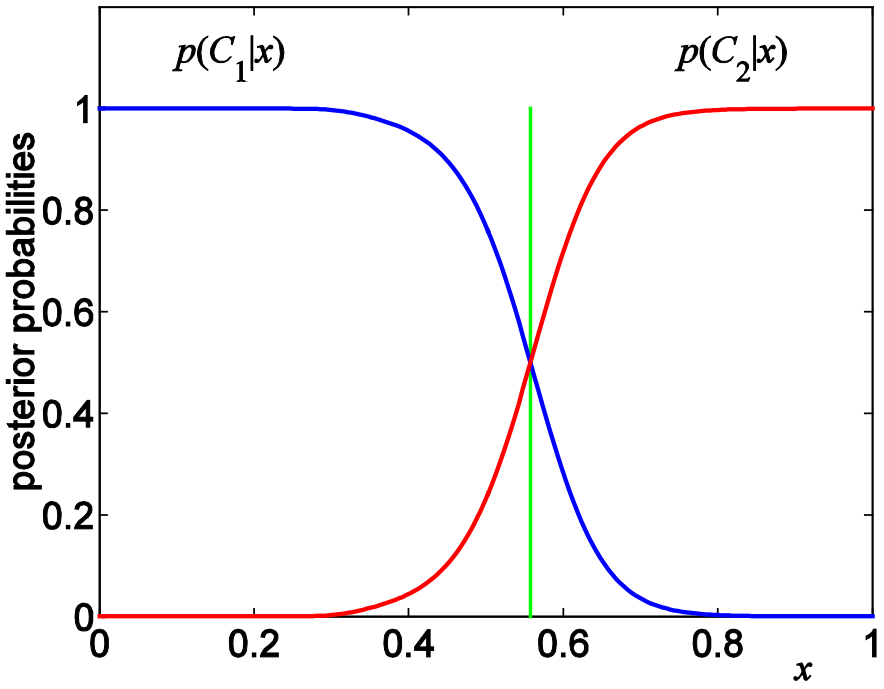
- Model class-conditional densities $p(\mathbf{x}|C_k)$ and priors $p(C_k)$ then evaluate posterior probabilities using Bayes' theorem

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)}$$

[C. Bishop 04]

Discriminative vs Generative: Function Modelled

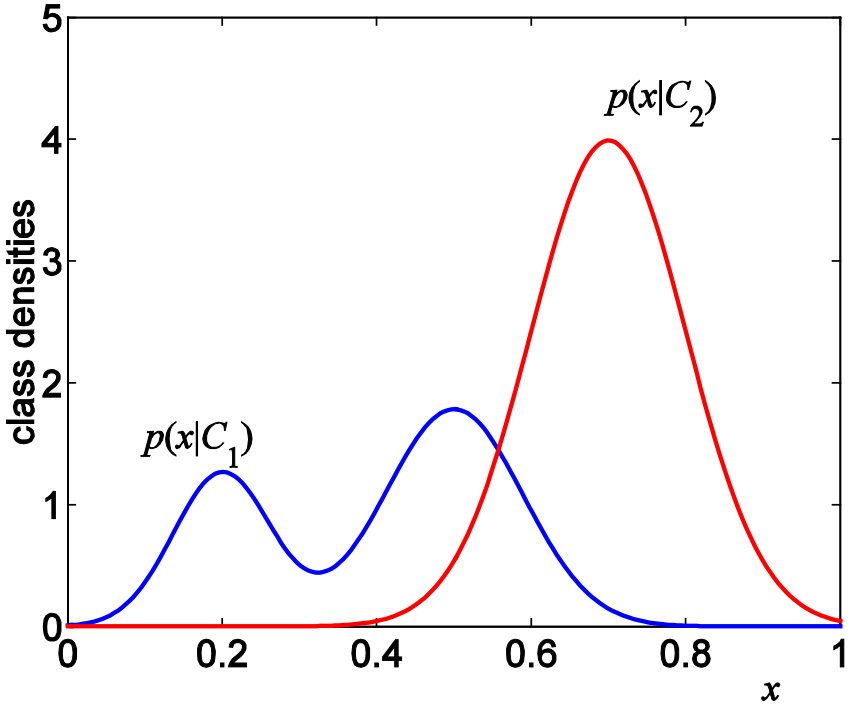
Discriminative



Learns the (hard or soft) **boundary** between classes

[C. Bishop 04]

Generative



Model the **distribution** of individual classes by estimating p(x, y) and then determining p(y|x) via Bayes rule

Discriminative vs Generative

Example Machine Learning Methods

Discriminative

- **Decision Trees**
- Boosting
- **Linear Regressions**
- Support Vector Machines (**SVM**)
- Random Forests
- Conditional Random Fields (CRF)

Generative

- **Naïve Bayes**
- **Hidden Markov Model** (HMM)
- Gaussian **Mixture Models** (GMM)
- Variational Bayes
- **Markov Random Fields** (MRF)
- Latent Drichlet allocation

[C. Bishop 04]

Generative vs. Discriminative: Pros and Cons

Discriminative

- ☺ Use flexibility of the model in relevant regions of input space
- ☺ Very **fast once trained**
- ☹ Interpolate between training examples, and hence can **fail if novel inputs are presented**
- ☹ They don't easily handle composition
  - e.g. faces can have glasses and/or moutaches and/or hats
- Sampling generally not possible

Generative

- ☺ Relatively straightforward to characterize invariances
- ☺ Can handle **partially labelled data**
- ☹ **Model variability** even if not needed
- ☹ Scale badly
  - number of classes
  - the number of invariant transformations
  - slow on test data
- Can sample from model

[C. Bishop 04]

Deep NN vs Deep Generative Model

Both exploits **layered hierarchical architectures** but are different in their goals.

Discriminative -

Deep Neural network (DNN) examples

- Multi-layer Perceptron
- Convolution Neural Net. (CNN)
- Recurrent Neural Net.

Generative -

Deep Generative Models (DGM) examples:

- Deep Belief Net. (DBN)
- Restricted Boltzmann Machines (RBM)
- Generative CNN
- Generative Adversarial Net. (GAN)

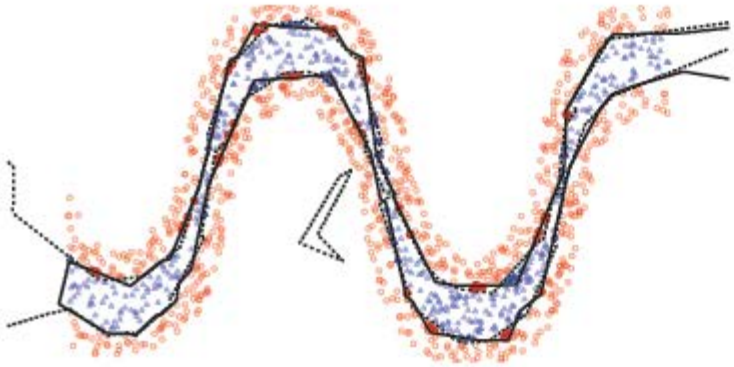
Modified from *Table 1* of [L. Deng and N. Jaitly. 2015]

	Deep Neural Network	Deep Generative Model
Structure	Graphical info flow: bottom-up	Graphical info flow: top-down
Domain knowledge	Hard	Easy
Semi/unsupervised	Harder	Easier
Interpretation	**Harder**	**Easier**
Representation	Distributed	Local or Distributed
Inference/decode	Easy	Harder
Scalability/compute	Easier	Harder
Incorp. uncertainty	Hard	Easy
Empirical goal	Classification, feature learning, etc.	Classification (via Bayes rule), latent variable inference, etc.
Learning algorithm	Backpropagation (unchallenged)	Variational EM, MCMC-based, belief propagation. etc
Evaluation	On a black-box score – end performance	On almost every intermediate quantity
Experiments	Massive, real data	Modest, often simulated data
Parameterization	Dense matrices	Sparse (often); Conditional PDFs

Making DNN models interpretable is an active ongoing research.

Montufar et a., NIPS'14

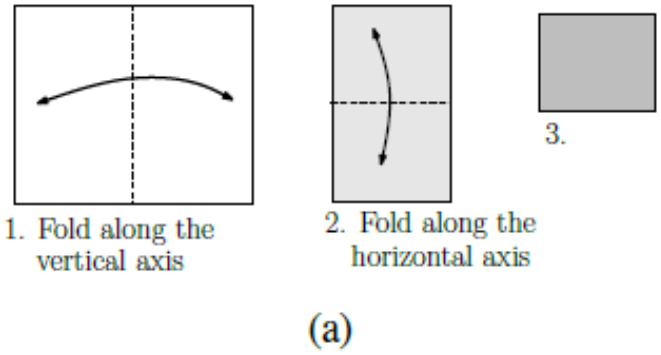
Why Deep?

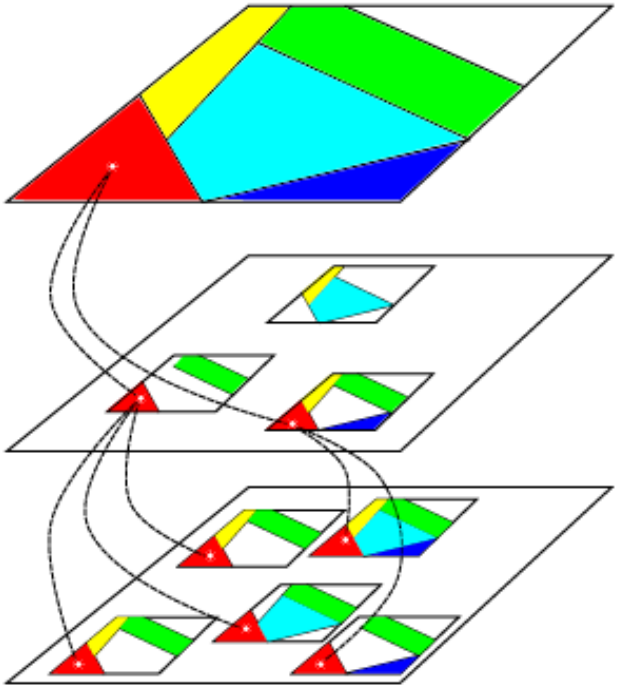


A deep network has significantly greater representational power than a shallow one.

Montufar et a., NIPS'14

Power of Layers: Space folding



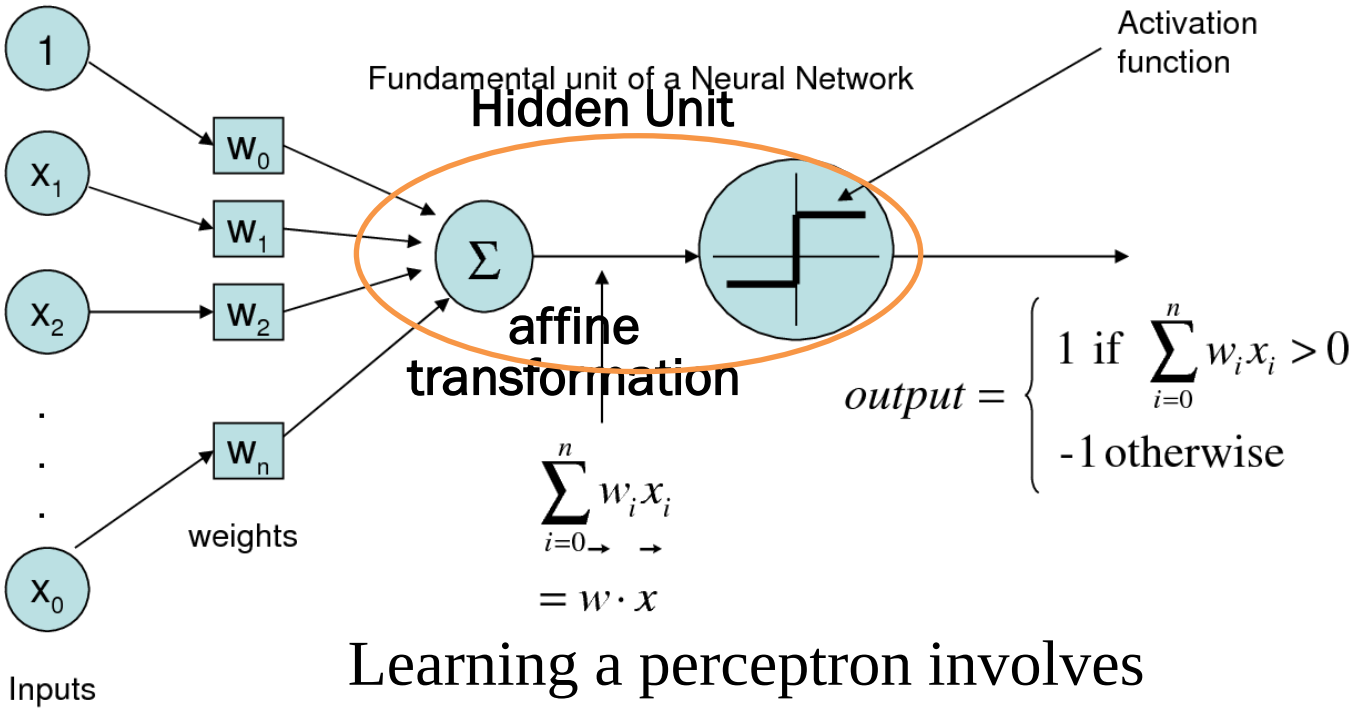


Deep Learning Models Categorized as Supervised vs Unsupervised

- Supervised Learning Methods (Classification)
  - Multi-layer perceptron
  - Convolution neural network
- Unsupervised Learning Methods (Representation Learning)
  - Restricted Boltzmann Machine
  - Deep Belief Nets (can be supervised)
  - Deep Bolzmann Machines
  - Autoencoders
- Semi-supervised Learning Method
  - Self-Taught Learning
  - Generative Adversarial Networks

Basic Unit of DNN: Perceptron

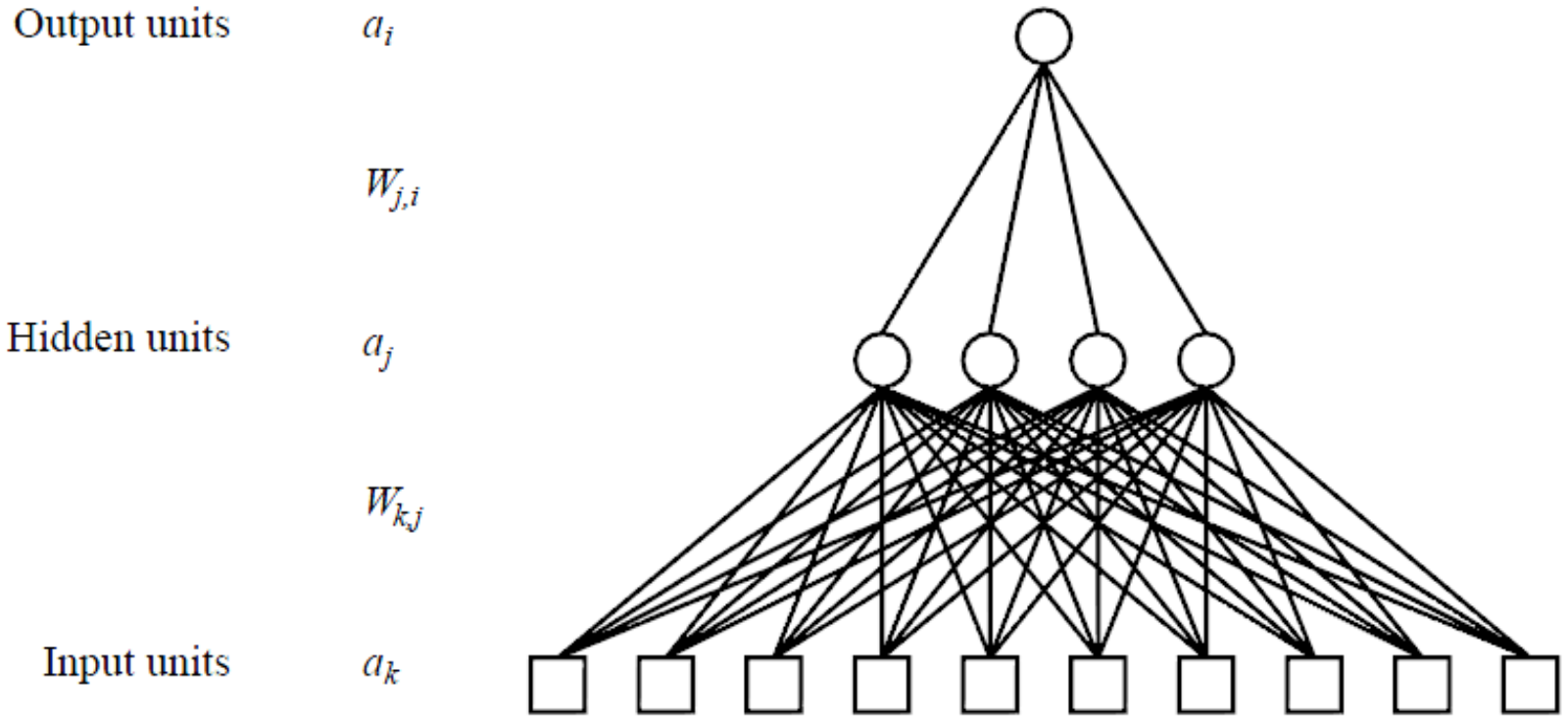
Perceptron: directed model



Learning a perceptron involves choosing values for the weights w_i

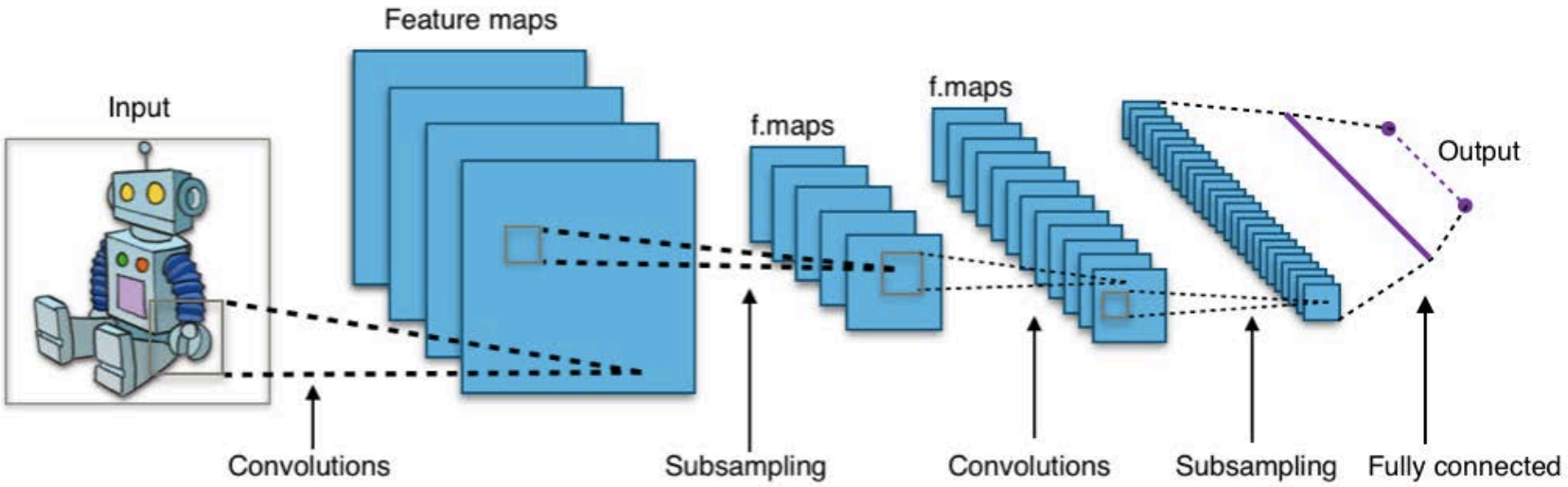
Supervised: Multi-Layer Perceptron

Layers are usually fully connected;
numbers of hidden units typically chosen by hand



Supervised: Convolution Neural Network (CNN)

- Connectivity pattern inspired by organization visual cortex.
  - Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field.
  - Receptive fields of neurons partially overlap forming tiles in visual field.
  - Response of a neuron approximated by a convolution operation



https://en.wikipedia.org/wiki/Convolutional_neural_network

Basic Unit of DGM: Restricted Boltzmann Machines

Restricted Boltzmann Machines (Harmonium):
Undirected probabilistic graphical models

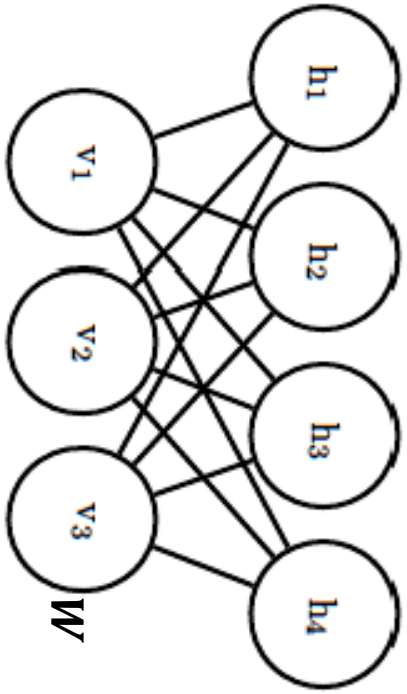
A type of **energy-based** model

$p(\mathbf{v},\mathbf{h}) = {}^{1}/_{Z} \exp(-E(\mathbf{v},\mathbf{h}))$

where $E(\mathbf{v},\mathbf{h}) = -\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{h} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{h}$

and $Z = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp\{-E(\boldsymbol{v},\boldsymbol{h})$

