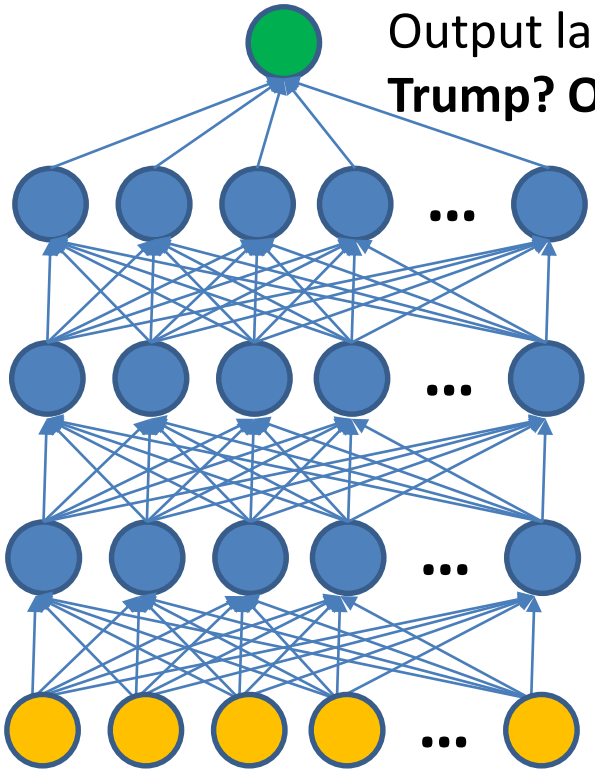
Not Just a Black Box: Interpretable Deep Learning

CS 273B / Oct 5 2016 / Avanti Shrikumar

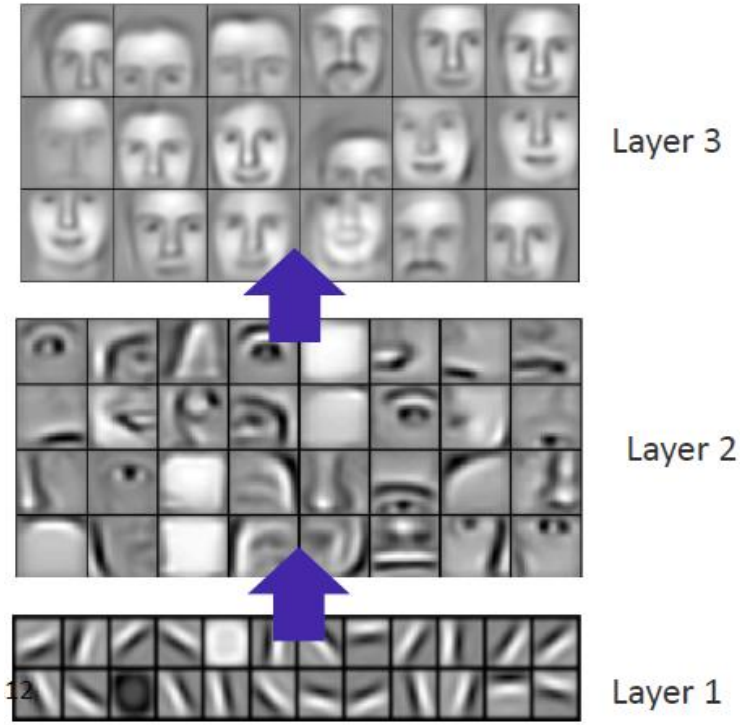
In this talk

- Compute importance scores for individual inputs in a deep learning model
- Discover patterns by summarizing importance scores

Deep neural networks (DNNs) for image classification



Input: image pixel values

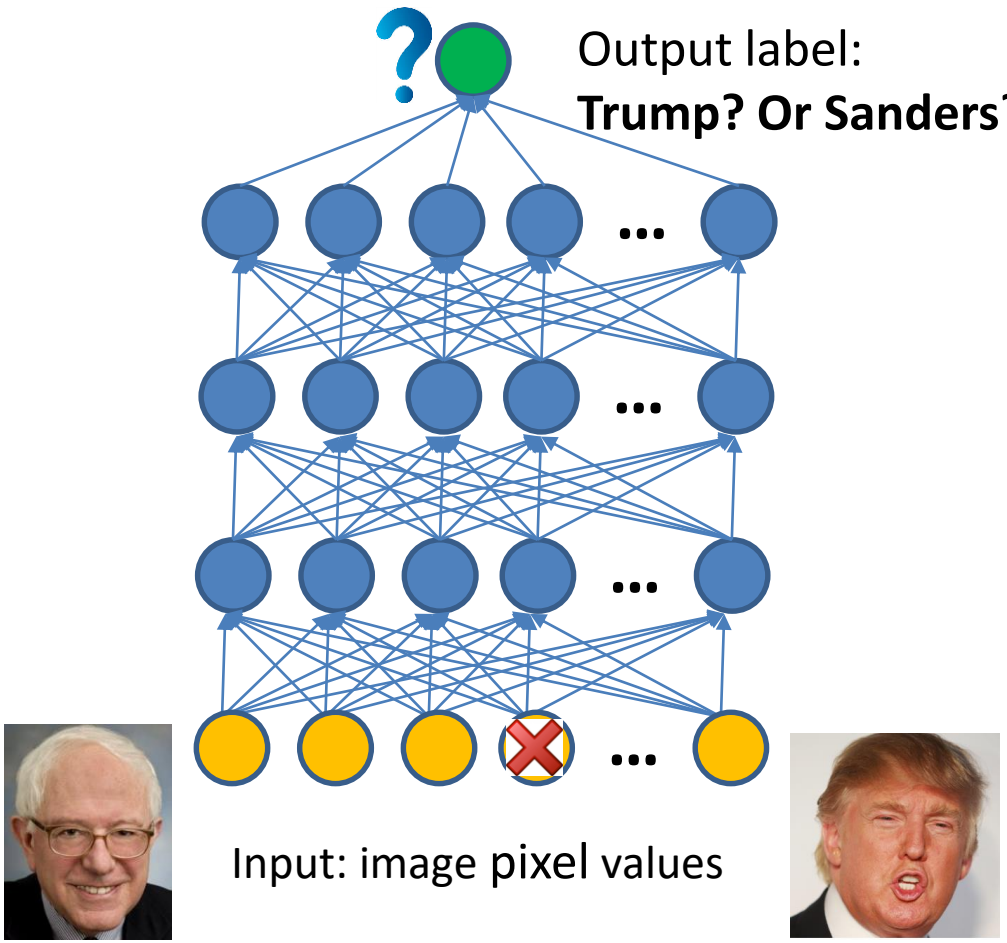


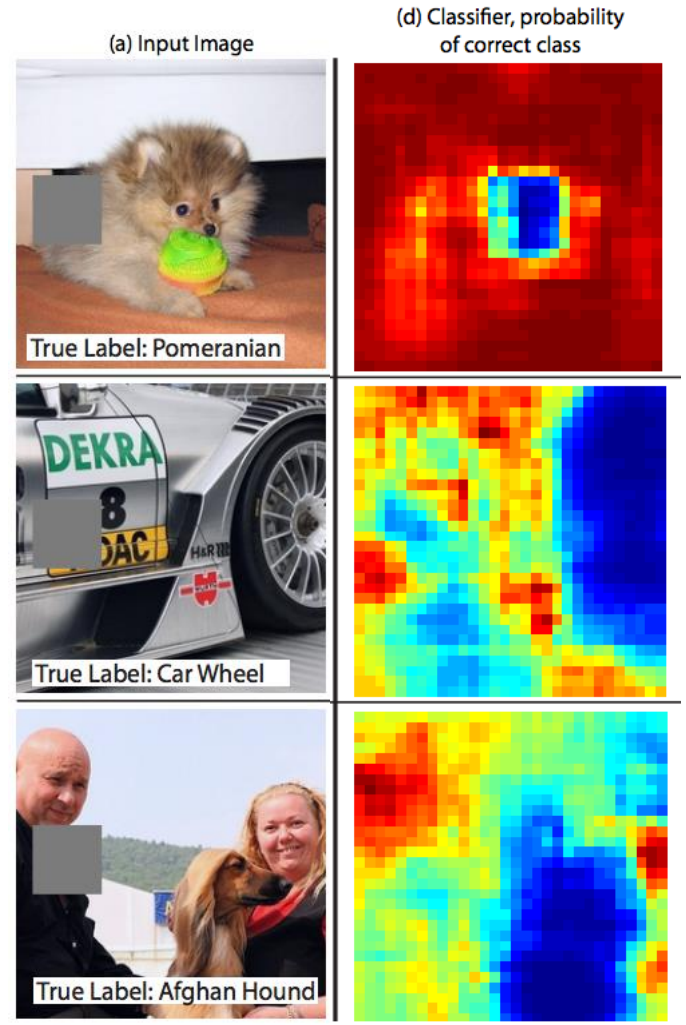
Lee et. al. (2009), ICML

Compute importance scores

How can we compute importance scores?