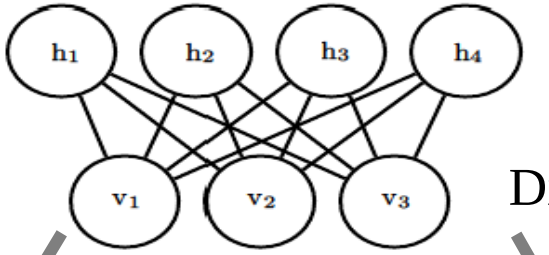
and b, c , and W are unconstrained, real-valued, learnable parameters.



bipartite graph

Unsupervised: RBM & DBN & DBM

Restricted Boltzmann Machine (RBM)

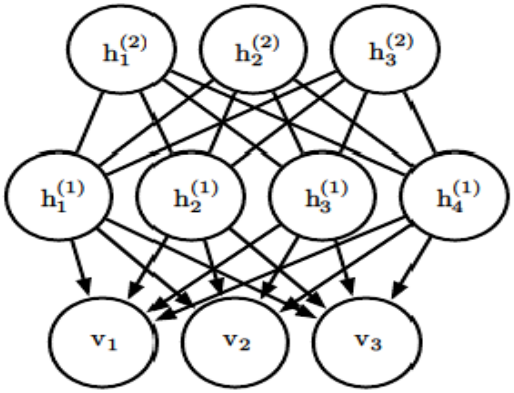


bipartite graph

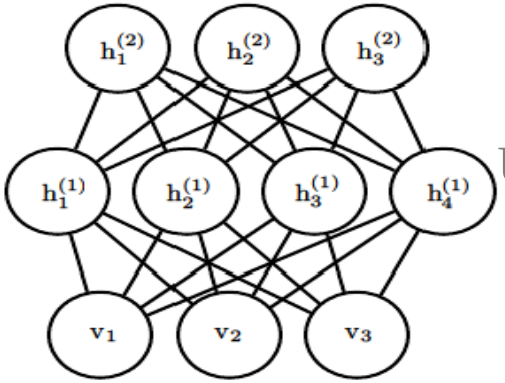
Disconnected hidden units

Disconnected visible units

Stacked RBMs



hybrid graph directed + undirected



Undirected graph

Deep Belief Network (DBN)

Deep Boltzmann Machine (DBM)

Deep, generative models

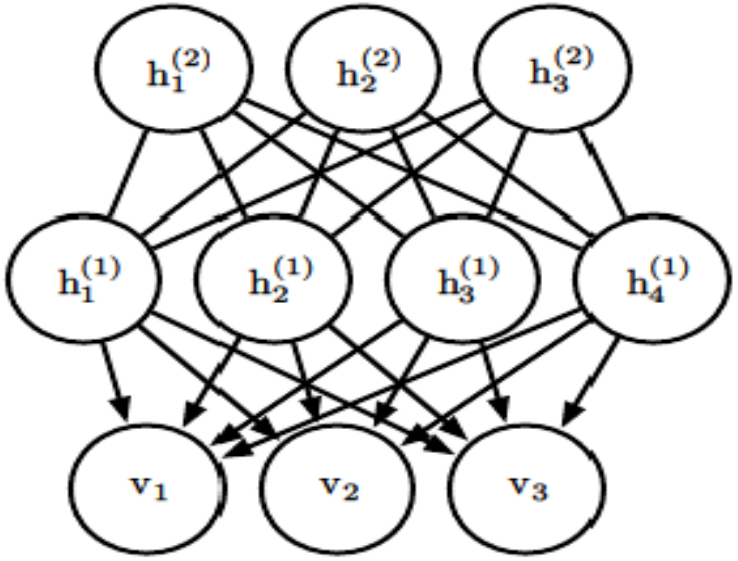
Unsupervised: Deep Belief Network (DBN)

- Characteristics: train layer by layer maximizing

$$E_{v\sim p_{data}} E_{h^{(l)}\sim p^{(l)}(h^{(l)}|v)} \log p^{(l+1)}(h^{(l)}) \text{ or}$$

$$E_{v\sim p_{data}} \log p\,(\boldsymbol{v}) \text{ if first layer}$$

*NOTE: The first of the deep learning models (2006)



undirected top two layers connections

directed connections between all other layers; arrows pointed toward the data

additional

Classification via DBN

DBN may be used directly as a generative model

But to be used as classification additional steps are needed:

- Take weights of DBN and used them to define MLP

$$h^{(1)} = \sigma\left(b^{(1)} + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{W}^{(1)}\right)$$

$$h^{(l)} = \sigma\left(b^{(1)} + \boldsymbol{v}^{(l-1)\mathrm{T}}\boldsymbol{W}^{(l)}\right) \forall\, l \in 2,..,.m$$

- Train the MLP for classification (optional)
  - Example of discriminative fine-tuning

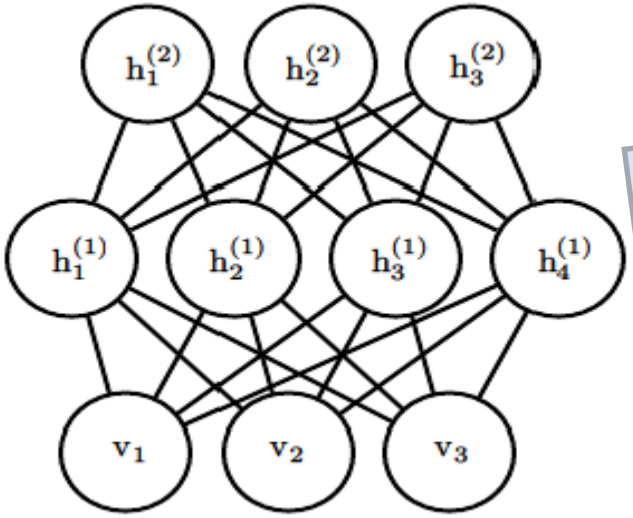
Unsupervised: Deep Boltzmann Machines

- Model variables is parametrized by an energy function E

$$P\left(v, h^{(1)}, \dots, h^{(L)}\right) = {}^{1}/_{Z(\theta)} \exp\left(-E\left(v, h^{(1)}, \dots, h^{(L)}; \theta\right)\right)$$

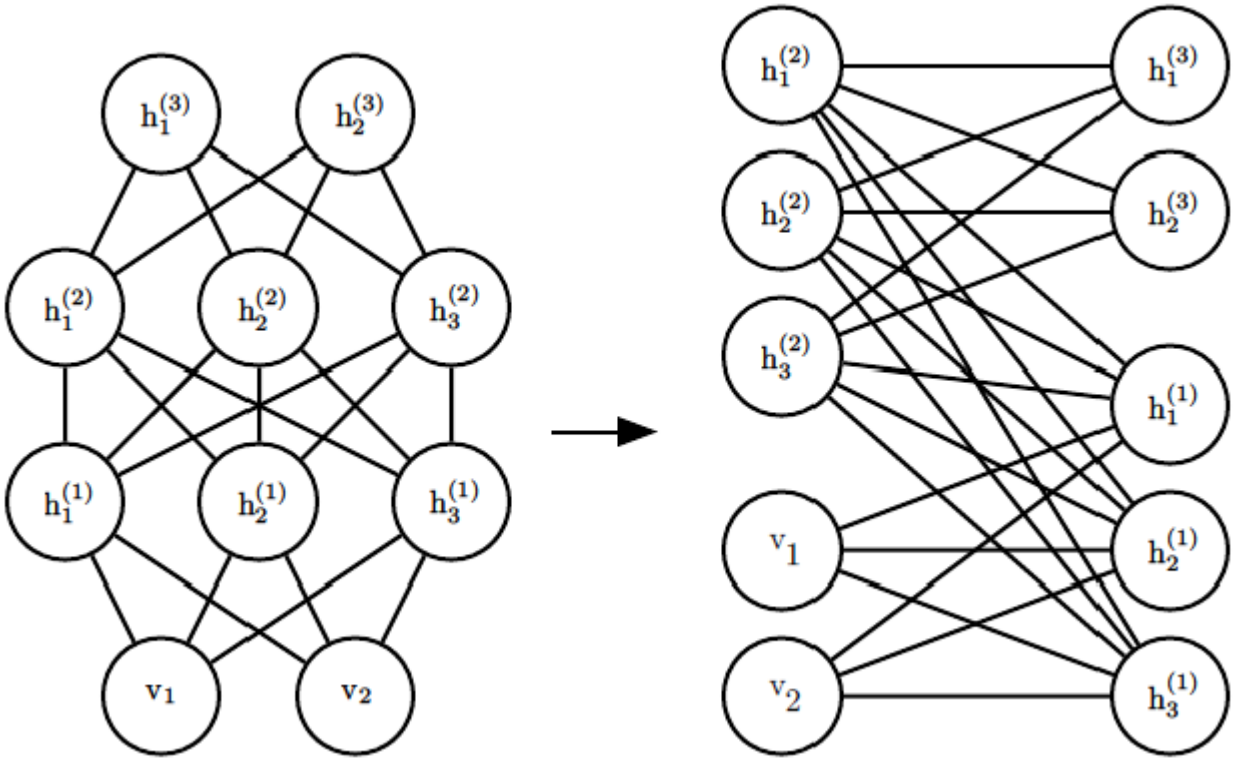
where

$$E\left(v, h^{(1)}, \dots, \boldsymbol{h}^{(L)}; \boldsymbol{\theta}\right) = -v^T W^{(1)} h^{(1)} - \sum_{l=2}^{L} h^{(l-1)^T} W^{(1)} h^{(l)}$$

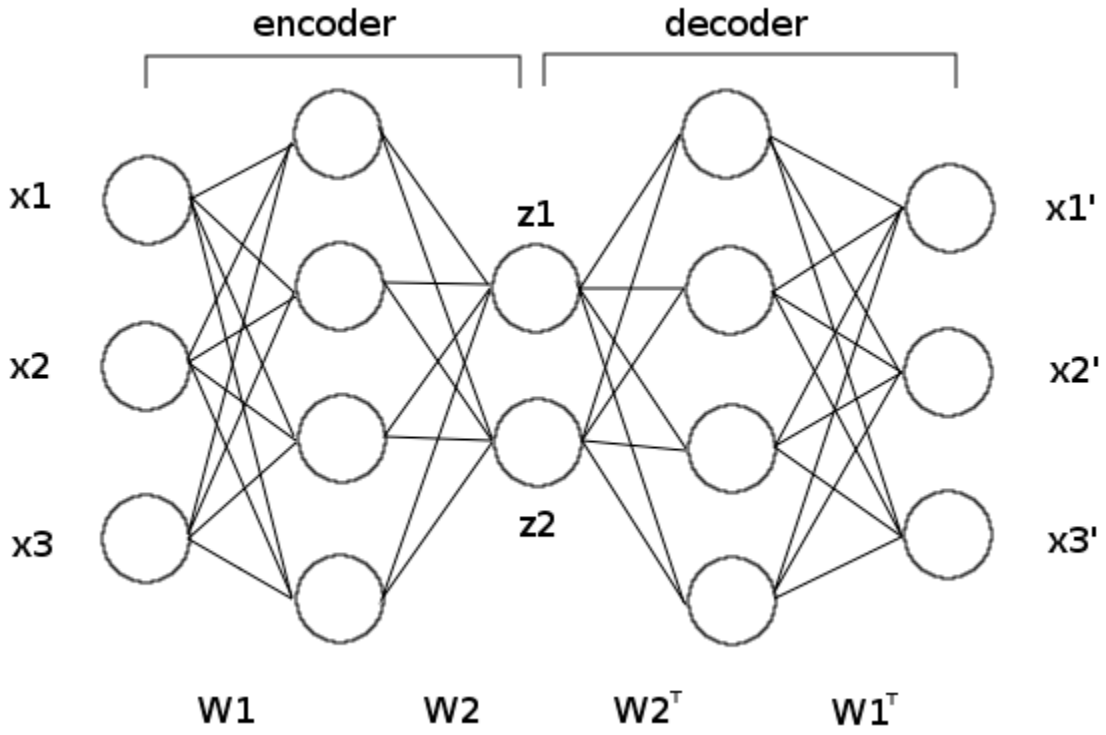


Undirected in every layer

DBM as Bipartite Graph



Unsupervised: Autoencoders

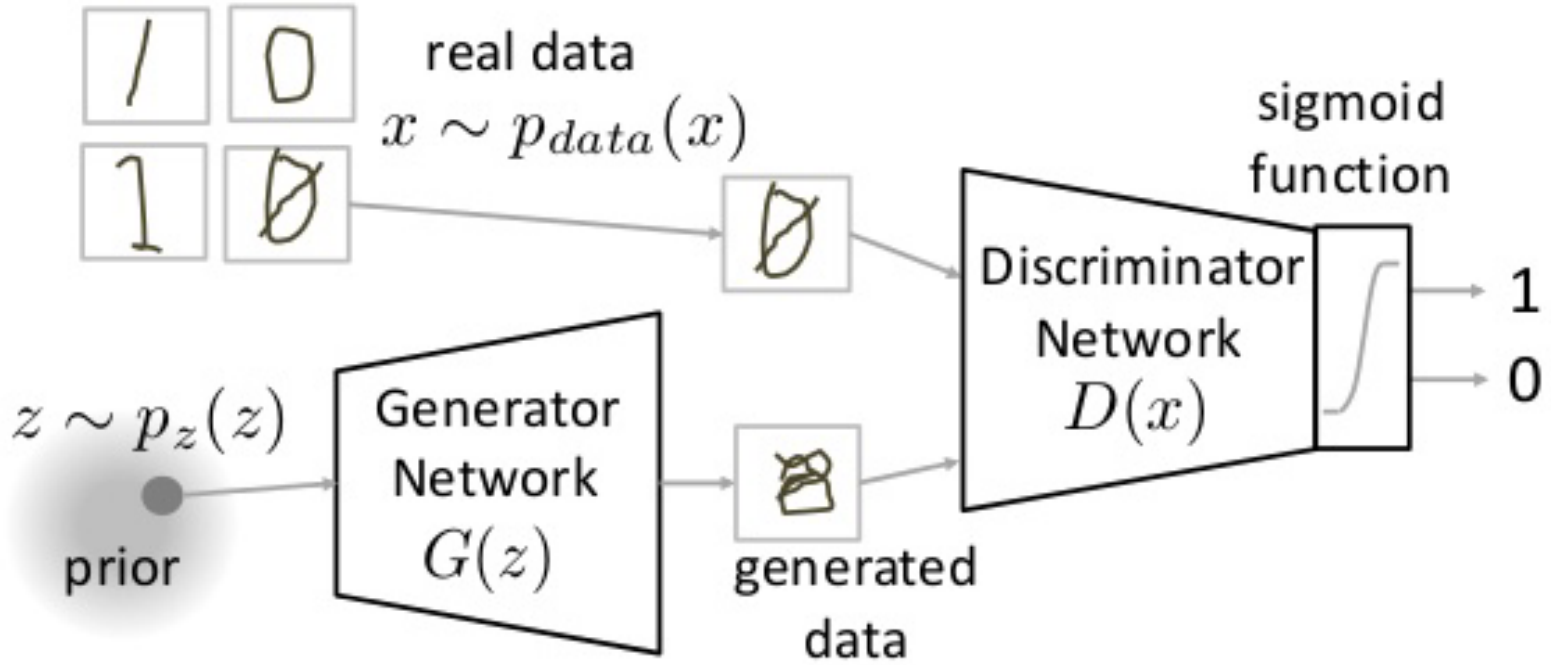


Goodfellow et al, 2014

Semi-supervised:Generative Adversarial Networks (GAN)

$$\min_G \max_D V(D,G)$$

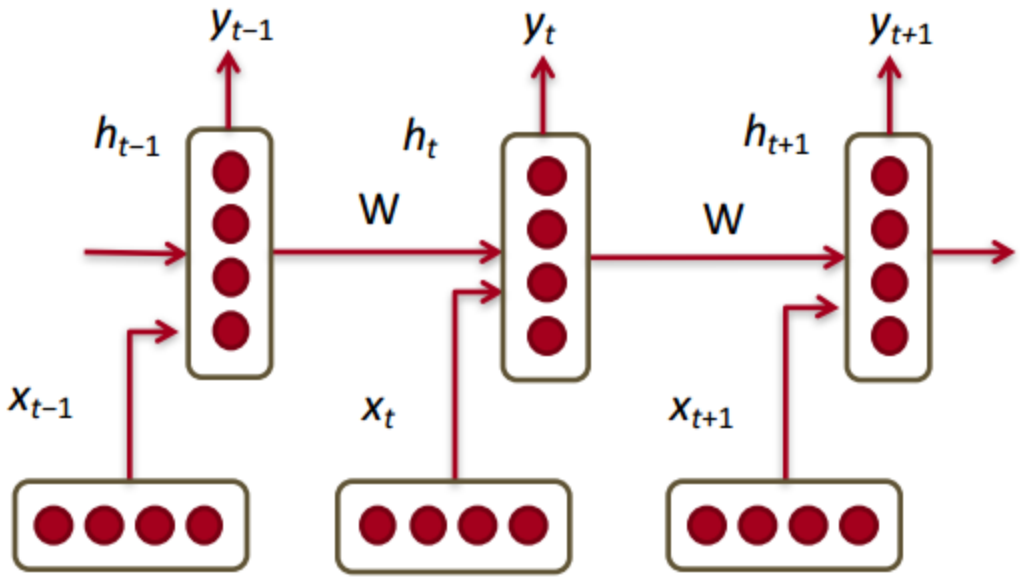
$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$



Slide from Mark Chang's GAN tutorial

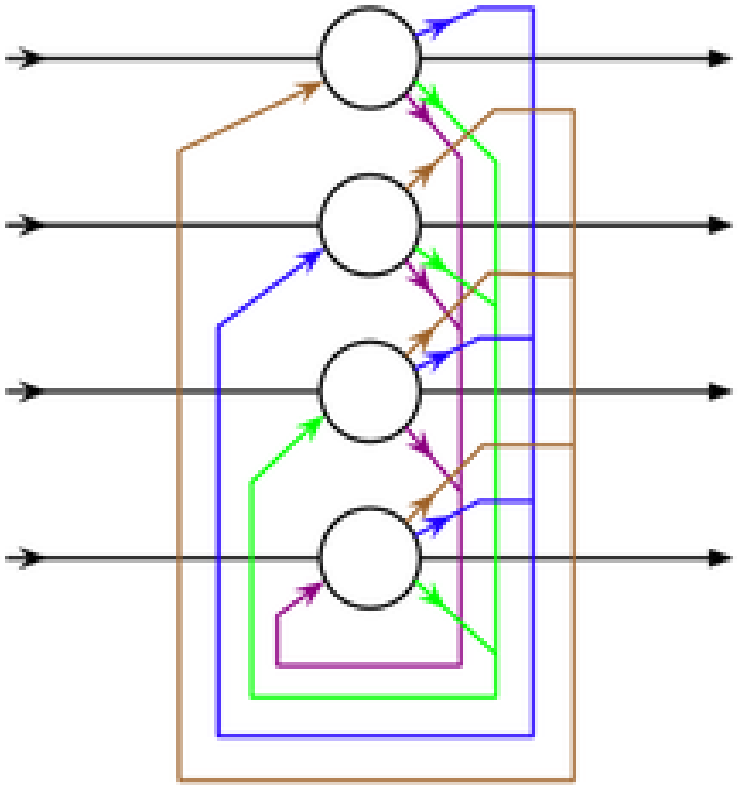
Recurrent Neural Network (RNN)

- Ties the weights at each time step
- Condition the neural network on all previous input
- RAM requirement only scales with number of input



RNN: Hopfield Network

- Earliest form of **Recurrent Neural Network** [devised by John Hopfield in 1982]



binary threshold units [-1,1]

RNN: Long Short-Term Memory Nets (LSTMs)

insensitivity to gap length

A advantage when there's unknown size bound between important events

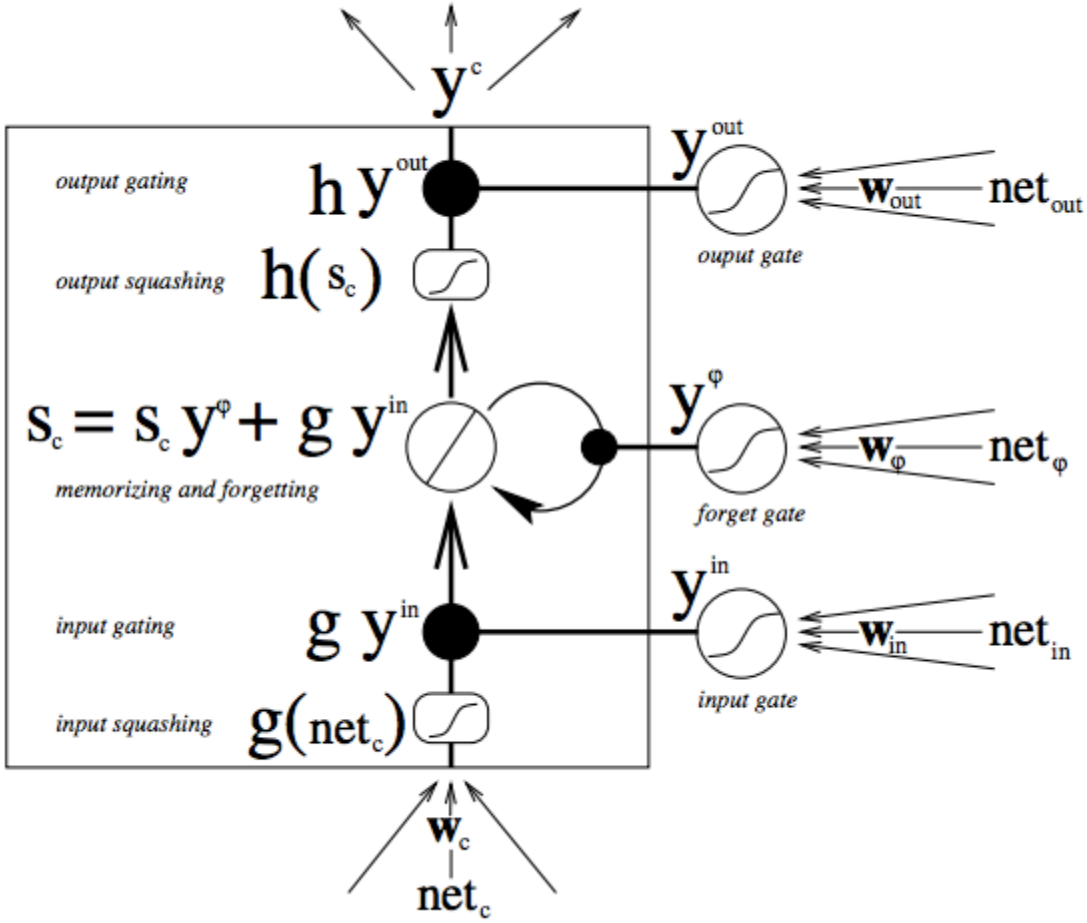


Figure from https://deeplearning4j.org/lstm.html

Random or Unsupervised Features

- Feature learning in CNN is very expensive
  - Every gradient step requires complete run of forward propagation and backward propagation
- Three ways to obtaining convolution kernels without supervised training.
  - Initialize them randomly
  - Design them by hand
  - Learn the kernels with an unsupervised criterion
    - *Apply k*-means clustering to small image patches, then use e ach learned centroid as a convolution kernel.
    - Greedy layer-wise pretraining (convolutional deep belief ne twork)

Gather More Data or Retune the Model?