Idea 1: perturbation





https://arxiv.org/pdf/1311.2901v3.pdf

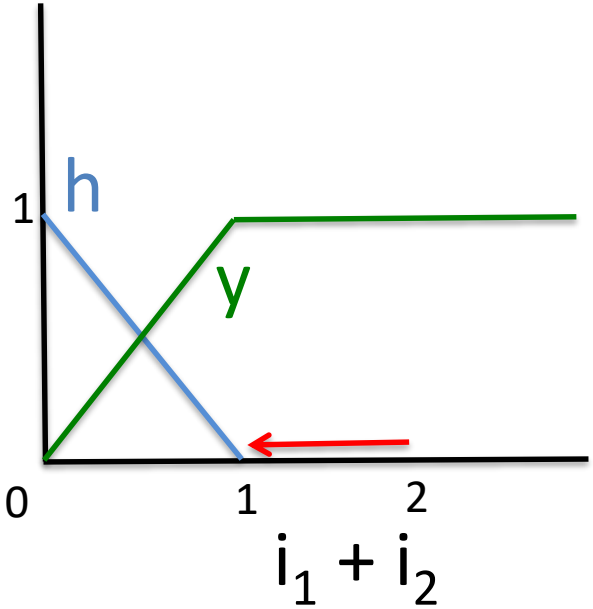
"In-silico mutagenesis" in the context of DNA sequence

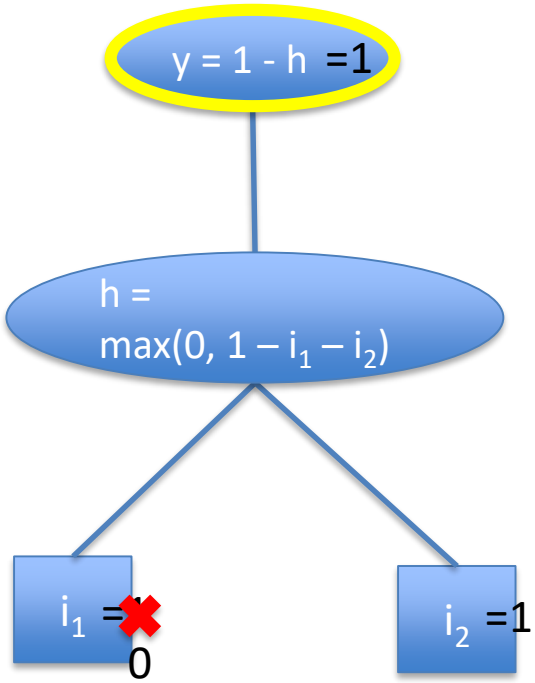
Drawbacks of perturbation

- Computational efficiency
  - Requires one forward propagation for each perturbation
- Saturation

Saturation problem illustrated

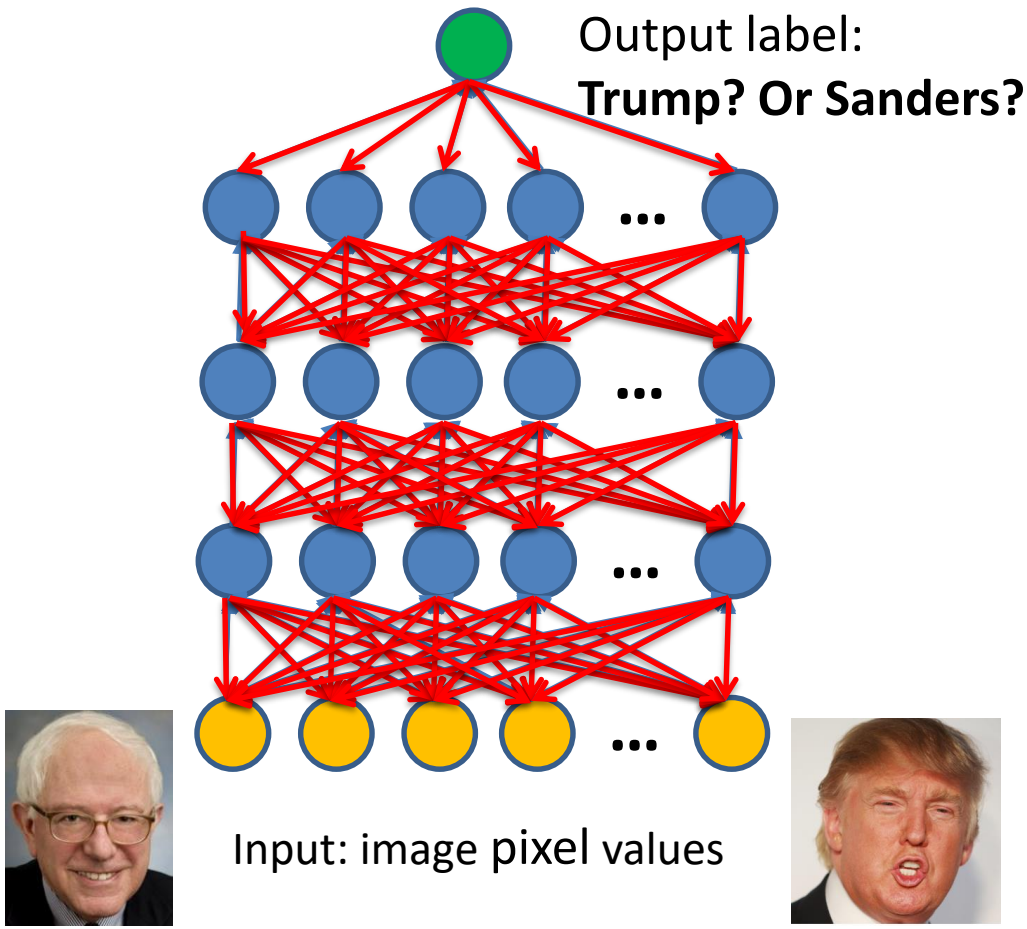
$y = (i_1 + i_2)$ when $(i_1 + i_2) < 1$
$= 1$ when $(i_1 + i_2) >= 1$