- It is often much better to gather more data than to improve the learning algorithm. But data can be expensive.
- Measure the training set performance.
  - Poor training set performance: the learning algorithm is not using the training data properly.
    - Try increasing the size of the model - more layers or more hidden units
    - Try improving the learning algorithm - tune the hyperparameters
    - If the two does not work, quality of the training data may be poor.

Gather More Data or Retune the Model?

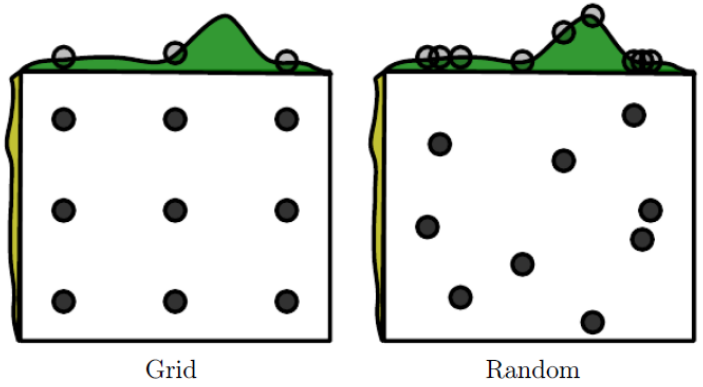
- Acceptable training set performance, then measure the performance of test set.
  - If test set performance is good enough – no more work to do.
  - If test set performance is bad (big gap between training and testing),
    - Gathering more data - most effective solutions.
    - Reduce the size of the model by adjusting hyperparameters, e.g., weight decay coefficients,
    - Adding regularization strategies such as dropout.

Selecting Hyperparameters

- Choosing hyperparameters manually
  - Requires understanding what the hyperparameters do and how machine learning models achieve good generalization
  - Requires understanding of how the data behaves
- Choosing hyperparameters automatically
  - Model-Based optimization
  - Computationally costly

Selecting Hyperparameters cont.

- Grid search
  - Search Evenly of the para. space
- Random Search
  - Finds good solutions faster than grid search
- Combination approach
  - Gird search then random search on selected range of values



Grid search vs random search - two hyperparameter case

Parameter Initialization

- Important to initialize all weights to small random values.
- Bias terms can be initialized to zero or to small positive values.

References

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning Book*. MIT Press.
- Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. 2014. On the Number of Linear Regions of Deep Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 1–9.
- Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Stegle Oliver. 2016. Deep Learning for Computational Biology. *Molecular Systems Biology*, 12: 878.
- Christopher M. Bishop. 2006. *Pattern Recognition And Machine Learning*. Springer.
- Li Deng and Navdeep Jaitly. 2015. *Deep Discriminative and Generative Models for Pattern Recognition*. Microsoft Research
- Juha Karhunen, Tapani Raiko, and Kyunghyun Cho. 2015. Unsupervised Deep Learning: A Short Review. *Advances in Independent Component Analysis and Learning Machines*: 125–142
- Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*: 1–8.
- https://deeplearning4j.org
- http://ufldl.stanford.edu/tutorial/

2. Interpretable Deep Learning Models

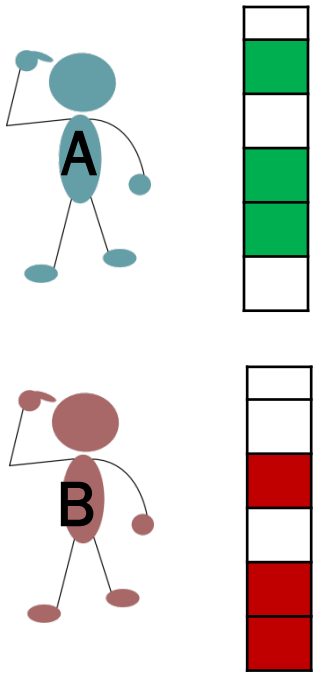
*Note: Many of the contents are extracted from a tutorial given in ICASSP 2017 by G. Montavon, W. Samek, K.-R. Müller

Why Interpretable?

- **Verify** that classifier works as expected
  - Esp. in areas where wrong decisions can be costly and dangero us (autonomous car, medical decision support systems, etc)
- **Improve classifier** by finding out the cause of low accur acy
  - Allows for human intervention
- **Learn** from the learning machine
  - Learn moves or rules we didn't know about
  - **Advance in science**
- **Compliance to legislation**
  - EU's "right to explanation"
  - Retain human decision in order to assign responsibility

Interpretable (Explainable) Method

- Because the doctors will not trust you unless you can **verify why** the model came to that conclusion.



Omics feature

Interpretable Machine Learning

Suggested treatment of patient A is 
because the model detected/learned  features in the patient.

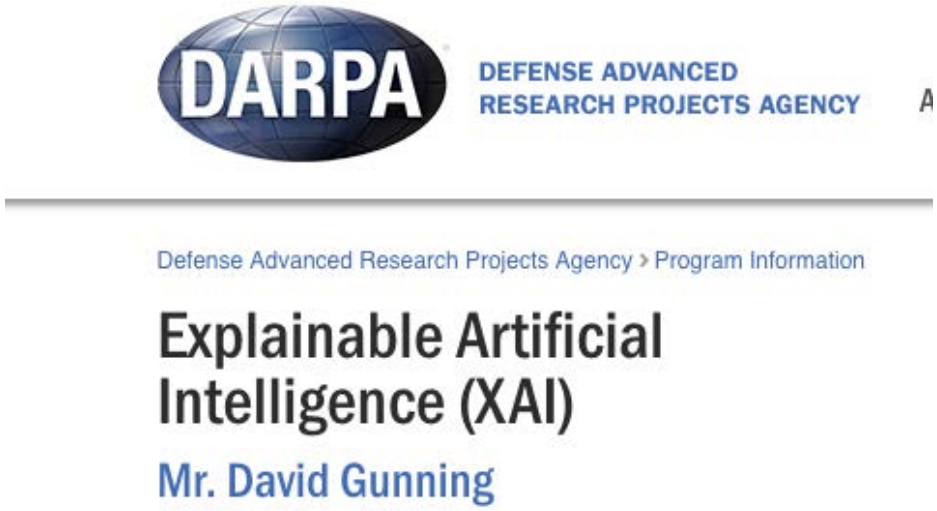
Suggested treatment of patient B is 
because the model detected/learned  features in the patient.

Q1: How can we explain the results?

Demands for Interpretable Methods

- EU's Right to Explanation

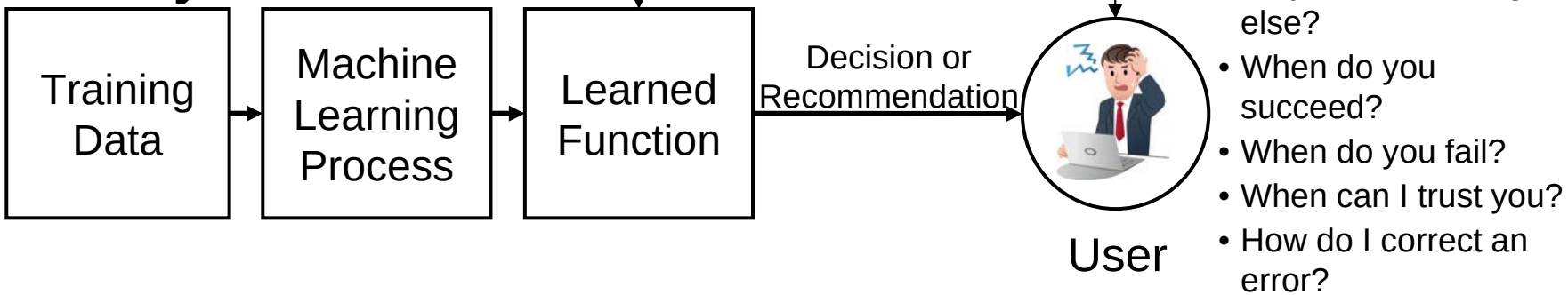
- DARPA's Explainable Artificial Intelligence (XAI) program 2017-2021



DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

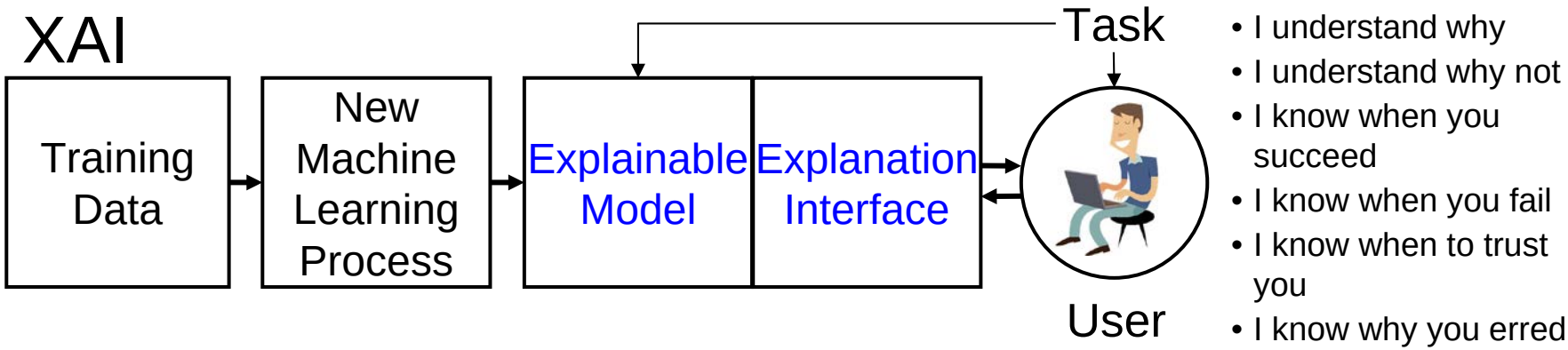
DARPA XAI Project

Today



- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How do I correct an error?

XAI



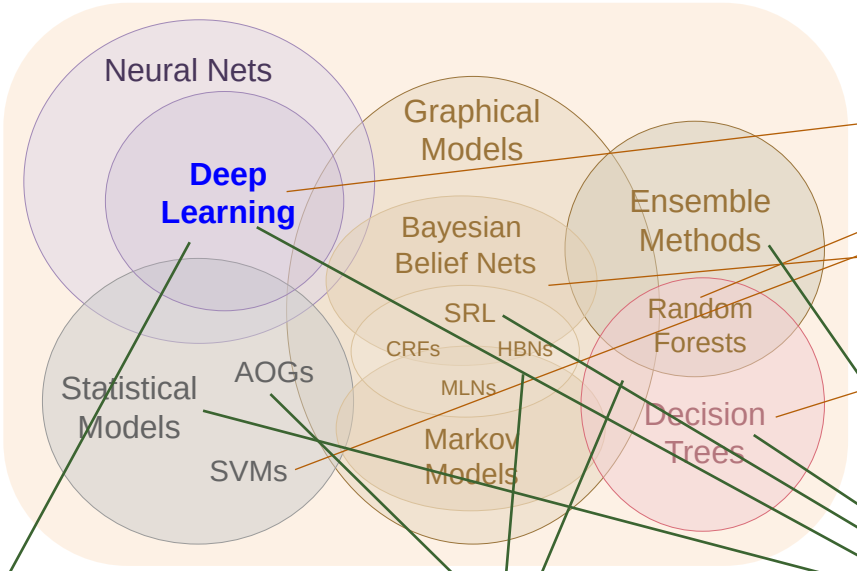
- I understand why
- I understand why not
- I know when you succeed
- I know when you fail
- I know when to trust you
- I know why you erred

Explainable Models

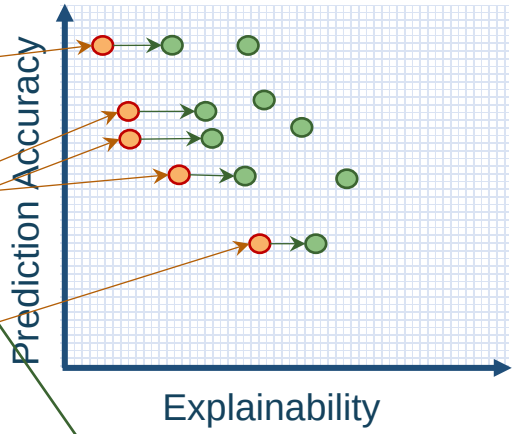
New Approach

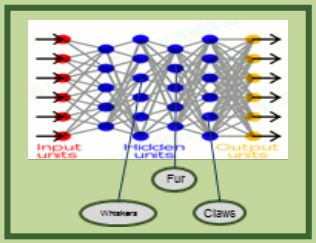
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



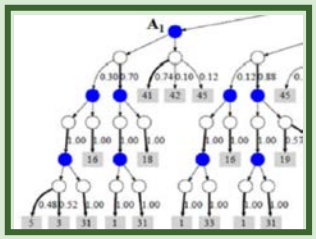
Explainability (notional)





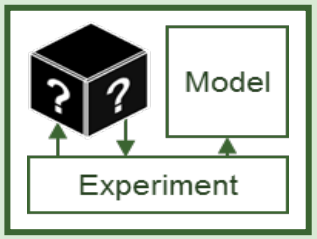
Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

Techniques to learn more structured, interpretable, causal models

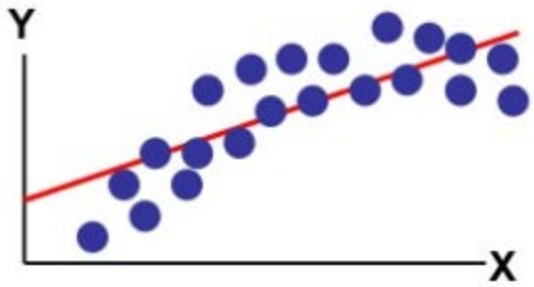


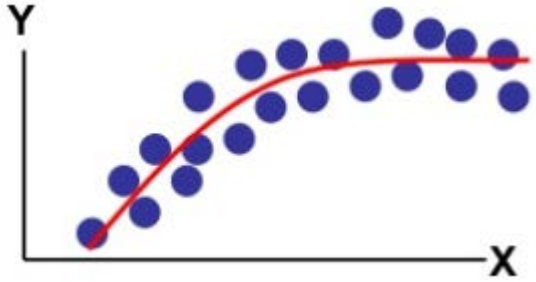
Model Induction

Techniques to infer an explainable model from any model as a black box

Interpretable vs Accurate Models

Linear models **vs** **Non-Linear models**





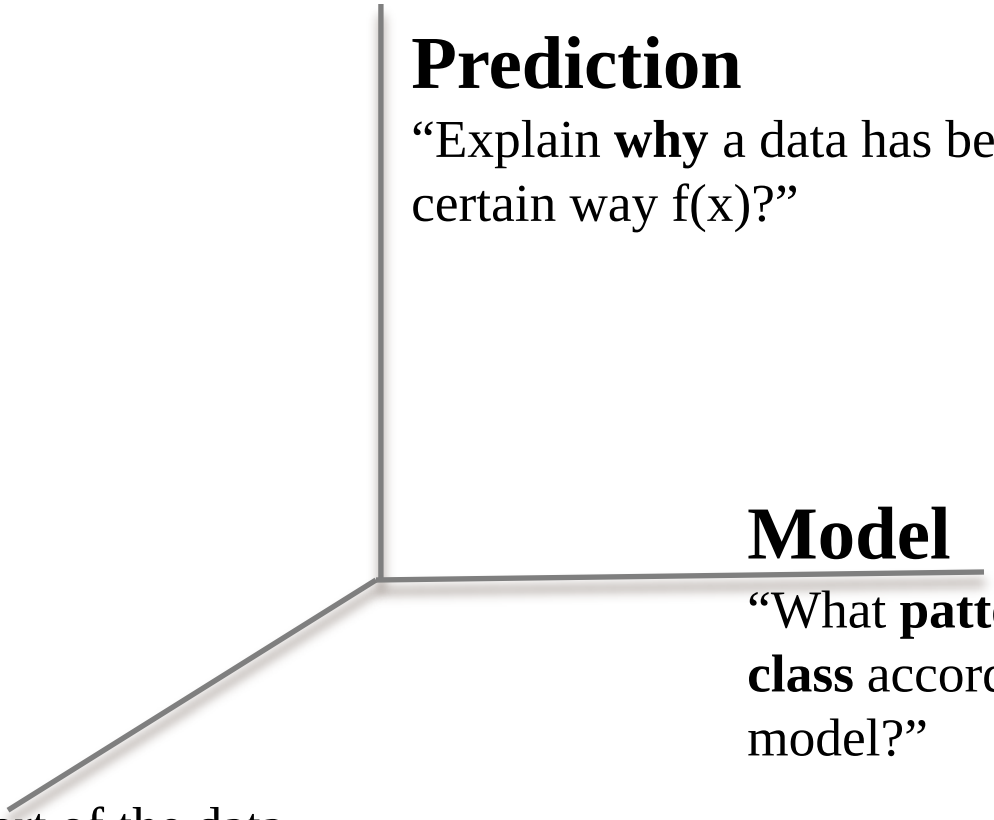
Traditionally Interpretability and Complexity of the model was thought to be anticorrelated.

Z. Lipton. 2016.

Measures of Interpretability

- Interpretability as a means to engender trust
  - Faith in a model's performance, robustness, or to some other property of the decisions it makes?
  - Low-level mechanistic understanding of our models?
- Uncovers causal structure in data
  - Uncover patterns in data as a whole
  - Uncover what part of the data is relevant to the result

Dimension of Interpretability



Prediction

"Explain **why** a data has been classify in a certain way f(x)?"

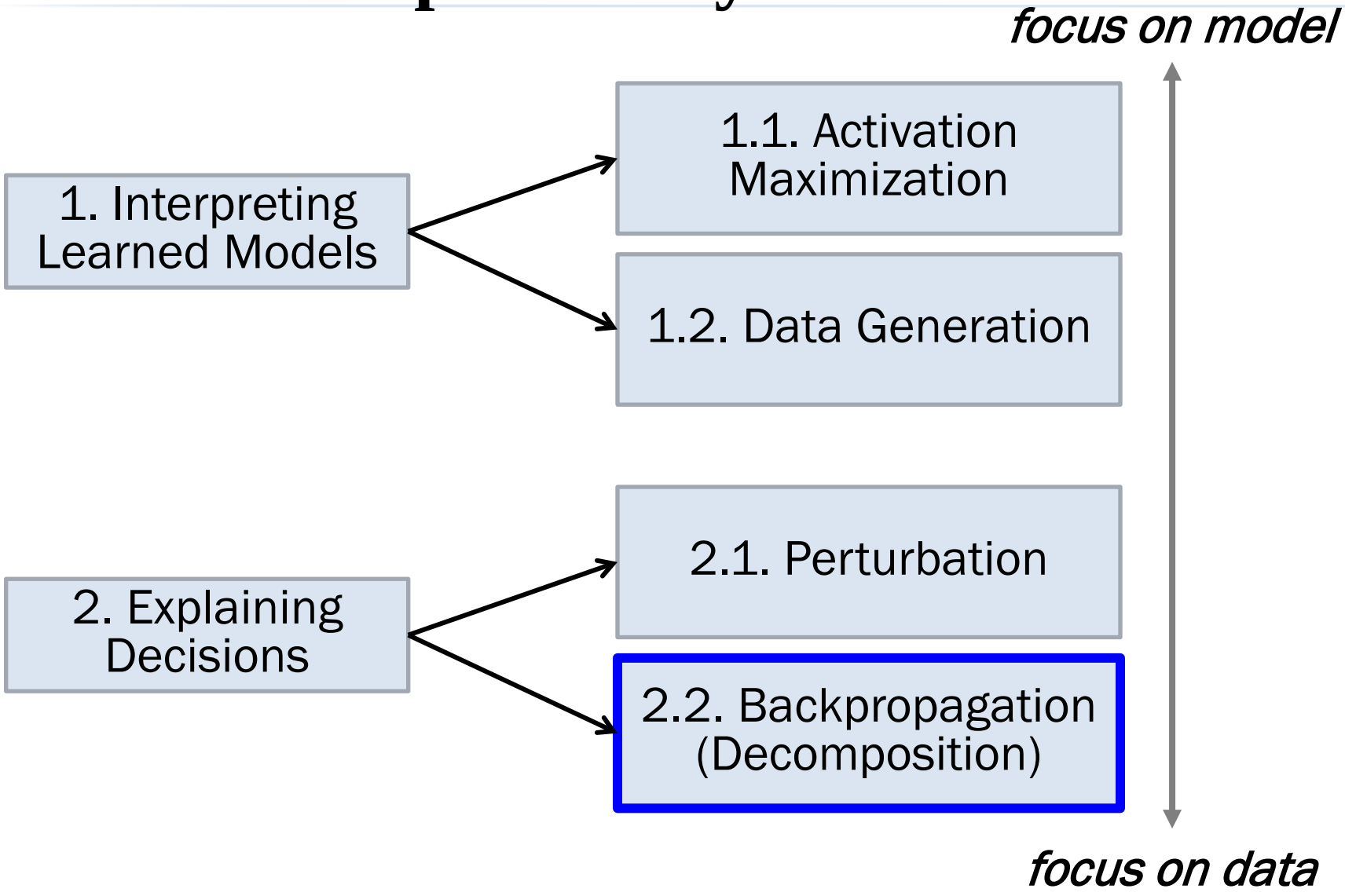
Model

"What **pattern** describes a **class** according to the model?"

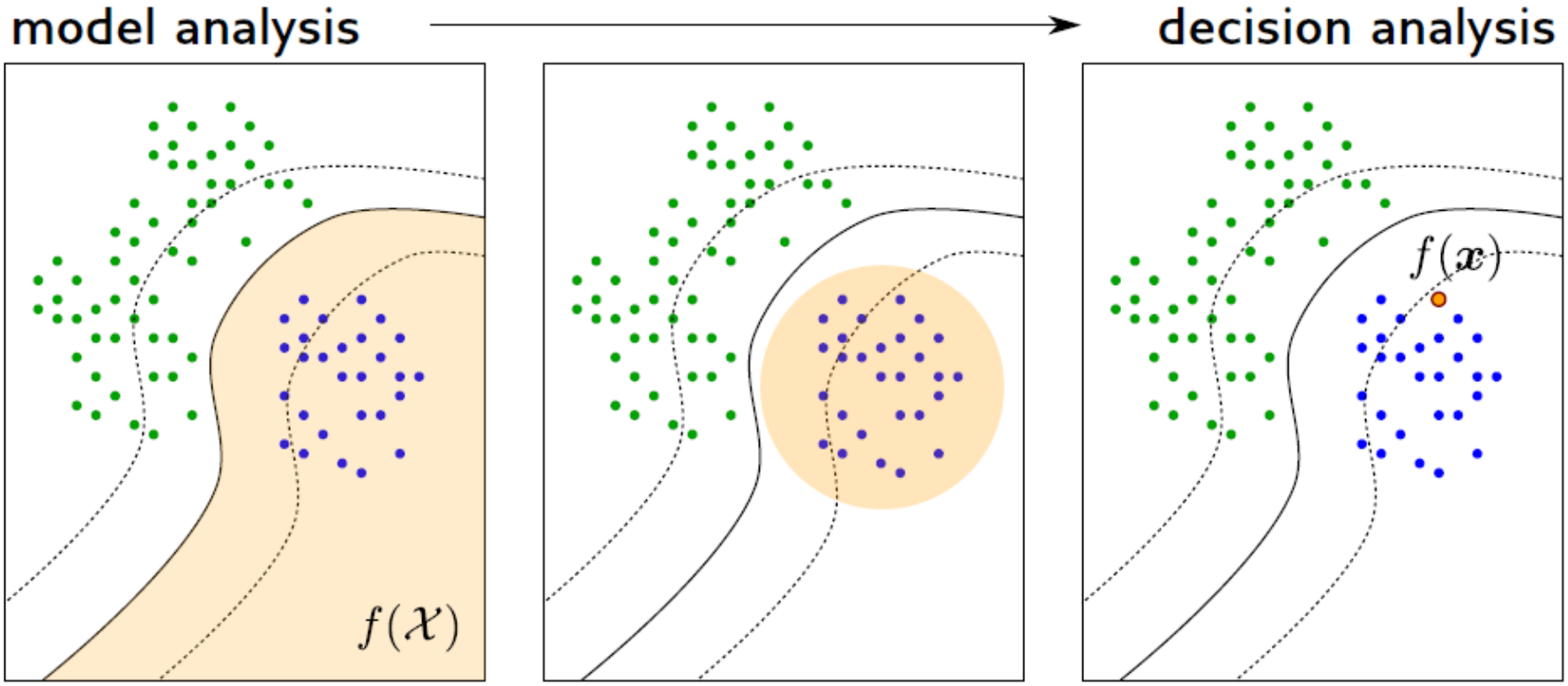
Data

"Which part of the data are **relevant** or **predictive** of the task?"

Goals of Interpretability



From Model Analysis to Decision Analysis



- Discriminative models

- Generative models

Interpreting Learned Models

Interpreting Classes and Outputs

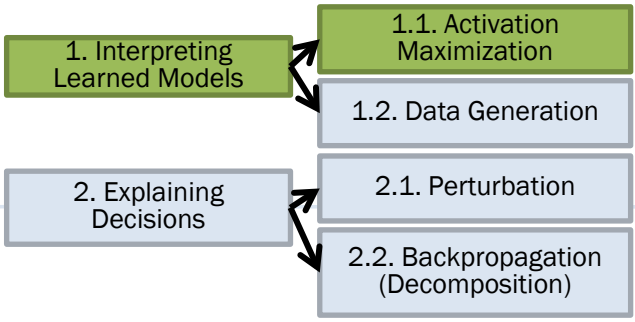
- **Activation Maximization**

  Q: What is the representative of class A?

  Q: What does high/low value of output neuron B mean?

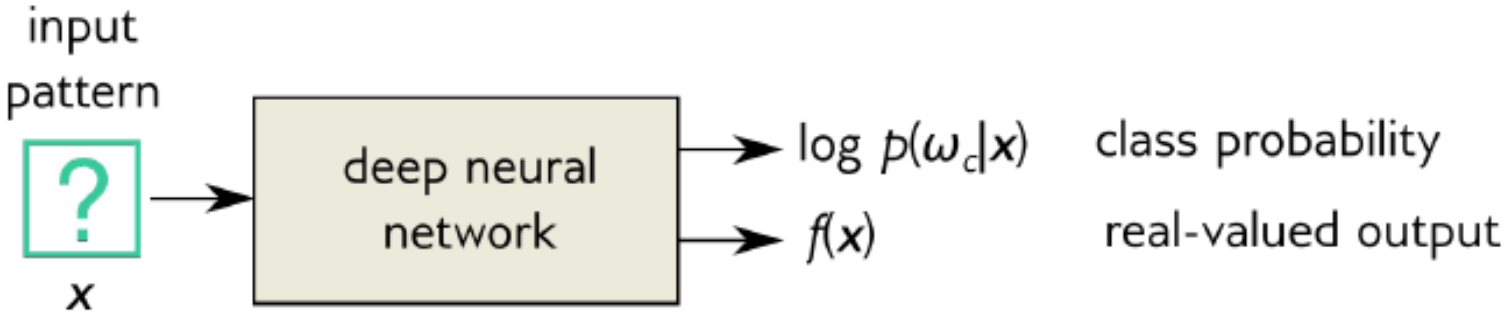
- **Data Generation**

  Q: What did each of its neurons learn?



1.1. Activation Maximization

Interpreting concepts predicted by a deep neural net via activation maximization



- Example :
  - Creating class prototype: $argmax_{x \in \chi} \log p(w_c|\boldsymbol{x})$
  - Synthesizing extreme case: $argmax_{x \in \chi} f(\boldsymbol{x})$

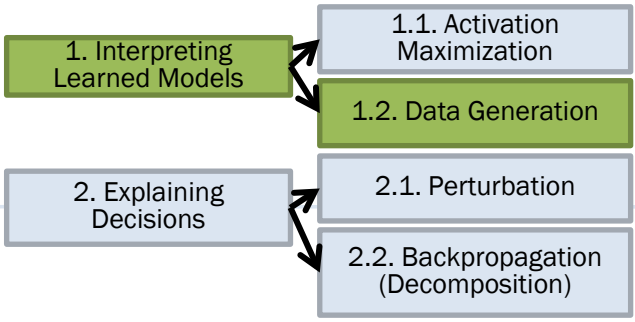
Improving Activation Maximization

- **Idea**: Force the features learned to match the data more closely.
- Now the optimization problem become

Finding the input pattern that maximizes **class probability.**

Find the **most likely input pattern** for a given class.

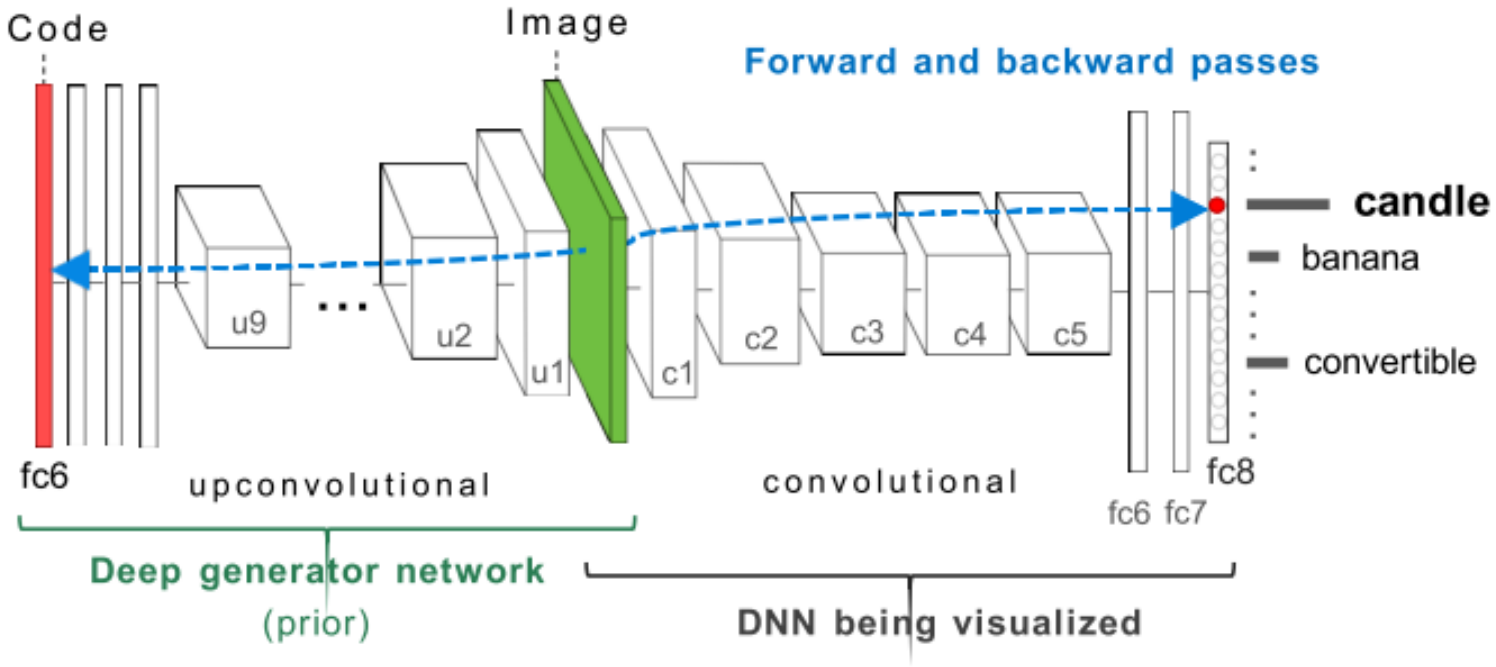
[Nguyen et al. 2016]:



1.2. Data Generation

Problem: Activation maximization problem as finding a code $\boldsymbol{y}^l$ such that:

$$\widehat{\boldsymbol{y}^l} = \arg\max_{\boldsymbol{y}^l} \Phi_h\left(G_l\left(\boldsymbol{y}^l\right)\right) - \lambda\left\|\boldsymbol{y}^l\right\|$$

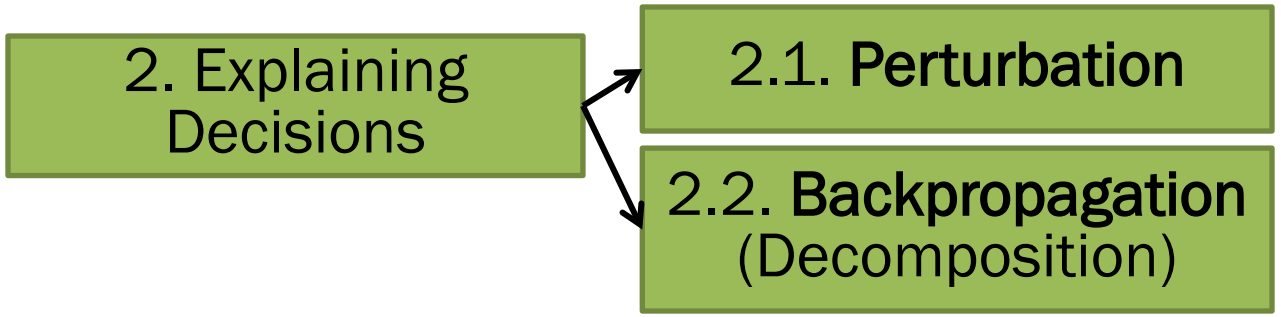


Deep generator network proposed by Nguyen et al. 2016

2. Explaining Decisions

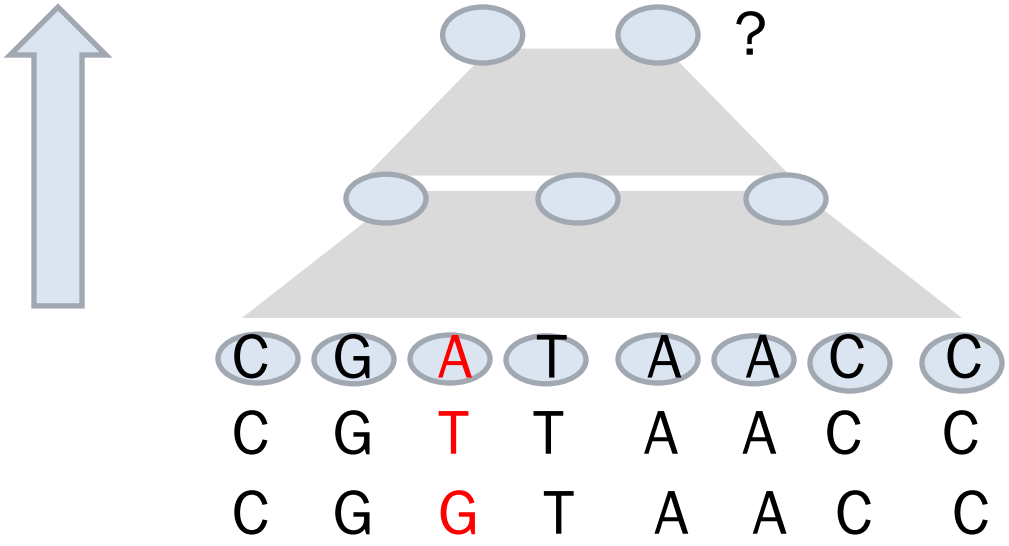
- **Goal:** Determine the relevance of each input feature for a given decision, by assigning to these variables **relevance scores** to each feature.
- Important for **Personalized Healthcare**

- Two approaches:

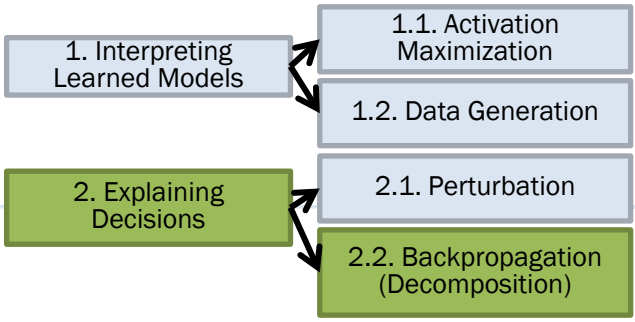


Perturbation Approaches

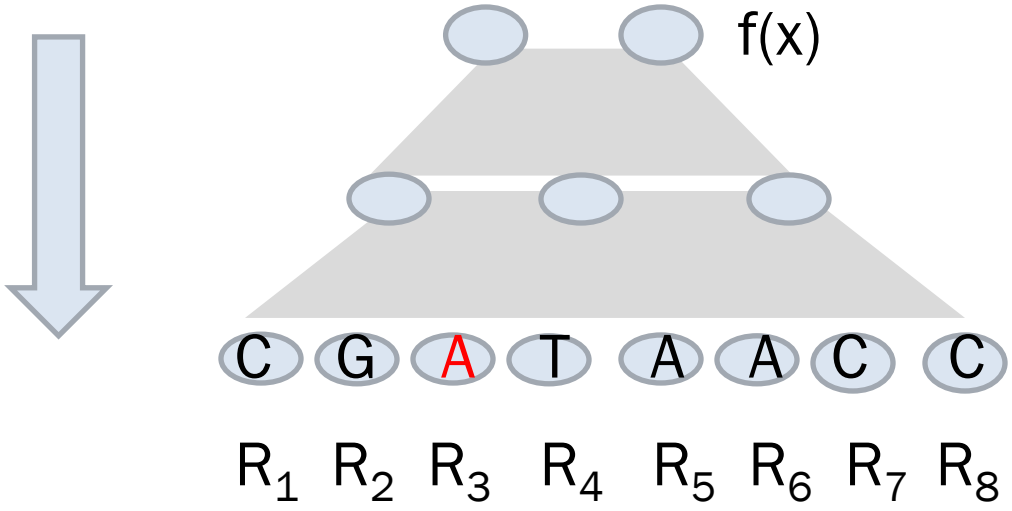
- Make perturbation to input and observe the difference in the output
- ☹ Every time you make a perturbation output needs to be recomputed



Backpropagation methods



- Sensitivity analysis
- Layer-wise relevance propagation (Deep Tylor)
- DeepLIFT



Explaining by Sensitivity Analysis

Given prediction function $f(x_1, x_2, \ldots, x_d)$ on d dimensional input data $\boldsymbol{x} = \quad x_1, x_2, \ldots, x_d\ )$,

Sensitivity analysis is the measure of local variation of the prediction function f along each input dimension

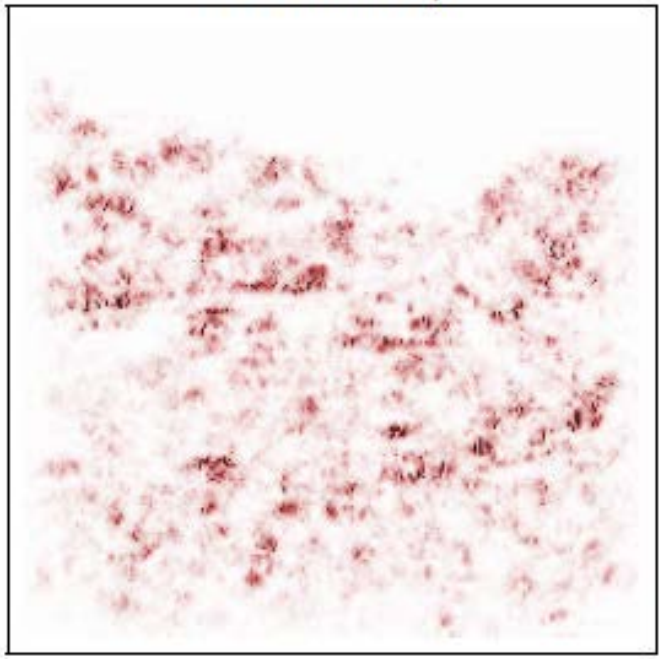
$$R_i = \left(\frac{\partial f}{\partial x_i}\Big|_{x=x}\right)^2$$

- Easy to implement
  - Requires access to the gradient of the decision function
  - May not explain the prediction well

Sensitivity Analysis

input image

sensitivity



Explaining by Decomposing

Decomposition methods decompose prediction value f(x) to **relevance scores** R_i such that

$$\sum_i R_i = f(x_1, \ldots, x_d)$$

Decomposition **explains the function value** itself.