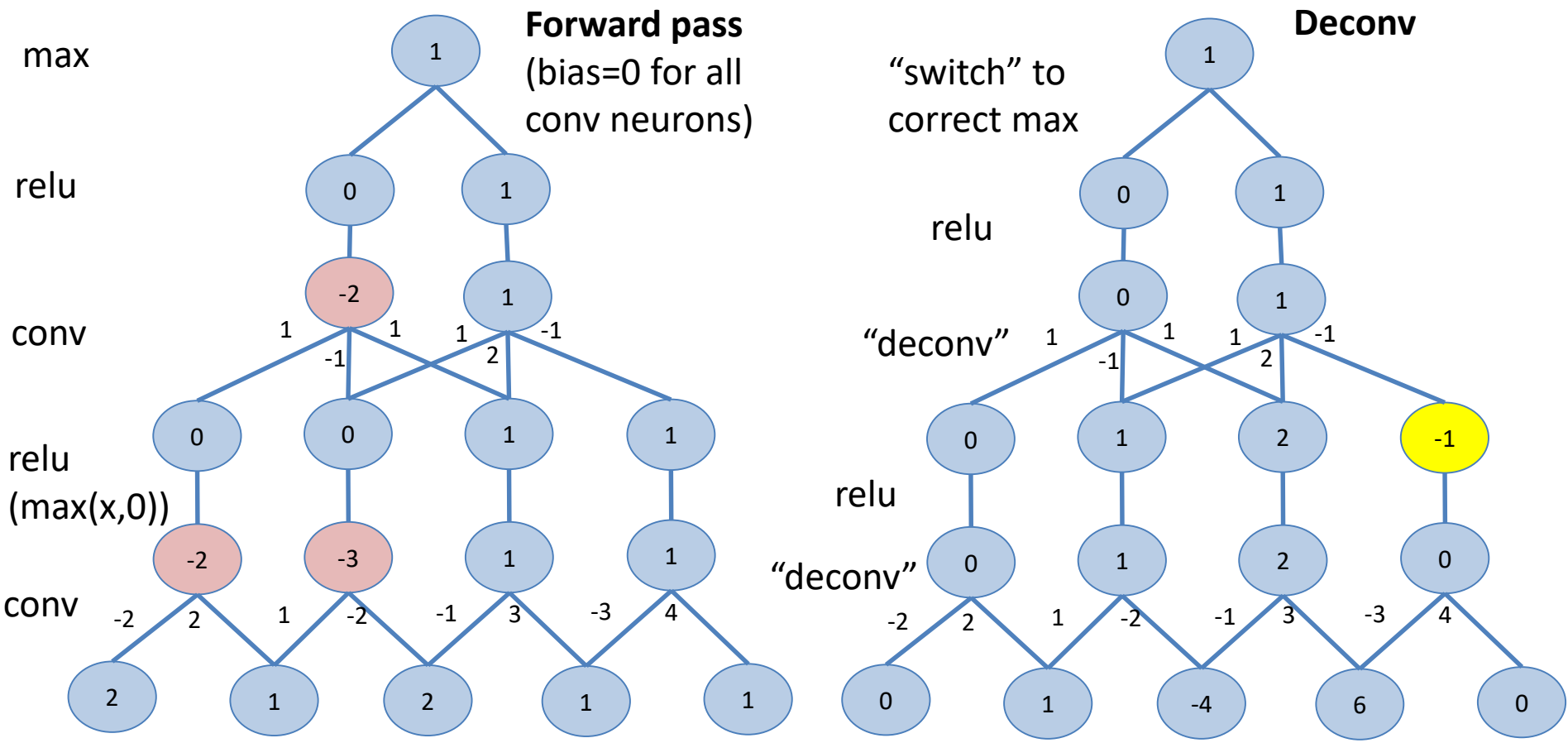
How can we compute importance scores?

Idea 2: backpropagate importance

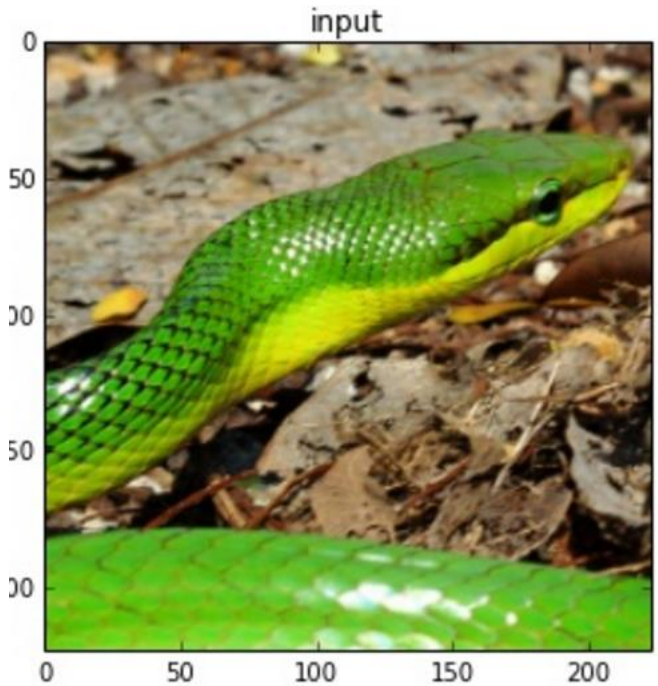


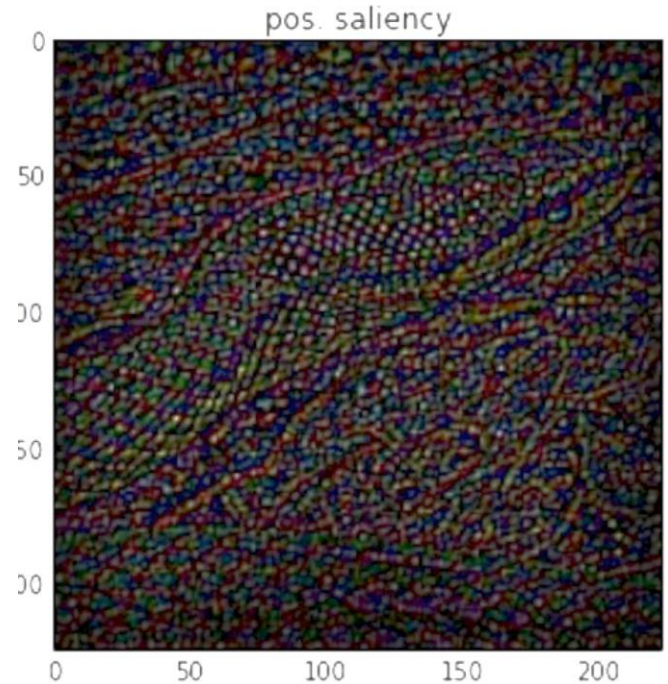
Idea 1: deconvolutional nets

- Originally applied as unsupervised learning:
  - http://www.matthewzeiler.com/pubs/cvpr2010/cvpr2010.pdf
- Adapted for visualization



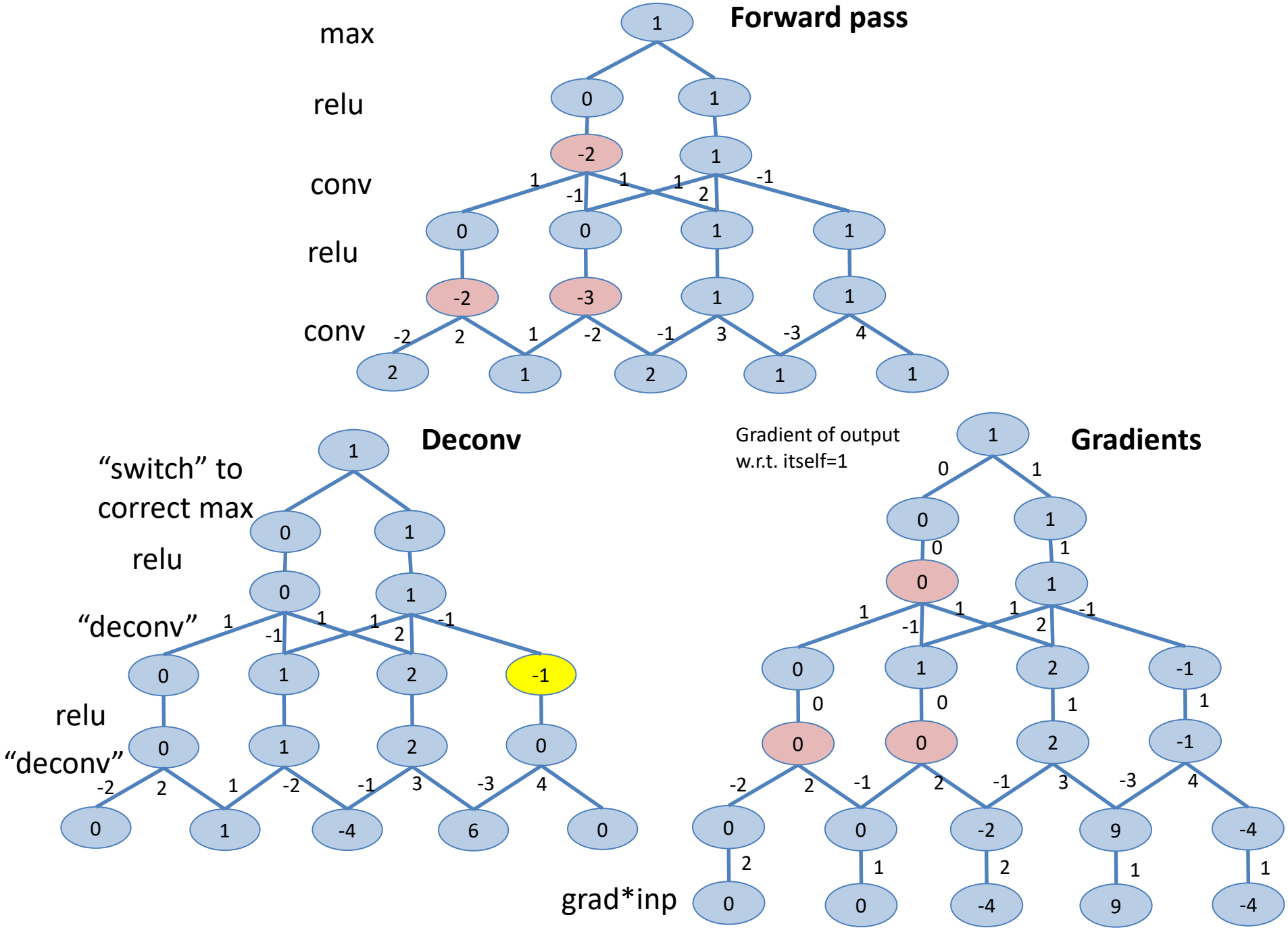
Deconv net: example

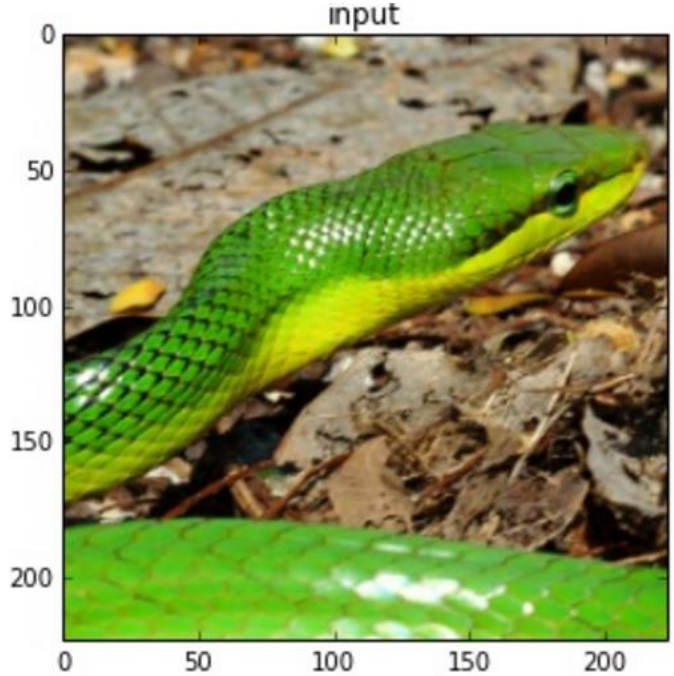




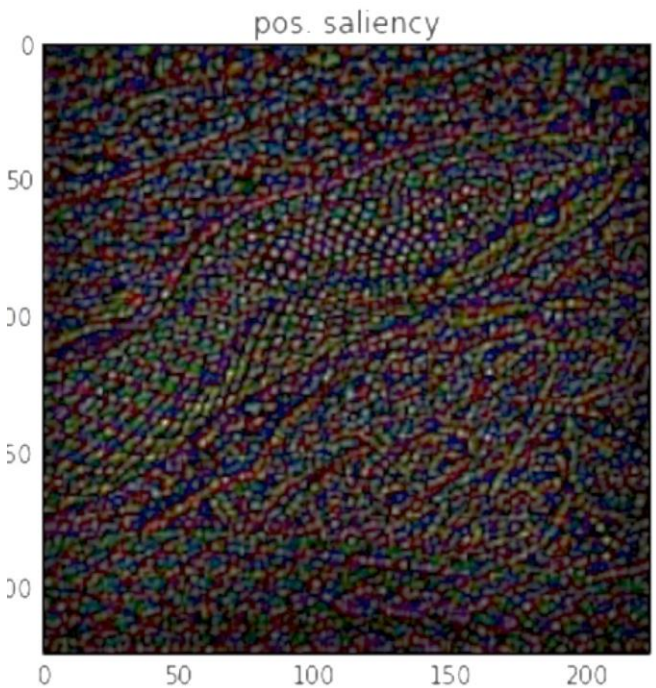
Idea 2: Gradients

- First proposed by Simonyan et al. Used to improve image segmentation.
- Equivalent to deconv nets, except in handling of ReLUs

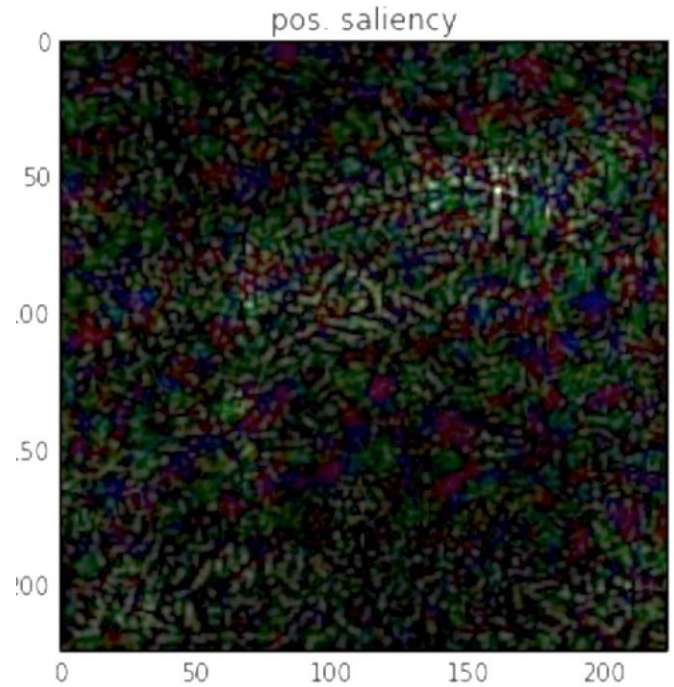

Forward pass
max
relu
conv
relu
conv
Deconv
"switch" to correct max
relu
"deconv"
relu
"deconv"
Gradients
Gradient of output w.r.t. itself=1
grad*inp