Sensitivity Analysis in Decomposition View

- Decomposition: $\sum_i \ R_i = f(x_1, \ldots, x_d)$

- Sensitivity Analysis:

$$R_i = \left(\frac{\partial f}{\partial x_i}\Big|_{x=x}\right)^2$$

$$\sum_i \ R_i = \|\nabla_x f\|^2$$

  - Sensitivity analysis **explains a variation** of the function.

Decomposition on Shallow Nets

- Taylor decomposition of function $f(x_1, \ldots, x_d)$

$$f(\boldsymbol{x}) = \underbrace{f(\tilde{\boldsymbol{x}})}_{0} + \sum_{i=1}^{d} \underbrace{\frac{\partial f}{\partial x_i}\Big|_{x=\tilde{x}} (x_i - \tilde{x}_i)}_{R_i} + \underbrace{O(\boldsymbol{x}\boldsymbol{x}^\top)}_{0}$$

- Can it be applied on Deep Learning?
  - Doesn't work well on DNN
  - Also subjected to gradient noise

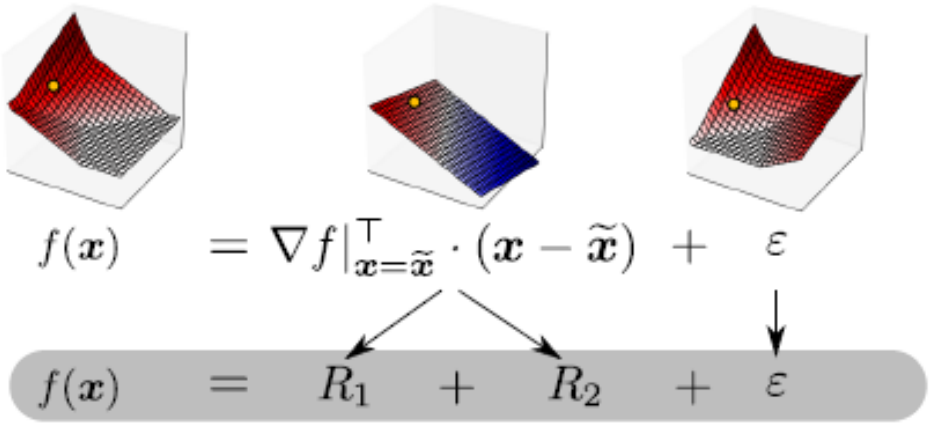
[ICASSP 2017 Tutorial]

[Montavon et al. 2017]

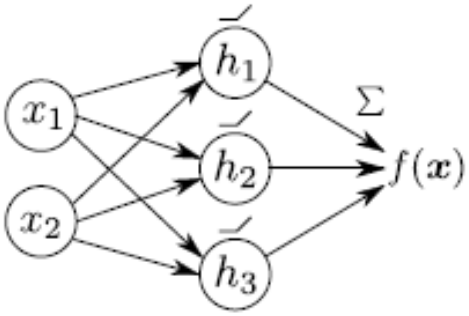
Deep Taylor Decomposition

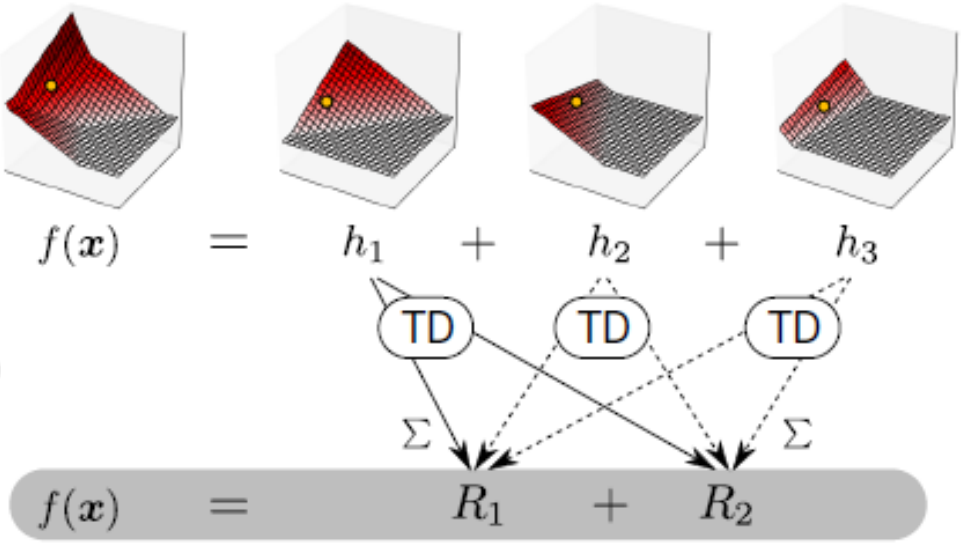
Taylor decomposition (TD)

$f(\boldsymbol{x}), \nabla f, \ldots$

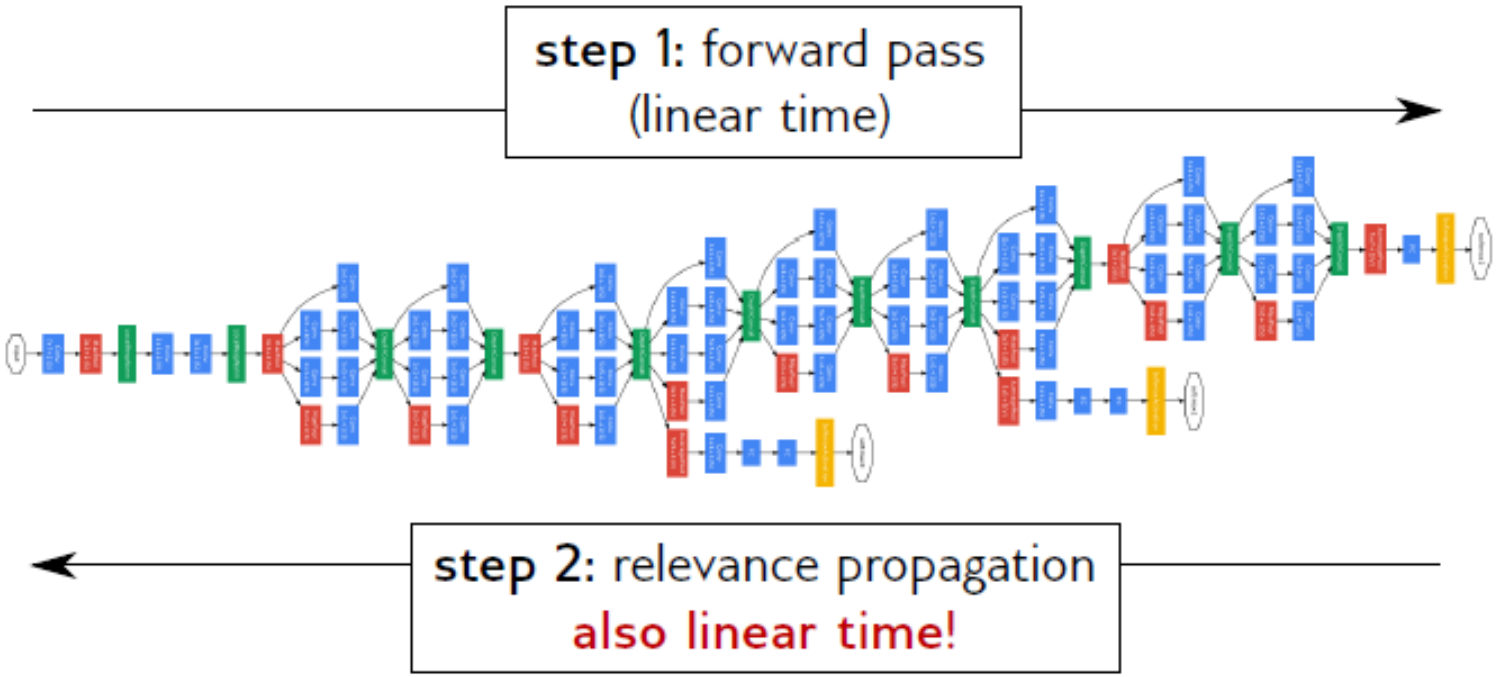


deep Taylor decomposition (DTD)





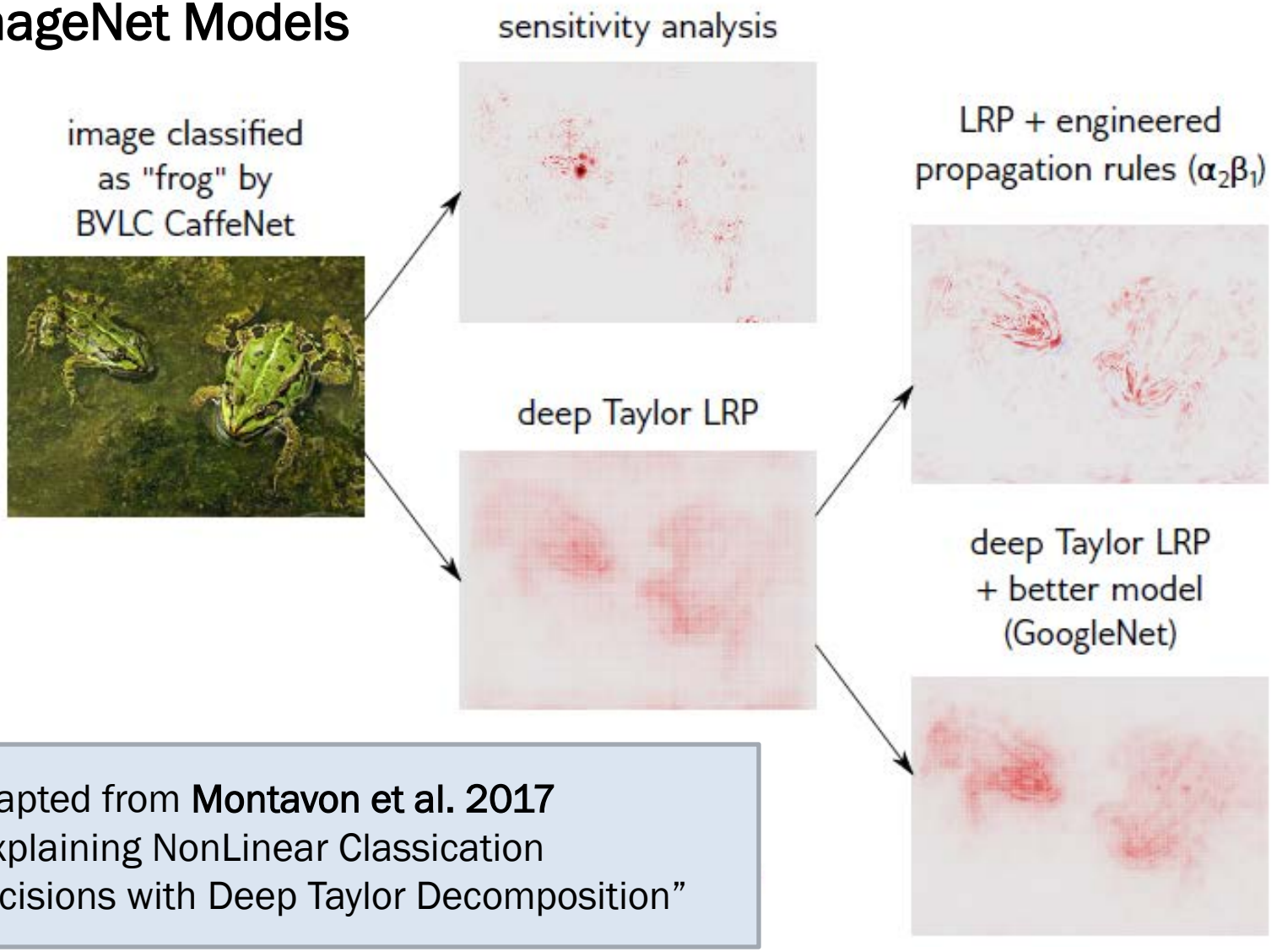
Layer-Wise Relevance Propagation (LRP)



Propagation rule:

$$R_i = \sum_i q_{ij} R_j \qquad \sum_i q_{ij} = 1$$

ImageNet Models



Adapted from **Montavon et al. 2017**
"Explaining NonLinear Classication
Decisions with Deep Taylor Decomposition"

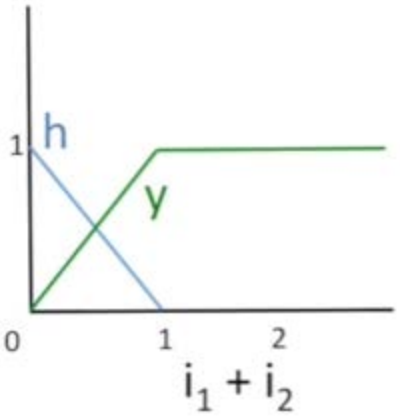
Shrikumar et al 2017 CVPR

DeepLIFT

- DeepLIFT explains the difference in output from some ‘reference’ output in terms of the difference of the input from some ‘reference’ input.
- The ‘reference’ input represents some default or ‘neutral’ input that is chosen according to what is appropriate for the problem at hand
- **Activation difference** propagated down to input
- Capable to propagate relevance down even when the gradient is zero. (solves saturation problem)

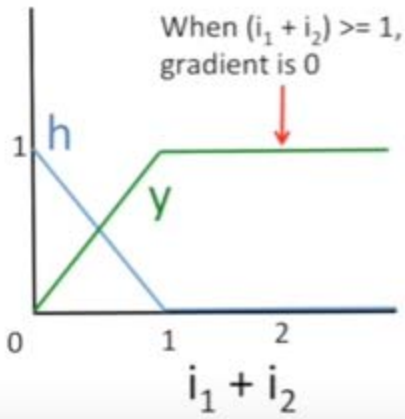
Saturation problem illustrated

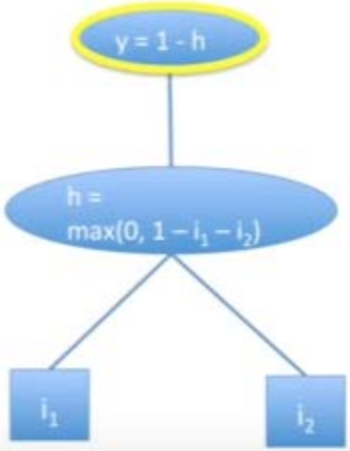
$y = (i_1 + i_2)$ when $(i_1 + i_2) < 1$
$= 1$ when $(i_1 + i_2) >= 1$



Saturation revisited

$y = (i_1 + i_2)$ when $(i_1 + i_2) < 1$
$= 1$ when $(i_1 + i_2) >= 1$





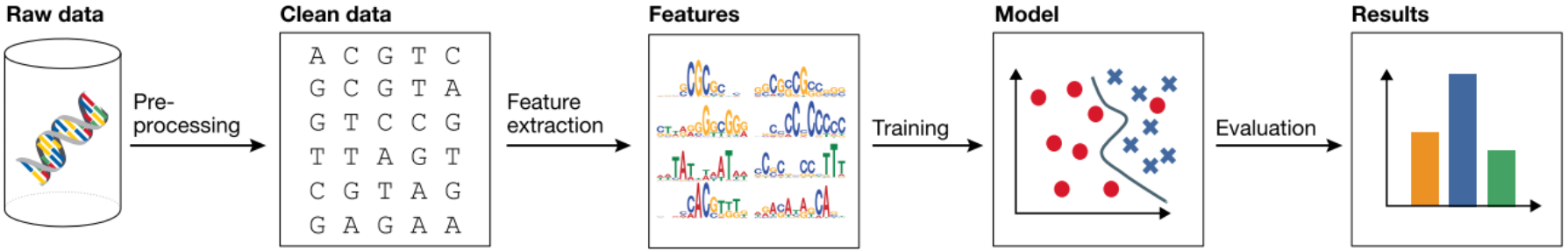
Reference

- W. Samek, G. Montavon & K.-R. Müller “Tutorial on Methods for Interpreting and Understanding Deep Neural Networks.” ICASSP 2017 Tutorial.
- David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. volume 11, pages 1803–1831, 2010.
- Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus Robert Muller. 2016. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*: 1–13.
- Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus Robert Müller. 2017. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition* 65, August 2016: 211–222.
- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. volume 10, page e0130140, 2015.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop at International Conference on Learning Representations*, 1–8.
- Korattikara A, Rathod V, Murphy K, Welling M. Bayesian Dark Knowledge. arXiv preprint arXiv:150604416. 2015;.
- Zachary C Lipton. 2016. The Mythos of Model Interpretability. *ICML Workshop on Human Interpretability in Machine Learning*.
- D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, Jun 2009.
- M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901v3, 2013.
- Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In Proc. ICML, 2012.
- Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *29th Conference on Neural Information Processing Systems (NIPS 2016)*, 1–29.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *CVPR*.

3. DL Applications in Bio-Healthcare

Benefits of DNN Learning Revisited

Classical Machine Learning Pipeline



Deep Learning Pipeline

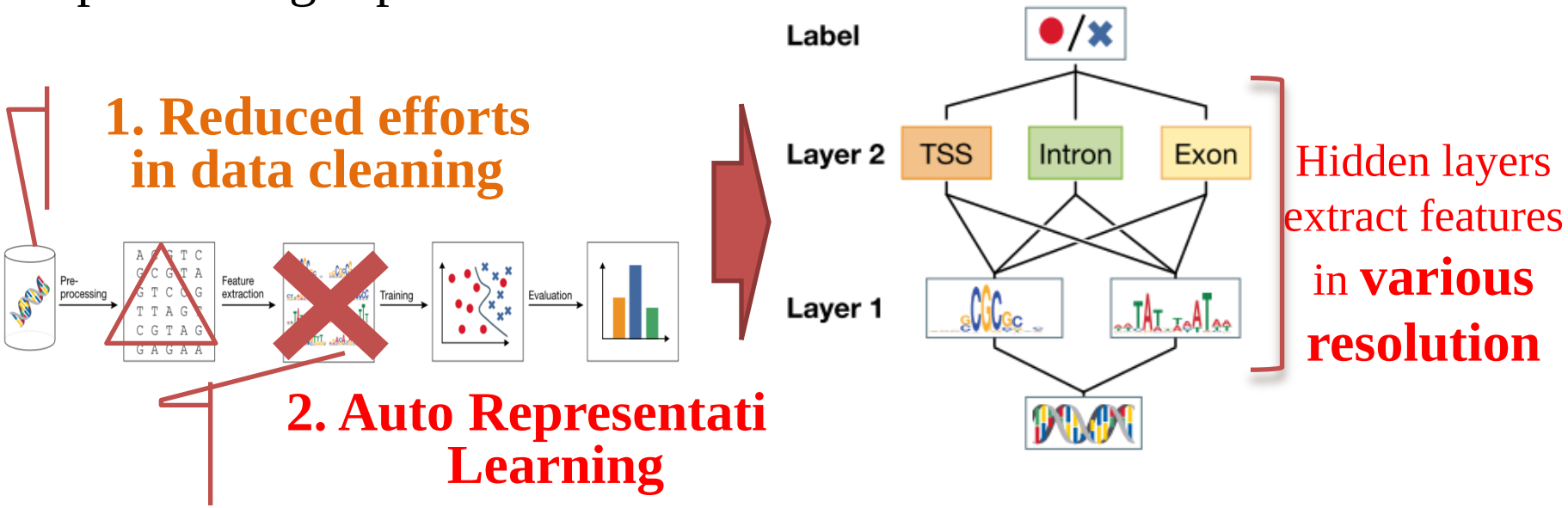


Fig 1A,D from Angermueller et al. (2016) *Molecular Systems Biology*, (12), 878.