Deconv

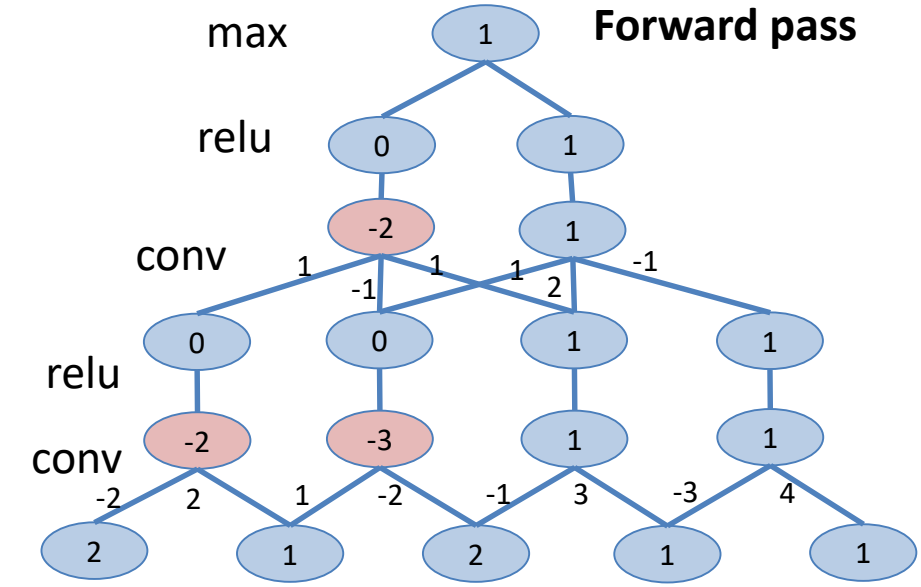


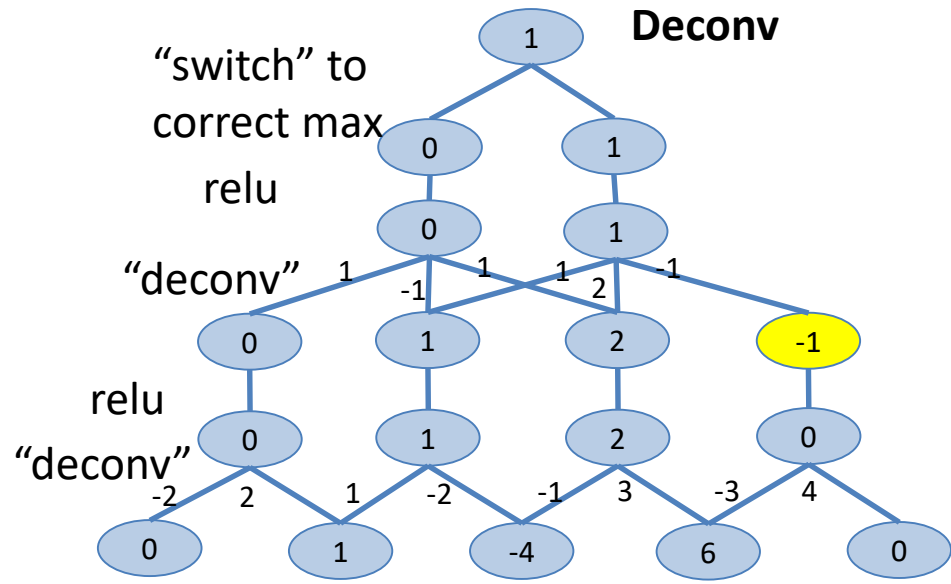
Gradients

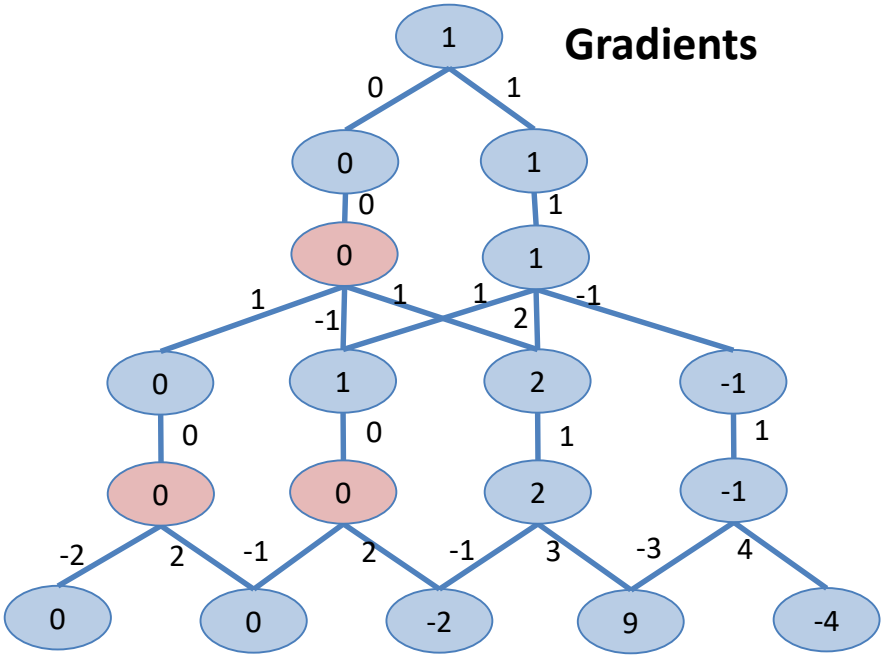


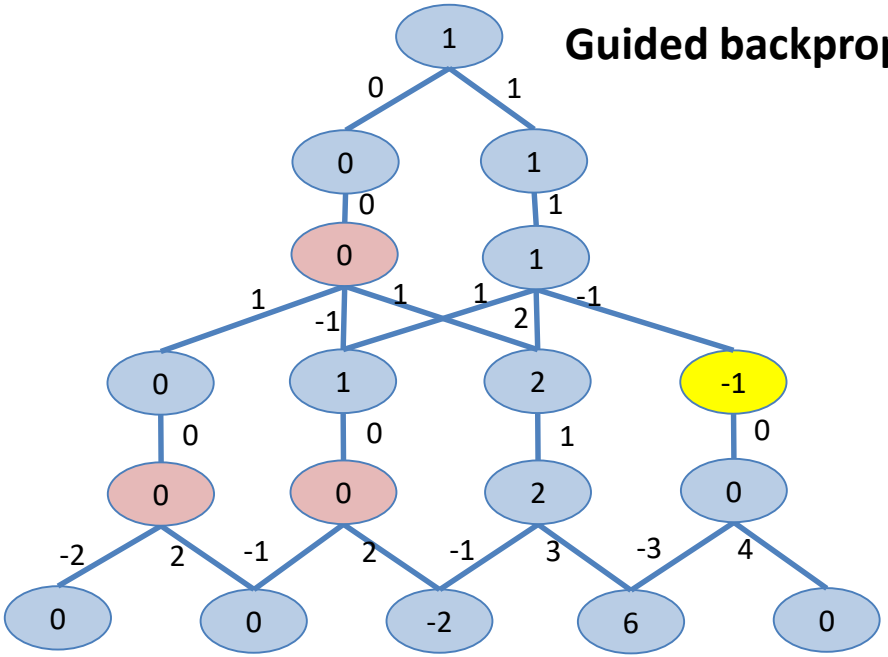
Idea 3: "Guided backpropagation"

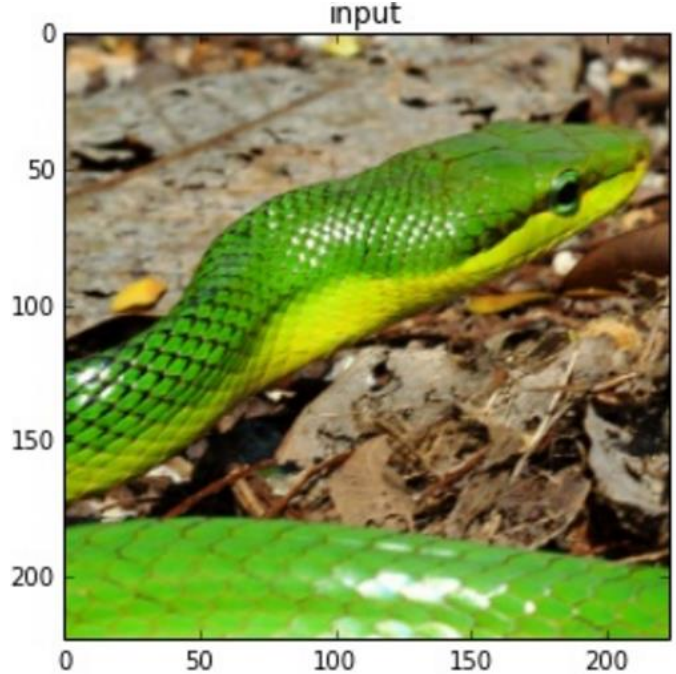
- Deconv seems to have some advantages, gradients seems to have some advantages
- Just combine the two
- When going through ReLUs, zero out gradients when input is < 0 AND when gradient is < 0


Forward pass
max
relu
conv
relu
conv

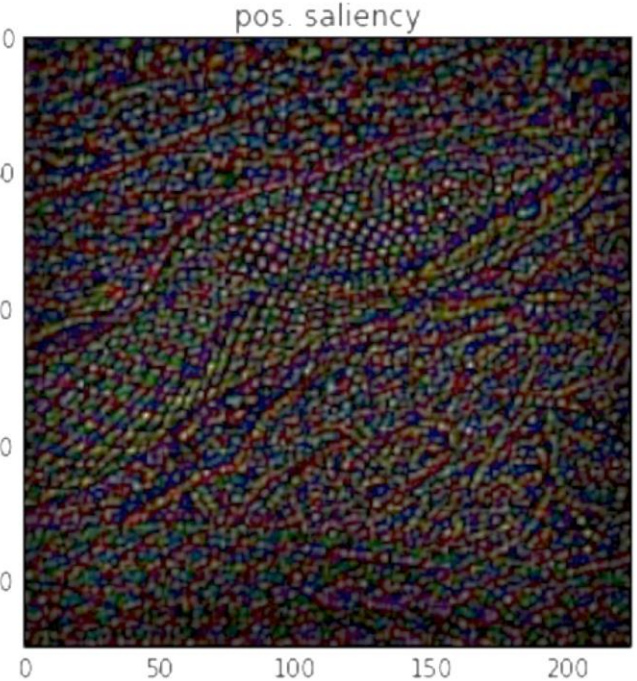

Deconv
“switch” to correct max
relu
“deconv”
relu
“deconv”


Gradients

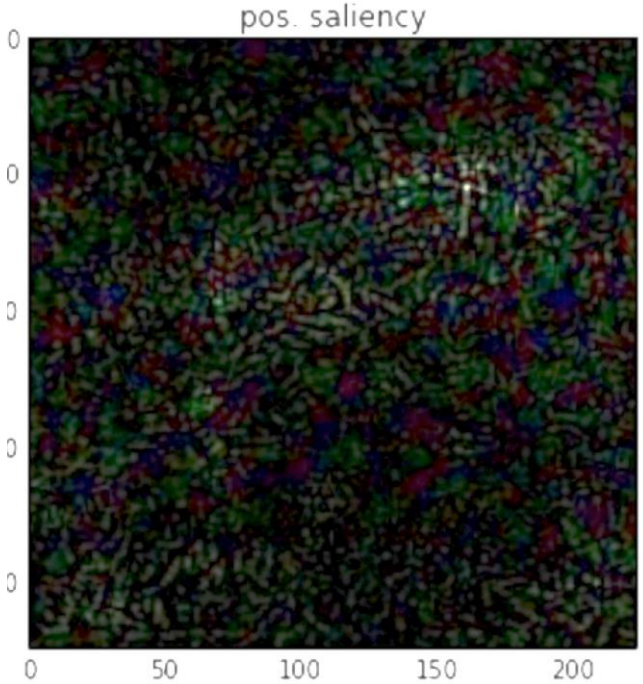

Guided backprop



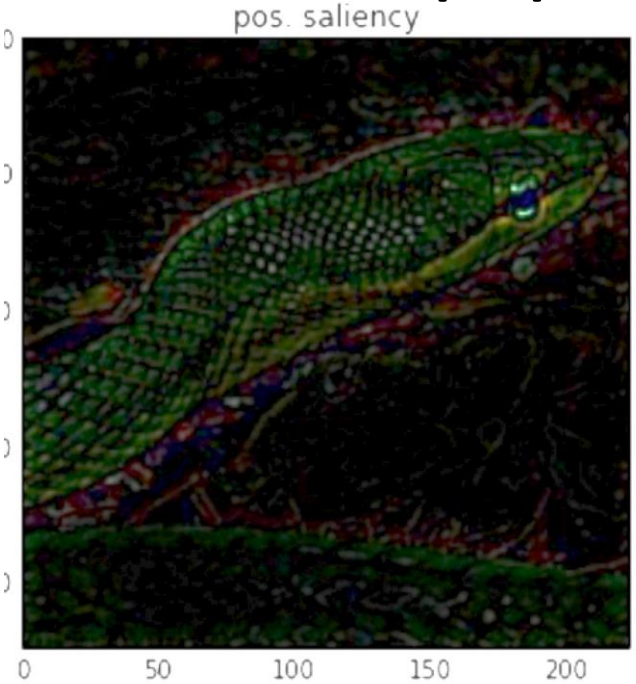
Deconv



Gradients



Guided Backprop



Summary of differences

ReLU(x) = max(0, x)