Various DNN Applications

- Genomics Applications
  - Regulatory Genomics
  - Protein Structure Prediction
  - Applications on High throughput Data
- Healthcare Applications
  - ICU data analysis
  - EHR data analysis
  - Computational Drug Development

Early works of DNN in Alternative Splicing

Deep Feedforward NN

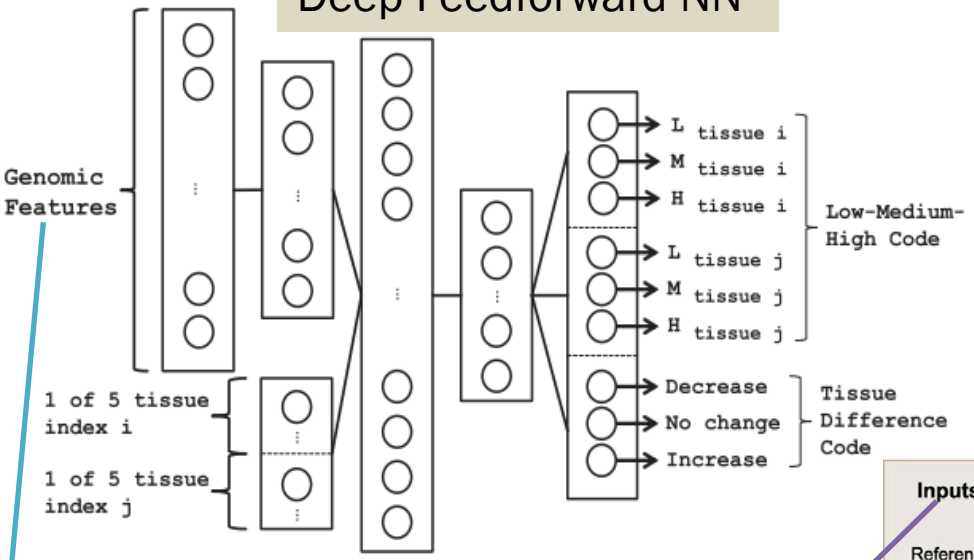


Fig 1 of Leung et al. (2014) Bioinformatics 30(12) 121-129

"Deep learning of the tissue-regulated splicing code"

1393 features extracted from each exon of 5 different tissue types

1000 predetermined features from candidate exon and adjacent introns

Early works still utilize **selected (large size) features**

Fully connected Feedforward NN (Bayesian Deep Learning)

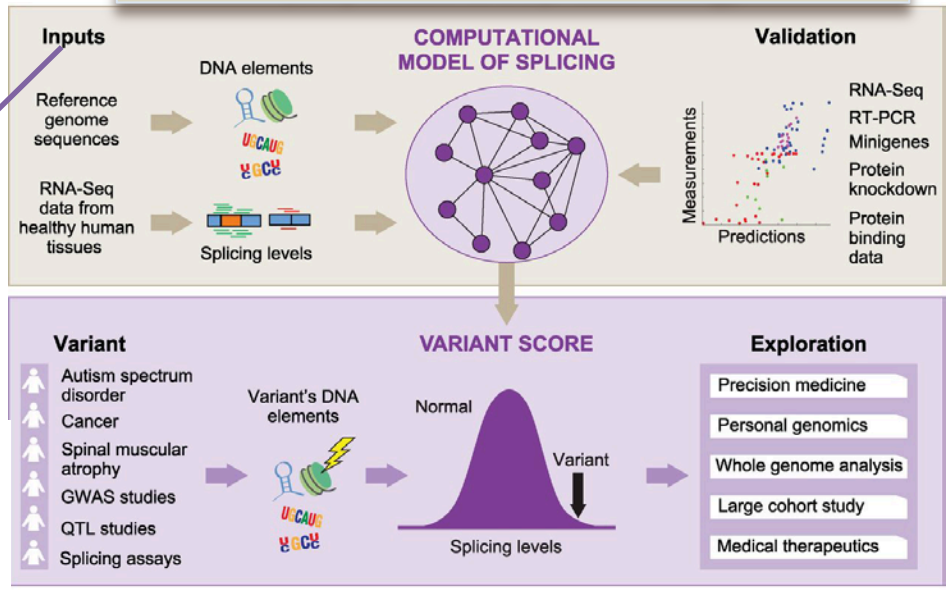


Fig 1 of Xiong et al. (2015) Science 347(6218):1254806

Feature listing

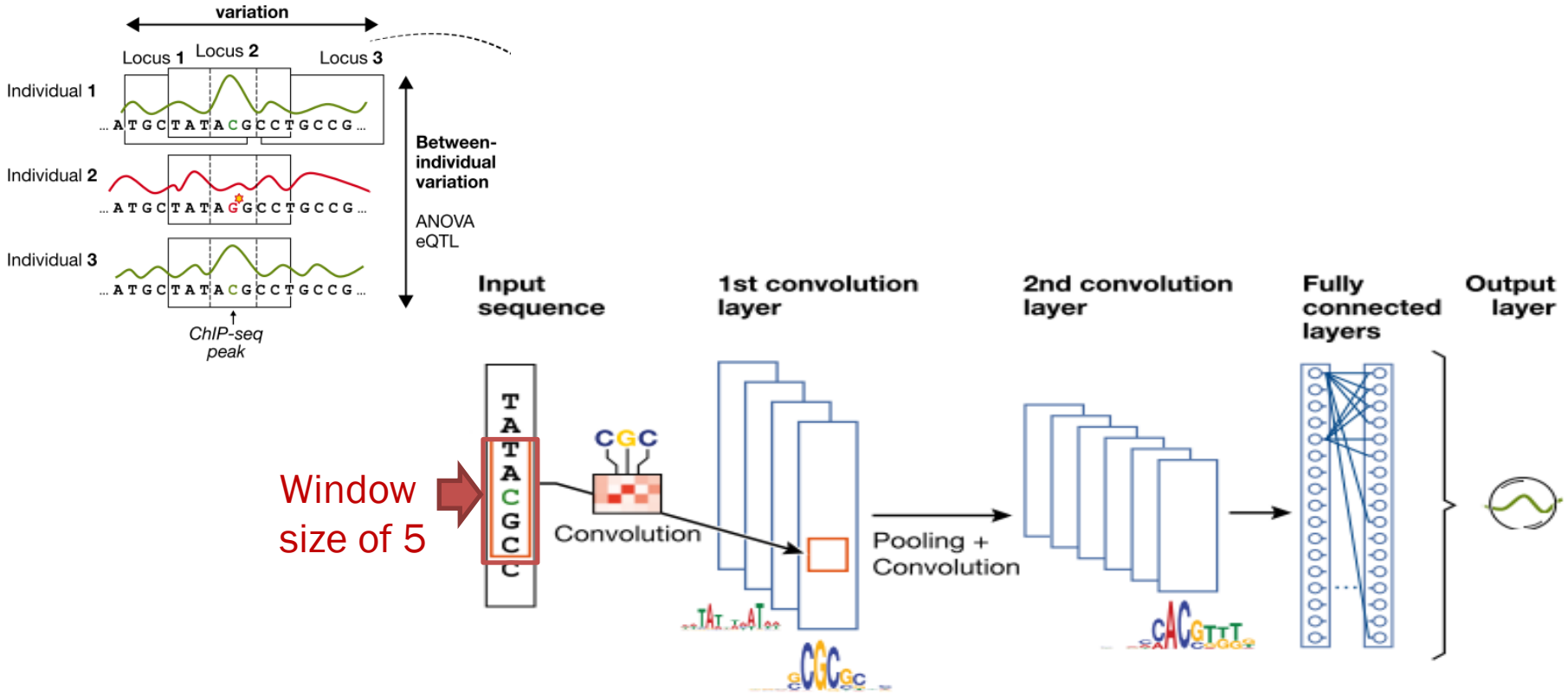
Leung et al. (2014)
Bioinformatics 30(12)
121-129

Group #	Name	Description	Type	# of Features
01	short-seq-1mer	Frequency of nucleotide patterns of different lengths (1 to 3).	real (0-1)	28
02	short-seq-2mer			112
03	short-seq-3mer			320
04	translatable-C1	Describes whether exons can be translated without a stop codon in one of three possible reading frames. For example, C1A means the exons of interest are C1 + A.	binary	1
05	translatable-C1A			1
06	translatable-C1AC2			1
07	translatable-C1C2			1
08	mean-con-score-AI2	Mean conservation score.	real (0-1)	1
09	mean-con-score-I1A			1
10	mean-con-score-I2C2			1
11	mean-con-score-C1I1			1
12	log-length	Log base 10 lengths of exons.	real	5
13	log-length-ratio	Log base 10 length ratios of exons.	real	3
14	acceptor-site-strength	Strength of acceptor and donor sites.	real	2
15	donor-site-strength			2
16	frameshift-exonA	Whether exon A introduces frame shift.	binary	1
17	rna-sec-struct	RNA secondary structures.	real (0-1)	32
18	5mer-motif-down	Counts of motif clusters of different lengths (5 to 7) weighted by conservation upstream and downstream from alternative exon.	real	54
19	6mer-motif-down			76
20	7mer-motif-down			28
21	5mer-motif-up			49
22	6mer-motif-up			78
23	7mer-motif-up			29
24	ese-ess-A	Counts of exonic splicing enhancers and silencers.	real	4
25	ese-ess-C1			4
26	ese-ess-C2			4
27	pssm-SC35	PSSM scores of SC35 splicing regulator protein.	real	5
28	pssm-ASF-SF2	PSSM scores of ASF/SF2 splicing regulator protein.		5
29	pssm-SRp40	PSSM scores of SRp40 splicing regulator protein.		10
30	nucleosome-position	Nucleosome positioning.	real	4
31	PTB	Phosphotyrosine-binding domain.	real	50
32	Nova-counts	Counts of Nova motif.	integer	27
33	Nova-cluster	Nova cluster score.	real	8
34	T-rich	Counts of motif with and without weighting by conservation.	real	24
35	G-rich			8
36	UG-rich			16
37	GU-rich			32
38	Fox	Counts of motif with and without weighting by conservation.	real	24
39	Quak			8
40	SC35			22
41	SRm160			11
42	SRrp20/30/38/40/55/75			77
43	CELF-like			2
44	CUGBP			16
45	MBNL			24
46	TRA2-alpha			22
47	TRA2-beta			22
48	hnRNP-A			44
49	hnRNP-H			22
50	hnRNP-G			22
51	9G8			22
52	ASF/SF2			11
53	Sugnet			2
54	alt-AG-pos	Position of the alternative AG and GT position.	integer	2
55	Alu-AI2	Counts of ALU repeats.	integer	12

C1 and *C2* denote the flanking constitutive exons ; *A* denotes the alternative exon ; *I1* denotes the intron between *C1* and *A* ; *I2* denotes the intron between *A* and *C2*

DNA/RNA Sequence Analysis with Deep CNN

Convolution step in Deep CNN resembles traditional sequence "**windowing**" scheme

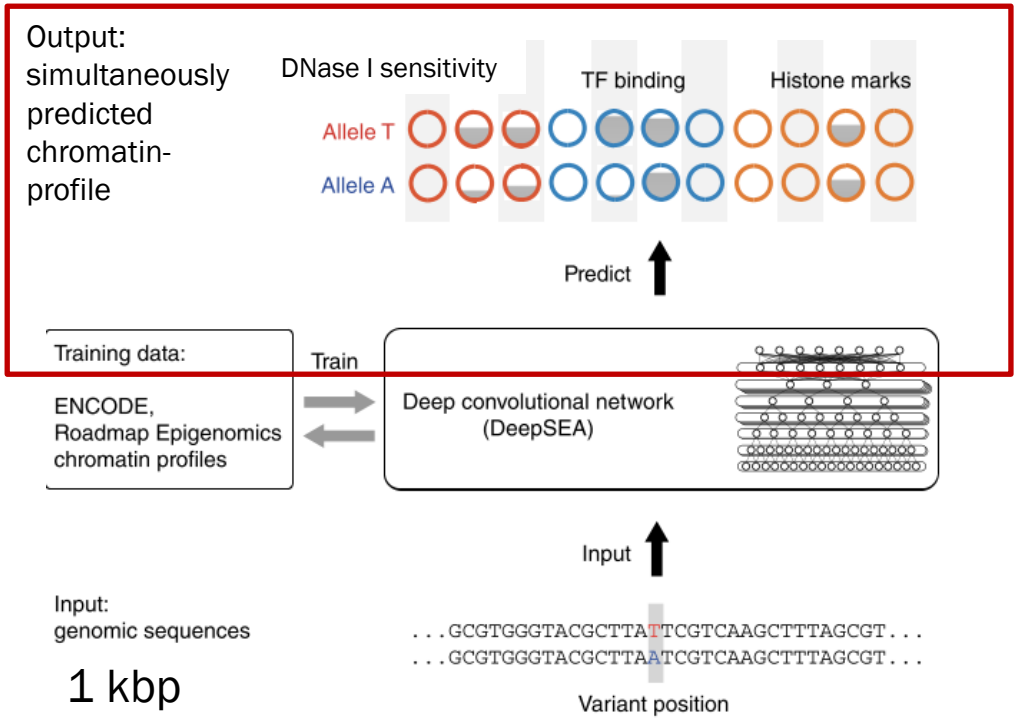


Angermueller et al. (2016) *Molecular Systems Biology*, (12), 878.

DeepSEA: CNN-based noncoding variant effect prediction

GOAL: Identifying functional effects of noncoding variants

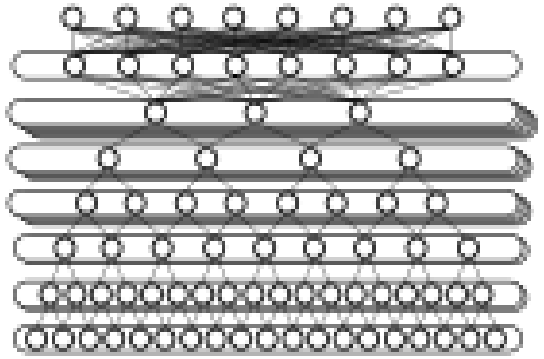
DeepSEA CNN structure



Innovative points:

1. Use long seq. 1kbp
2. multitask architecture

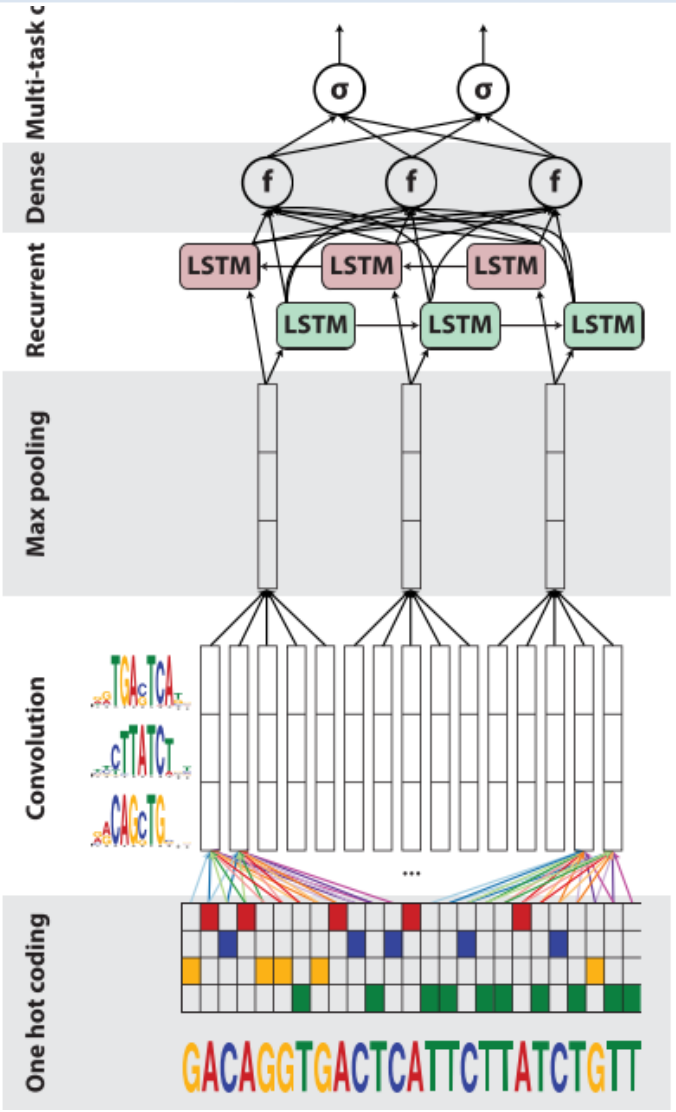
-> multiple output variables
919 chromatin features (125 DNase features, 690 TF features, 104 histone features)



Zhou, J., & Troyanskaya, O. G. (2015). *Nature Methods*, *12*(10), 931–4.

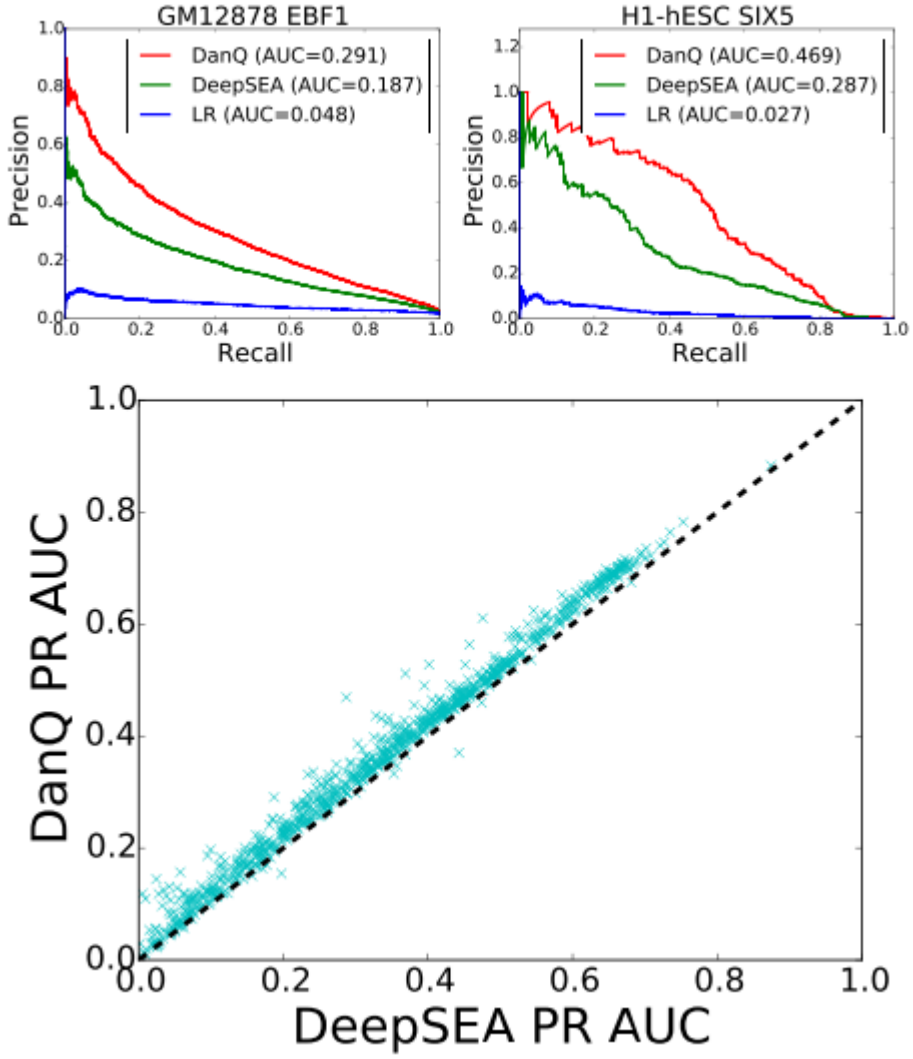
DanQ: Quantifying the Function of DNA

- Motivation: Over 98% of the human genome is non-coding and 93% of disease-associated variants lie in non-coding regions.
- Proposed: DanQ, hybrid convolutional and bi-directional long short-term memory recurrent neural network predicting non-coding function.
- Data:
  - Input: GRCh37 reference genome segmented into non-overlapping 200-bp bins.
  - Labels: Intersecting 919 ChIP-seq and DNase-seq peak sets from uniformly processed ENCODE and Roadmap Epigenomics data

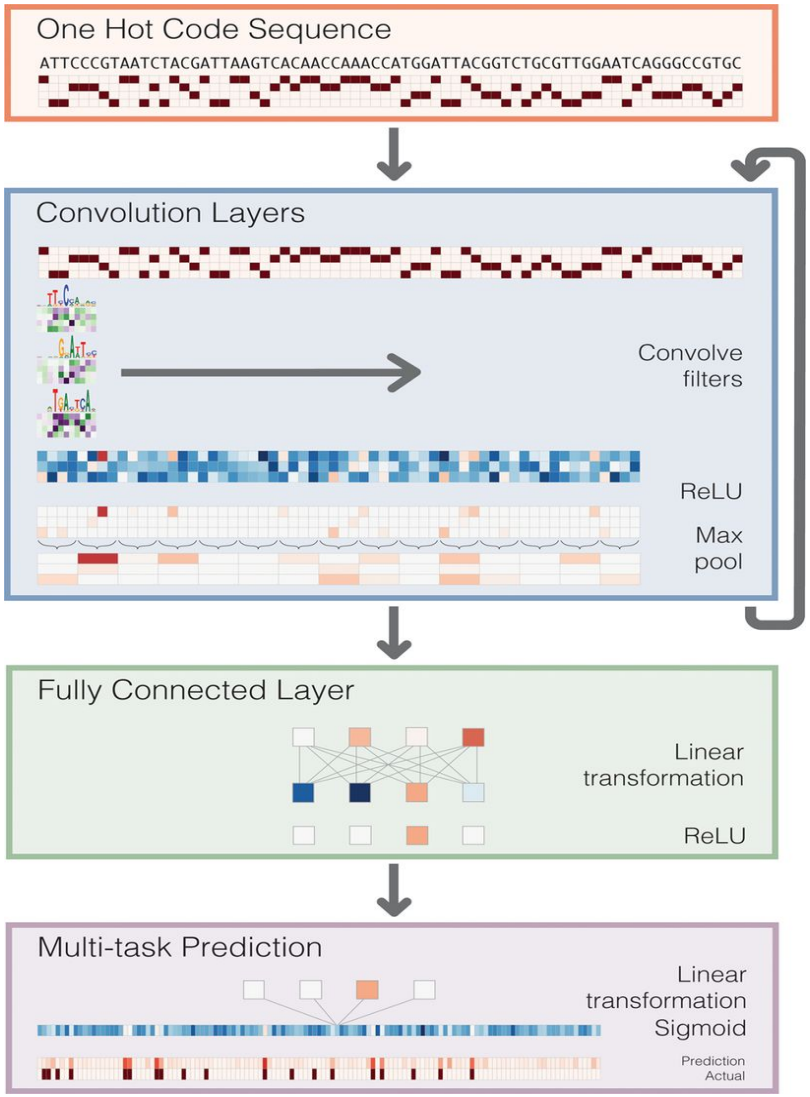


Daniel Quang and Xiaohui Xie. 2016. DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research* 44, 11.