Forward-pass

1	-1	5
2	-5	-7
-3	2	4

ReLU →

1	0	5
2	0	0
0	2	4

Red = input < 0
Yellow = gradient < 0

Zero-out gradients through inactive ReLUs

-2	0	-1
6	0	0
0	-1	3

←

-2	3	-1
6	-3	1
2	-1	3

Gradients

Zero-out negative gradients & ignore ReLU

0	3	0
6	0	1
2	0	3

←

-2	3	-1
6	-3	1
2	-1	3

Deconv

(Combined)
Zero-out inactive ReLUs
Zero-out negative gradients

0	0	0
6	0	0
0	0	3

←

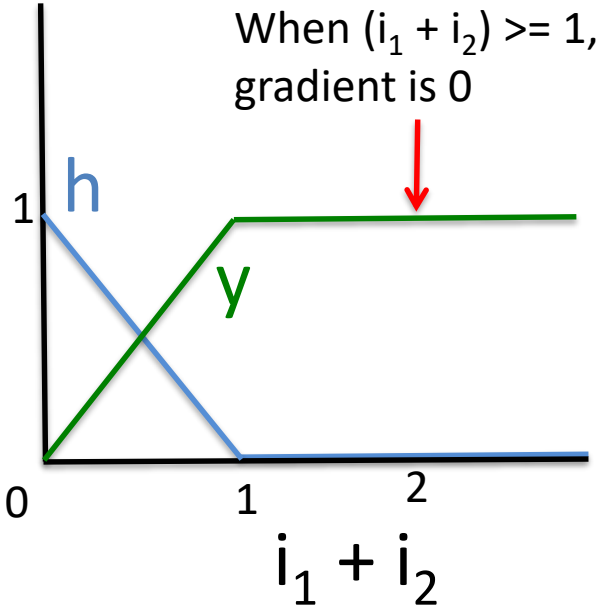
-2	3	-1
6	-3	1
2	-1	3

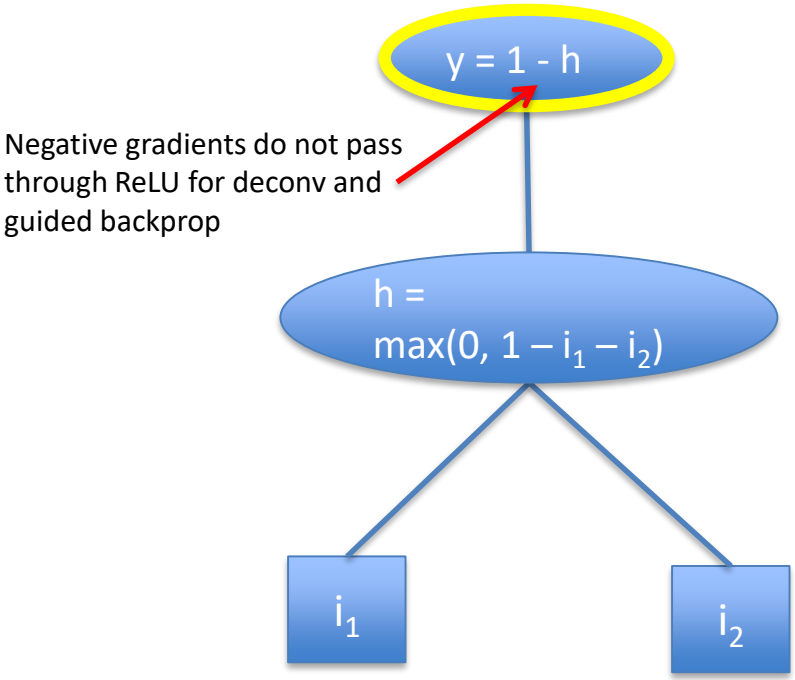
Guided backprop

No consideration of nonlinearities other than ReLU
No special handling of biases

Example failure-mode: saturation

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



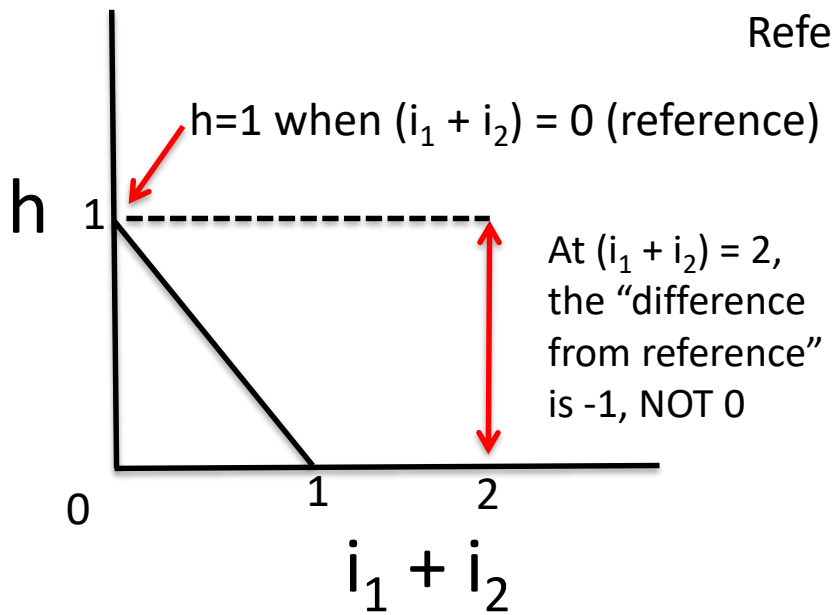


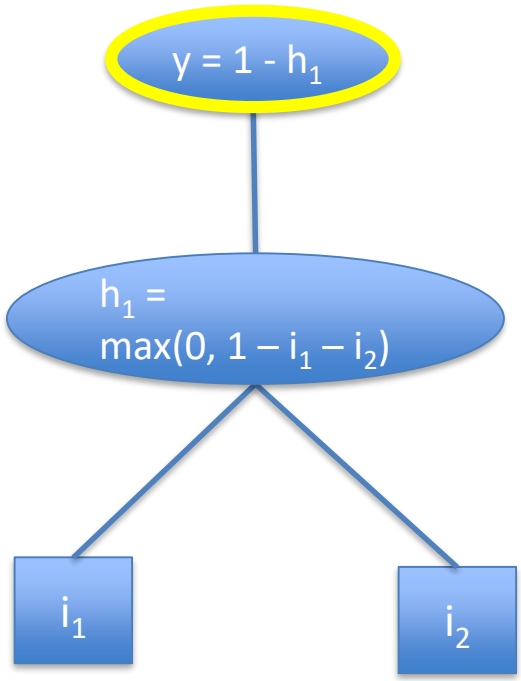
Idea 4: DeepLIFT

- Moves away from gradients, relying on "difference from reference"
- Solves saturation problem and other problems
- Equivalent to gradient*input when biases in the network are all zero

The DeepLIFT solution: difference from reference

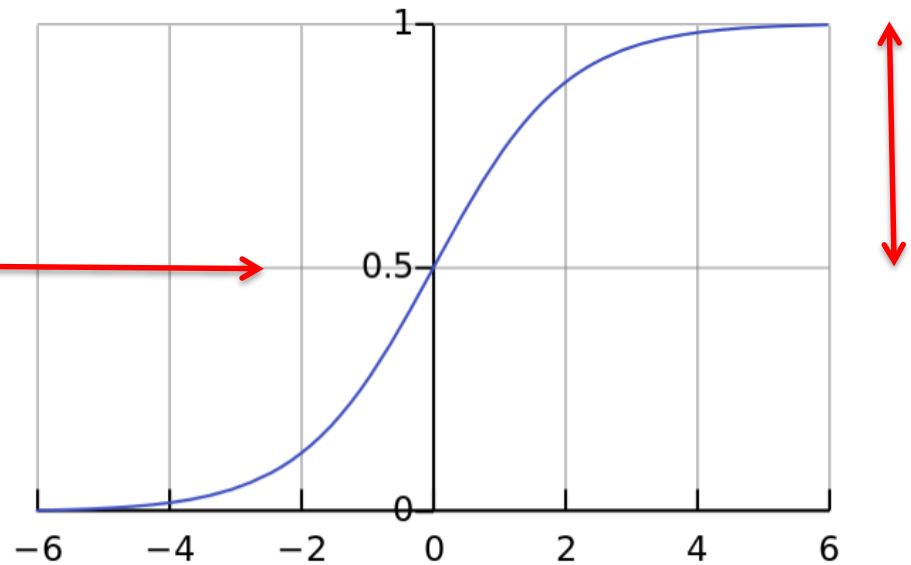
$y = (i_1 + i_2)$ when $(i_1 + i_2) < 1$
$= 1$ when $(i_1 + i_2) >= 1$
Reference: $i_1=0$ & $i_2=0$