DanQ vs DeepSEA



Basset: CNN-based Accessible Genome Analysis



1. convert the sequence to a "one hot code" representation

2. scanning weight matrices across the input matrix to produce an output matrix with a row for every convolution filter and a column for every position in the input

3. linear transformation of the input vector and apply a ReLU.

4. linear transformation to a vector of 164 elements that represents the target cells

Kelley et al. (2016). *Genome Research*, *26*(7), 990–999

DeepBind: Protein–Nucleic acid Binding Site Prediction

DeepBind is a CNN based supervised learning where

Input: segments of sequences and

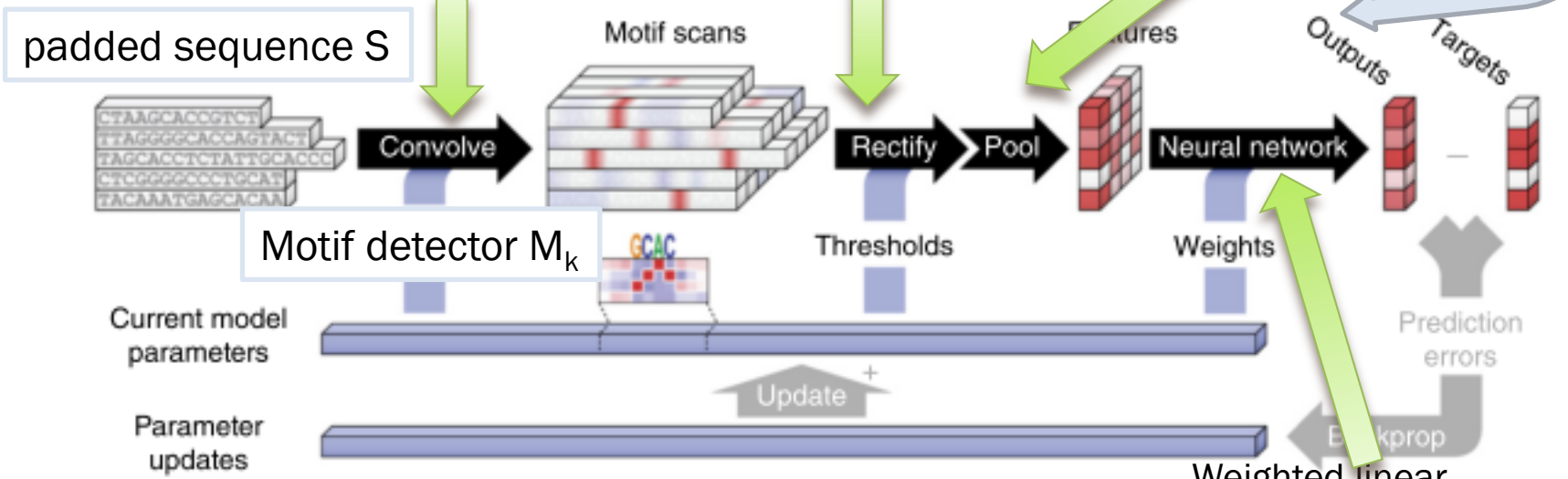
labels (output): experimentally determined binding score (ex. ChIP-seq peaks)

$$f(s) = \text{net}_W\left(\text{pool}\left(\text{rect}_b(\text{conv}_M(s))\right)\right)$$

$$X = \text{conv}_M(s) \qquad Y = \text{rectify}_b(X) \qquad z = \text{max\_pool}(Y)$$

$$X_{i,k} = \sum_{j=1}^{m}\sum_{l=1}^{4} S_{i+j,l}M_{k,j,l} \qquad Y_{i,k} = \max(0, X_{i,k} - b_k). \qquad z_k = \max(Y_{1,k}, \ldots, Y_{n,k}).$$

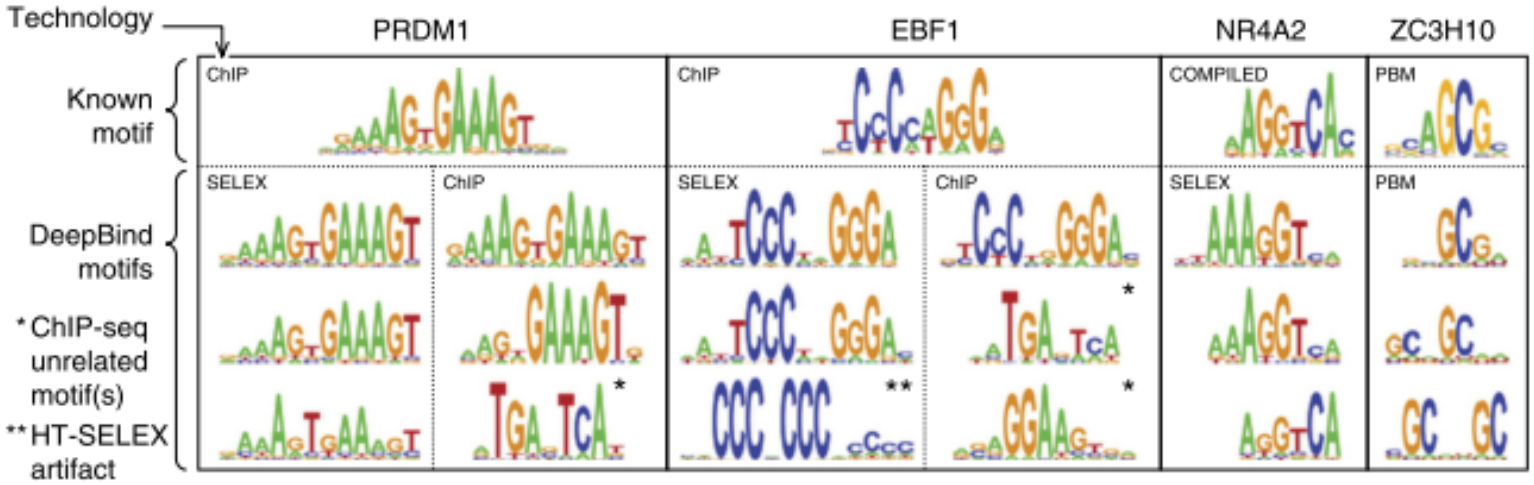
Threshold of each motif detector k



Alipanahi et al (2015) *Nature Biotechnology*, *33*(8), 831–838.

Motif Extraction capability of DEEPBIND

The trained motif detector M_k and visualization with sequence logo



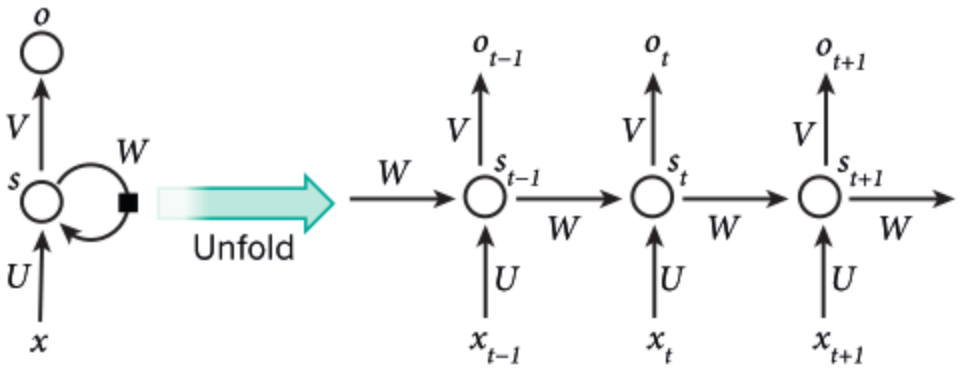
Generating sequence logo to find motifs

1. Feed all sequences from the test set through the convolutional and rectification stages of the DeepBind model,
2. Align all the sequences that passed the activation threshold for at least one position *i*.
3. Generate a position frequency matrix (PFM) and transform it into a sequence logo.

Alipanahi et al (2015) *Nature Biotechnology*, *33*(8), 831–838.

RNN for variable length Seq. Input

- Recurrent Neural Network
  - Able to work with sequence input of variable length
  - Capture long range interactions within the input sequences and acros s outputs.
  - Difficult to work with and train



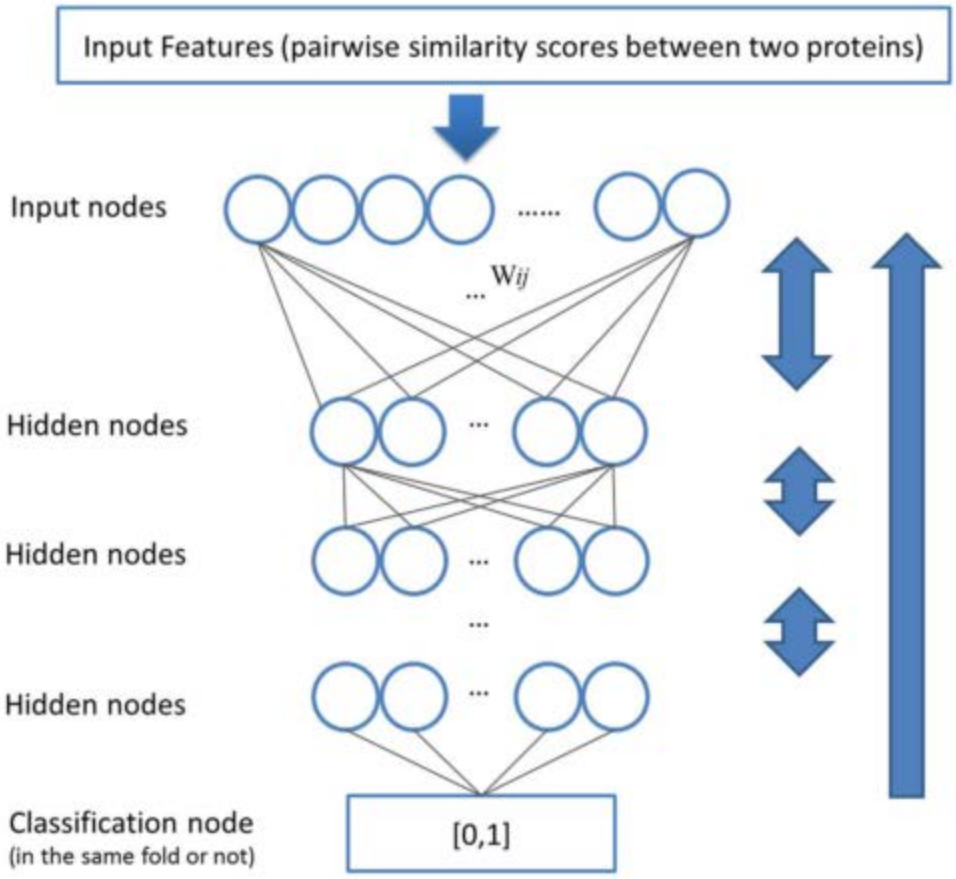
A recurrent neural network and the unfolding in time of the computation involved in its forward computation. (fig 5 of LeCun et al. 2015 Nature)

  - Not many success here

Protein Structure Prediction

- Protein structure prediction methods tend to apply uns upervised method or combination of NN methods
- Types of unsupervised DNN methods:
  - Restricted Boltzmann Machines (RBM)
  - Deep Belief Networks
- Combination methods
  - Deep Conditional Neural Fields

Stacking RBM in Protein Fold Recognition



84 features from five types of sequence alignment and/or protein structure prediction tools

Layer by layer learning with restricted Boltzmann machine (RBM).

Same fold or not

Jo et al. (2015). *Scientific Reports*, *5*, 17573.

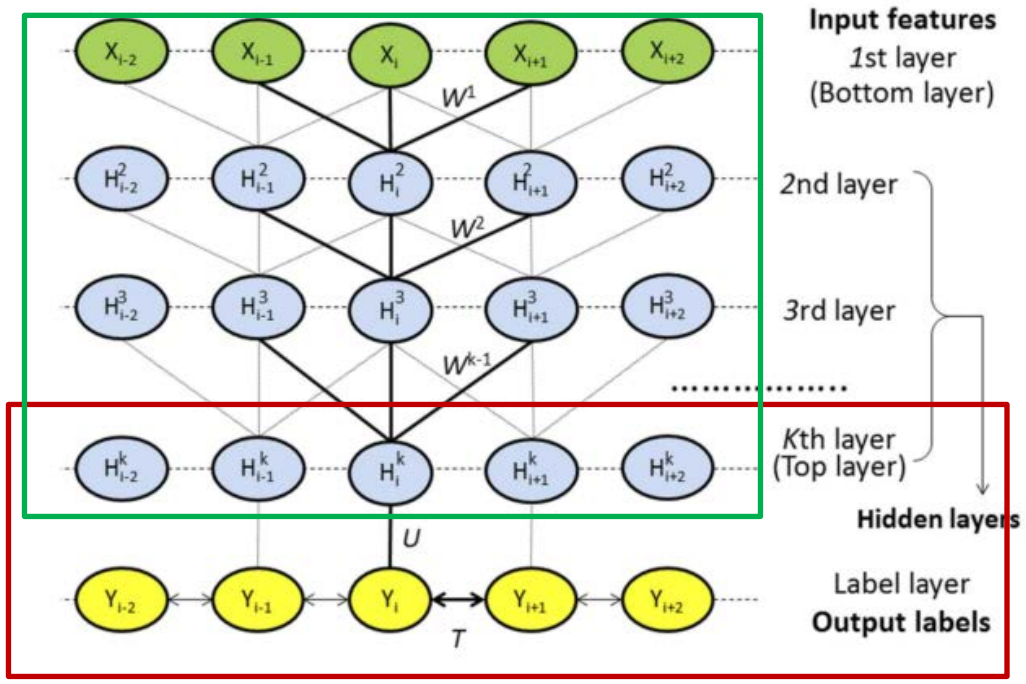
DEEPCNF: Secondary Structure Prediction

The architecture of Deep Convolutional Neural Field

fixed window size of 11:
average length of an alpha helix is around eleven residues and that of a beta strand is around six

Xi the associated input features of residue i.

5-7 layer CNN



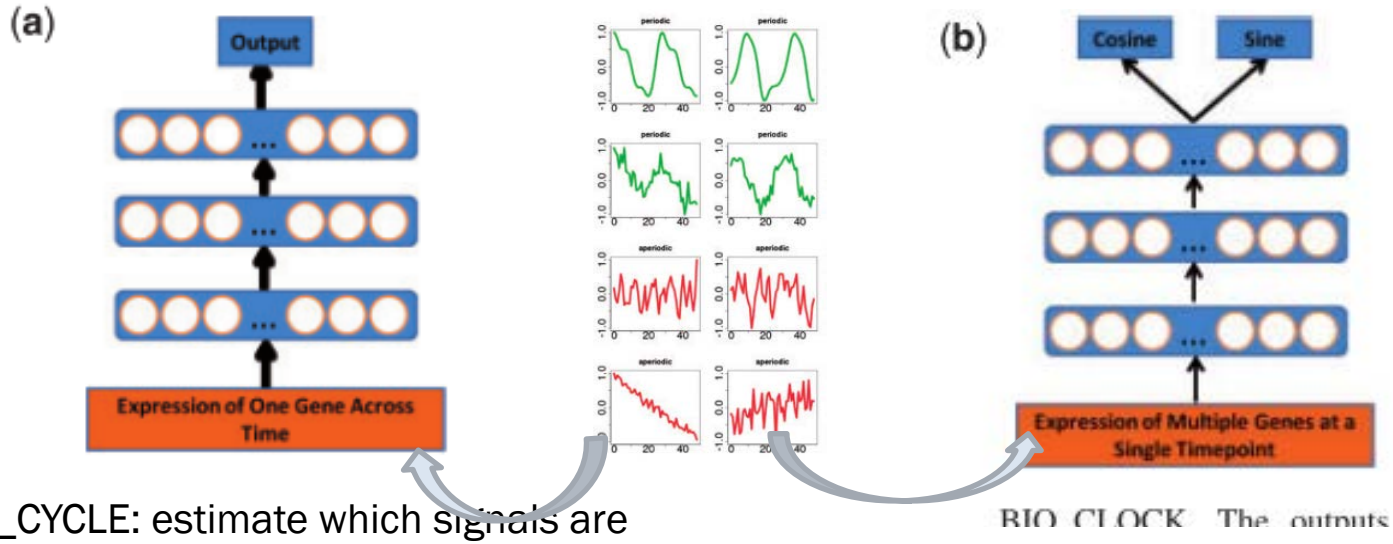
conditional random field (CRF) with U and T being the model parameters.

Calculates conditional probability of SS labels on input features

Wang et al. (2016) *Scientific Reports*, *6*(January), 18962.

Circadian Rhythms

GOAL: inferring whether a given genes oscillate in circadian fashion or not and inferring the time at which a set of measurements was taken



BIO_CYCLE: estimate which signals are periodic in high-throughput circadian experiments, producing estimates of amplitudes, periods, phases, as well as several statistical significance measures.
DATA: data sampled over 24 and 48h

BIO_CLOCK: estimate the time at which a particular single-time-point transcriptomic experiment was carried

Agostinelli, et al. (2016). *Bioinformatics*, *32*(12), i8–i17.

Cellular Image Analysis

Cellular Image Analysis

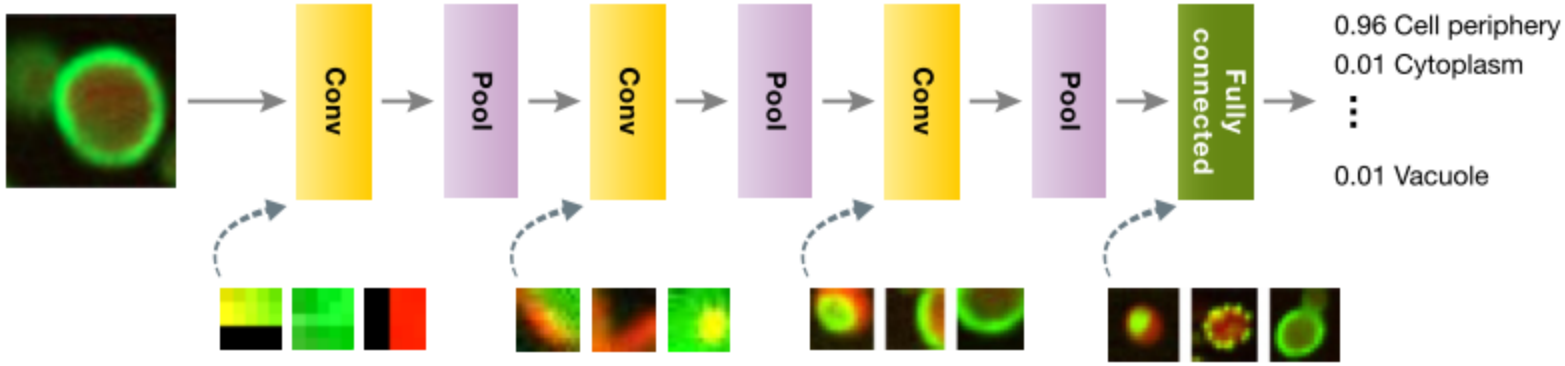
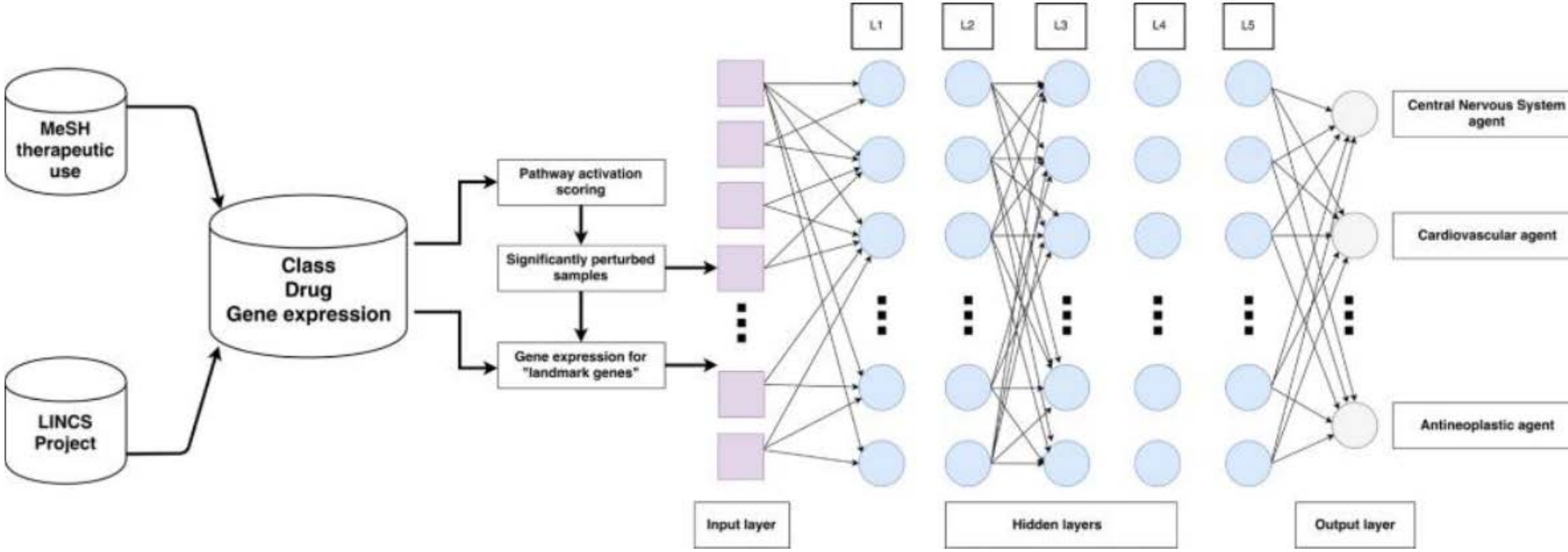


Fig 3 of Angermueller et al. (2016) *Molecular Systems Biology*, (12), 878.

Predicting Properties of Drugs

- Input: transcriptional response data sets (transcriptional p rofile)
- Goal: classify various drugs to therapeutic categories

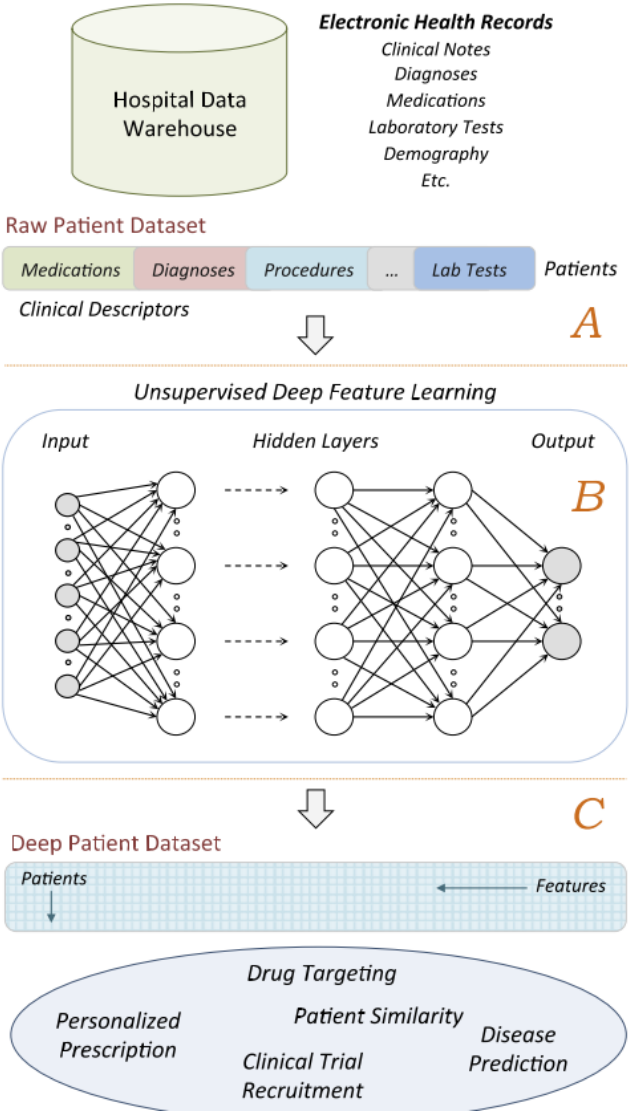


input layers of 977 and 271 neural nodes,

A. Aliper, et al. 2016. Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Molecular Pharmaceutics* 13, 7.

Deep Patient: Unsupervised Prognostic Prediction based on EHR

- Feature learning:
  - three-layer stack of denoising autoencoders
- Data: EHRs of
  - about 700,000 patients from the Mount Sinai data warehouse.
  - evaluation using 76,214 test patients comprising 78 diseases from diverse clinical domains and temporal windows
- Prediction: random forest classifier



R. Miotto et al. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific reports* 6, April.

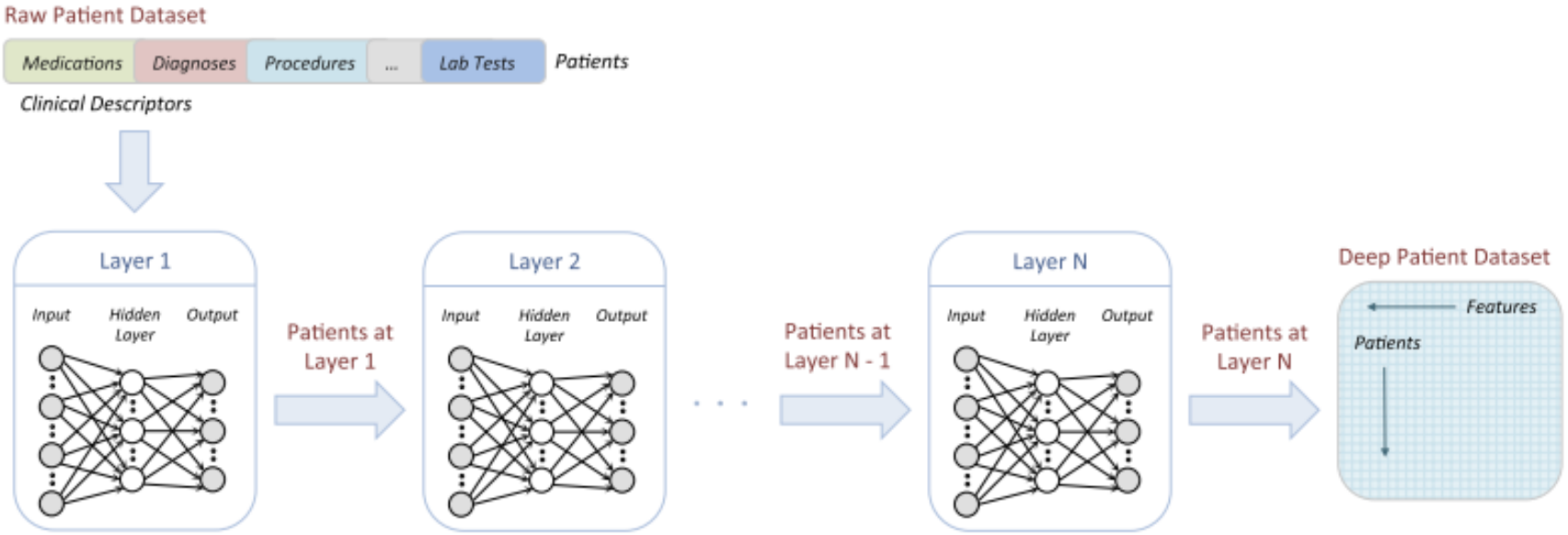


Figure 2. Diagram of the unsupervised deep feature learning pipeline to transform a raw dataset into the deep patient representation through multiple layers of neural networks. Each layer of the neural network is trained to produce a higher-level representation from the result of the previous layer.

Disease classification results

Time Interval = 1 year (76,214 patients)			
		Classification Threshold = 0.6	
Patient Representation	AUC-ROC	Accuracy	F-Score
RawFeat	0.659	0.805	0.084
PCA	0.696	0.879	0.104
GMM	0.632	0.891	0.072
K-Means	0.672	0.887	0.093
ICA	0.695	0.882	0.101
DeepPatient	**0.773***	**0.929***	**0.181***

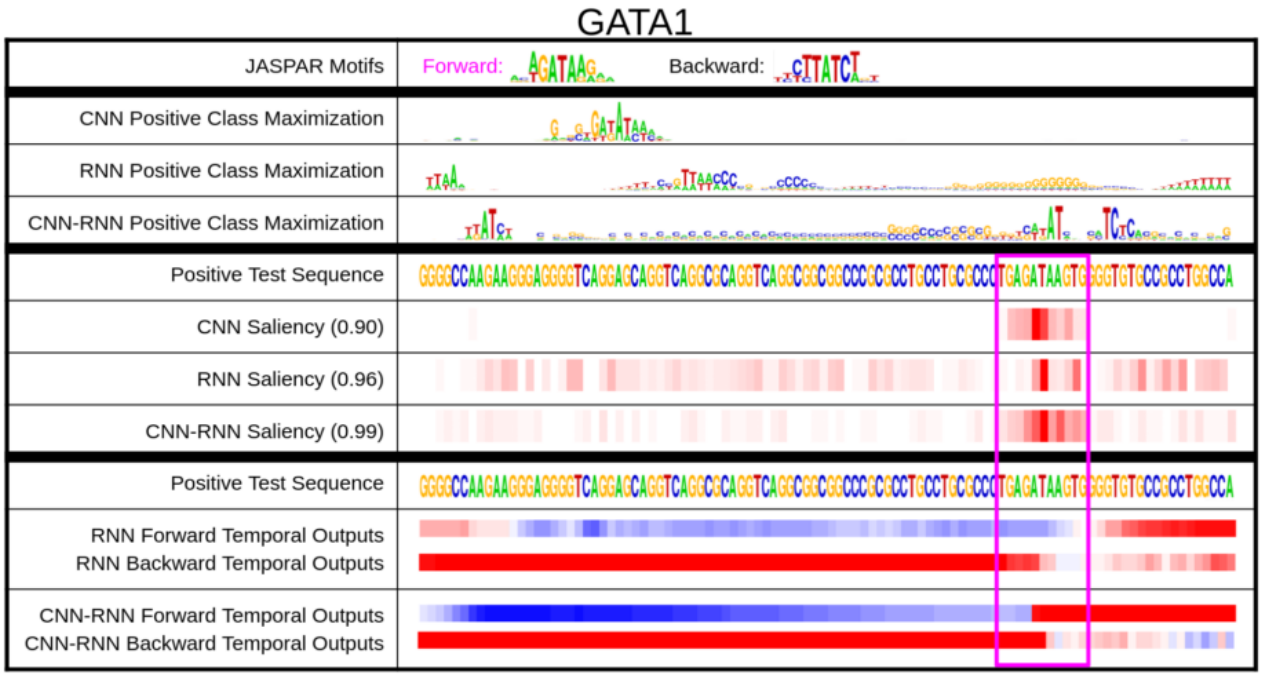
Disease classification experiment

Time Interval = 1 year (76,214 patients)			
	Area under the ROC curve		
Disease	RawFeat	PCA	DeepPatient
Diabetes mellitus with complications	0.794	0.861	**0.907**
Cancer of rectum and anus	0.863	0.821	**0.887**
Cancer of liver and intrahepatic bile duct	0.830	0.867	**0.886**
Regional enteritis and ulcerative colitis	0.814	0.843	**0.870**

[Lanchantin et al 2016]

Deep Motif Dashboard

- **Goal**: Motif visualization of Transcription Factor binding prediction
- **Models Used**: CNN, RNN, CNN-RNN
- Visualization: **Saliency Maps**, **Temporal Output Scores, Class Optimization**.



Models and Visualization Strategies

- Three Models
  - CNN
  - RNN
  - CNN-RNN (best performing)
- Visualization
  - Measuring nucleotide importance with **Saliency Maps**.
  - Measuring critical sequence positions for the classifier using **Temporal Output Scores**.
  - Generating class-specific motif patterns with **Class Optimization**.

Models - Common settings

- Input: one-hot encoded matrix of raw sequence
- Output
  - Output vector: linearly fed to a softmax function
  - Learns the mapping from the hidden space to the output class l abel space $C \in [+1, -1]$.
    - Probability indicating whether an input is a positive or a negative bindin g site (binary classification task).
- Training
  - Parameters: trained end-to-end by minimizing the negative log-likelihood over the training set.
  - Loss function optimization stochastic gradient algorithm Adam
  - Mini-batch size of 256 sequences.
  - Regularization - Dropout.

Saliency Map of CNN

Problem: Given a sequence X_0 of length $|X_0|$, and class $c \in C$, a DNN model provides a score function $S_c(X_0)$. We rank the nucleotides of X_0 based on their influence on the score $S_c(X_0)$.
Challenge: Since $S_c(X)$ is a non-linear function of X, it is hard to directly determine the influence of each nucleotide of X on Sc.
Solution: Approximated $S_c(X)$ as a linear function by computing the first-order Taylor expansion

$$S_c(X) \approx w^T X + b = \sum_{i=1}^{|X|} w_i x_i + b$$

<- a weighted sum of the input nucleotides

where w is the derivative of S_c with respect to the sequence variable X at the point X_0 (w_i, indicates the influence of that nucleotide position)

$$w = \left.\frac{\partial S_c}{\partial X}\right|_{X_0} = saliency\ map$$

Approach is similar to the methods used on images by Simonyan et al. 2013 and Baehrens et al. 2010.

Saliency Map of CNN cont.

- Derivative is simply one step of backpropagation in the DNN

- **Getting derivative values** of actual sequence:
  - Approach: pointwise multiplication of the saliency map with the one -hot encoded sequence
  - Interpretation: the influence value of the character at each position o n the output score.

- **Visualize** important each character (saliency map):
  - Approach: element-wise magnitude of the resulting derivative vector regardless of derivative direction.
  - Interpretation: indicates which nucleotides need to be changed the le ast in order to affect the class score the most.

Temporal Output Scores for RNN

- Description:
  - Visualize the output scores at each timestep (position) of a sequence.
- Assumption:
  - An imaginary time direction running from left to right
  - Each position in the sequence is a timestep
- Determine the TOS
  - The input series is constructed by using subsequences of an input X running along the imaginary time coordinate, where the subsequences start from just the first nucleotide (position), and ends with the entire sequence X.
  - TOS is calculated for each subsequences and visualized

Class-Specific Visualization

- Goal: Find the best sequence which maximizes the probability of a positive TFBS, which we call class optimization.
- Optimize $\arg\max_X S_+(X) + \lambda \|X\|_2^2$

  where $S_+(X)$ is the probability (or score) of an input sequence X (matrix) being a positive TFBS computed by the softmax equation of our trained DNN model for a specific TF.

Three Motif Extraction

For each of the three visualization methods

1. Saliency map:
   - From each positive test sequence, select the contiguous length -9 subsequence that achieves the highest sum of contiguous le ngth-9 saliency map values.
2. Temporal Output Scores:
   - For each positive test sequence, select the length-9 subsequen ce that shows the strongest score change from negative to pos itive output score.
3. Class-Specific
   - For each different TF, directly use the class-optimized sequen ce as a motif.

Results

- Training: 30,819 sequences (with an even positive/negative split), and each sequence consists of 101 DNA-base characters (A,C,G,T).
- Testing: Every dataset has 1,000 sequences

Table 1: Variations of DNN Model Hyperparameters

Model	**Conv. Layers**	**Conv. Size** (n_{out})	**Conv. filter Sizes** (k)	**Conv. Pool Size** (m)	**LSTM Layers**	**LSTM Size** (d)
Small RNN	N/A	N/A	N/A	N/A	1	16
Medium RNN	N/A	N/A	N/A	N/A	1	32
Large RNN	N/A	N/A	N/A	N/A	2	32
Small CNN	2	64	9,5	2	N/A	N/A
Medium CNN	3	64	9,5,3	2	N/A	N/A
Large CNN	4	64	9,5,3,3	2	N/A	N/A
Small CNN-RNN	1	64	5	N/A	2	32
Medium CNN-RNN	1	128	9	N/A	1	32
Large CNN-RNN	2	128	9,5	2	1	32

Results

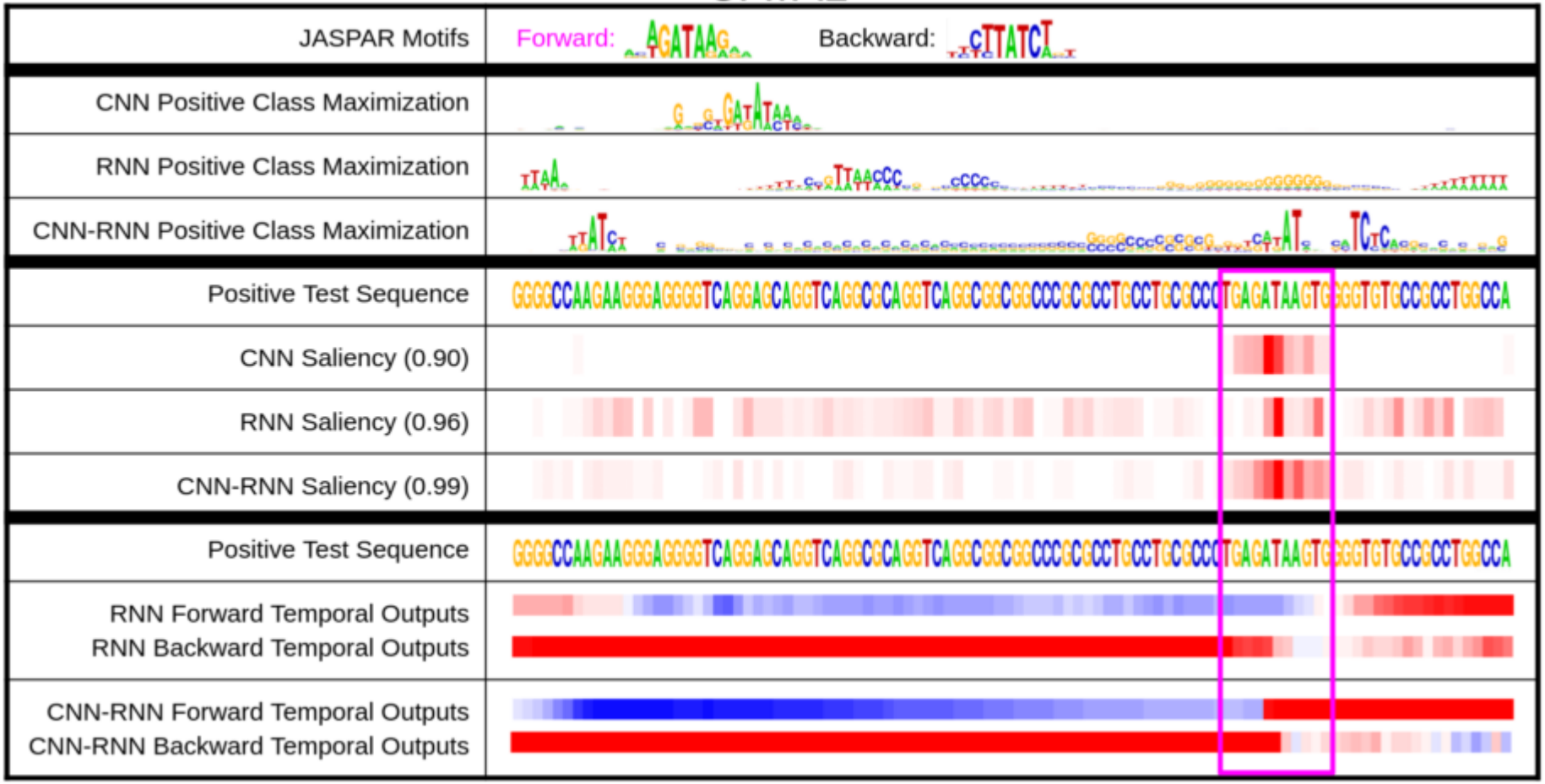
Table 2: Mean AUC scores on the TFBS classification task

Model	Mean AUC	Median AUC	STDEV
MEME-ChIP [16]	0.834	0.868	0.127
DeepBind [2] (CNN)	0.903	0.931	0.091
Small RNN	0.860	0.881	106
Med RNN	0.876	0.905	0.116
Large RNN	0.808	0.860	0.175
Small CNN	0.896	0.918	0.098
Med CNN	0.902	0.922	0.085
Large CNN	0.880	0.890	0.093
Small CNN-RNN	0.917	0.943	0.079
Med CNN-RNN	**0.925**	**0.947**	**0.073**
Large CNN-RNN	0.918	0.944	0.081

Table 3: AUC pairwise t-test

Model Comparison[3]	p-value
RNN vs MEME	5.15E-05
CNN vs MEME	1.87E-19
CNN-RNN vs MEME	4.84E-24
CNN vs RNN	5.08E-04
CNN-RNN vs RNN	7.99E-10
CNN-RNN vs CNN	4.79E-22

GATA1


JASPAR Motifs
Forward:
Backward:
CNN Positive Class Maximization
RNN Positive Class Maximization
CNN-RNN Positive Class Maximization
Positive Test Sequence
GGGGCCAAGAAGGGAGGGGTCAGGAGCAGGTCAGGCGCAGGTCAGGCGGCGGCCCGCGCCTGCCTGCGCCCTGAGATAAGTGGGGTGTGCCGCCTGGCCA
CNN Saliency (0.90)
RNN Saliency (0.96)
CNN-RNN Saliency (0.99)
Positive Test Sequence
GGGGCCAAGAAGGGAGGGGTCAGGAGCAGGTCAGGCGCAGGTCAGGCGGCGGCCCGCGCCTGCCTGCGCCCTGAGATAAGTGGGGTGTGCCGCCTGGCCA
RNN Forward Temporal Outputs
RNN Backward Temporal Outputs
CNN-RNN Forward Temporal Outputs
CNN-RNN Backward Temporal Outputs

Reference

1. Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. 2016. Interpretable Deep Models for ICU Outcome Prediction. *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2016: 371–380.
2. Jack Lanchantin, Ritambhara Singh, Beilun Wang, and Yanjun Qi. 2016. Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks. In *Pacific Symposium on Biocomputing*, 1–11. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531. 2015;.
3. Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* 33, 8: 831–838.
4. Unterthiner, T.; Mayr, A.; Klambauer, G.; Steijaert, M.; Ceulemans, H.; Wegner, J. K.; & Hochreiter, S. (2014) "Deep Learning as an Opportunity in Virtual Screening". Workshop on Deep Learning and Representation Learning (NIPS2014).