DeepLIFT generalizes to all activations

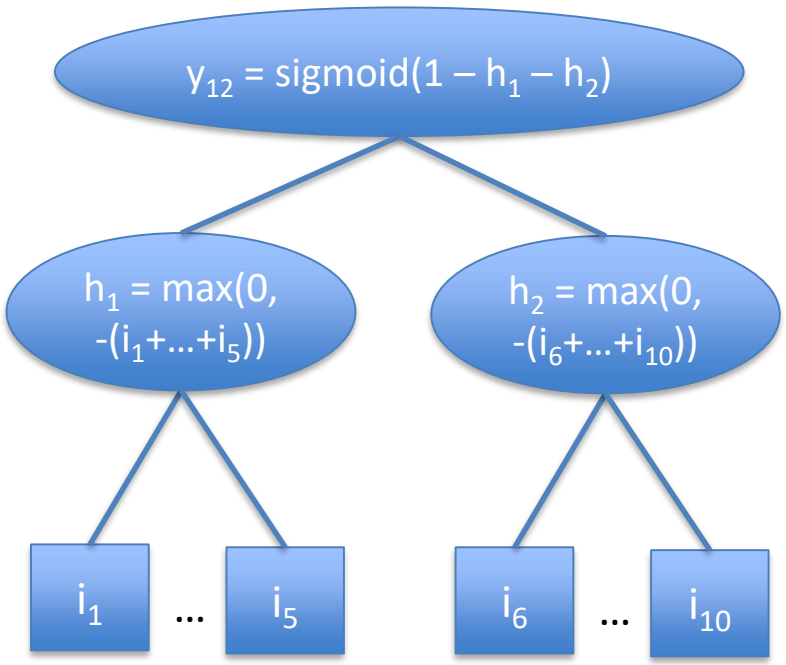
Sigmoid is 0.5 when input is 0



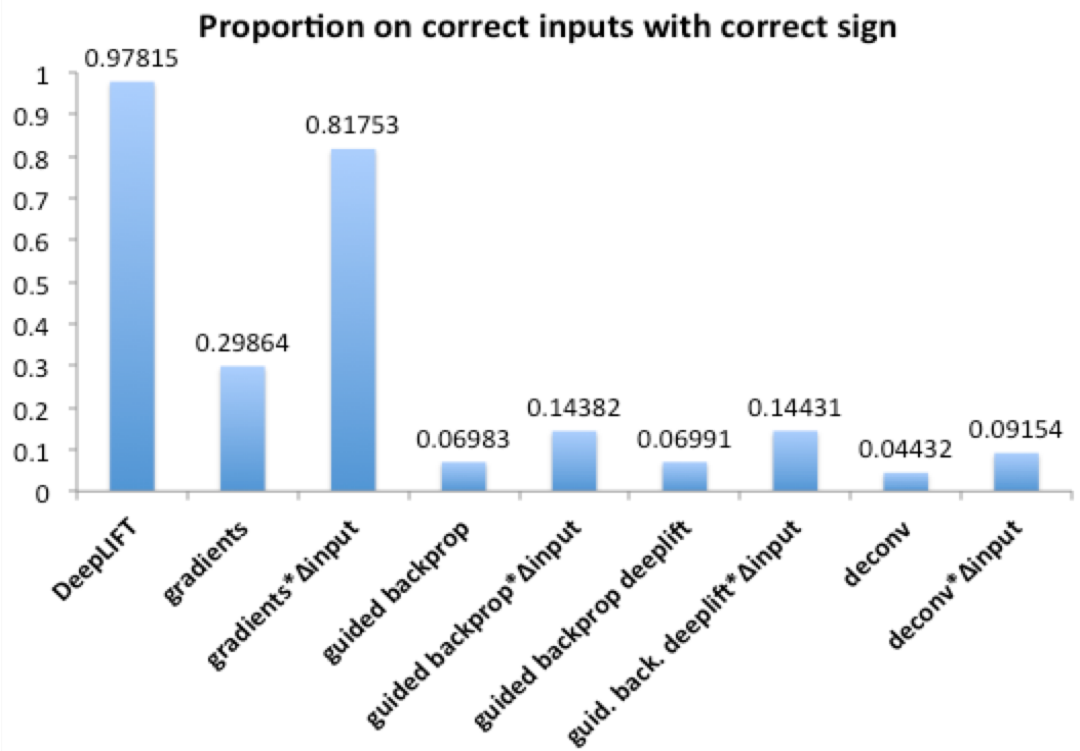
“difference from reference” is +0.5 when inputs is >> 0 (assuming reference input of 0)

Saturation simulation

- 50 inputs: uniformly sampled from -0.5 to 0.5
- 10 "rules": Each rule fires if an exclusive set of 5 inputs has a sum > 0.
- 10-choose-2: 45 outputs, each corresponding to a pair of rules. The label is 1 if both rules in the pair fire, and 0 otherwise



For a given output, what proportion of total importance over all inputs is attributed to the correct 10 inputs with correct sign?



“Multipliers”: alternative to gradients

$$A_y = f(A_{x_1}, A_{x_2}, \ldots, A_{x_i})$$

Let $A^0_{x_i}$ represent the “reference” activation of x_i

$$A^0_y = f(A^0_{x_1}, A^0_{x_2}, \ldots, A^0_{x_i})$$

$$\Delta_{x_i} = A_{x_i} - A^0_{x_i}$$

$$\Delta_y = A_y - A^0_y$$

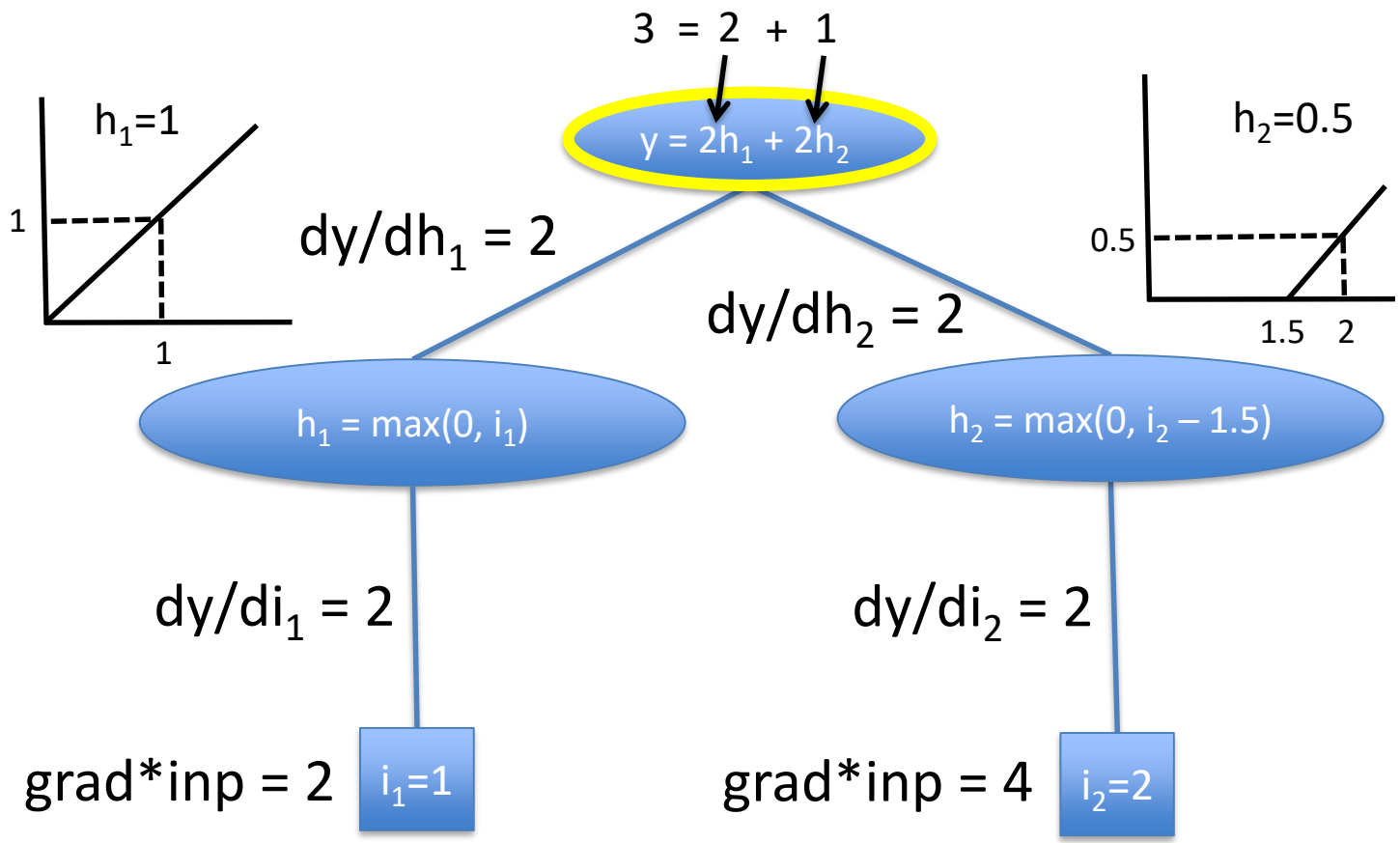
$C_{x_i y}$ is the portion of Δ_y attributed to Δ_{x_i}

Partial derivatives	DeepLIFT multipliers
$\frac{\partial_y}{\partial_{x_i}} = \frac{\text{change in } y \text{ due to } x_i}{\text{change in } x_i}$	$m_{x_i y} = \frac{C_{x_i y}}{\Delta x_i}$
$\frac{\partial_y}{\partial_w} = \sum_i \frac{\partial_{x_i}}{\partial_w} \frac{\partial_y}{\partial_{x_i}}$ (chain rule)	$m_{wy} = \sum_i m_{wx_i} m_{x_i y}$

Multiplier definitions

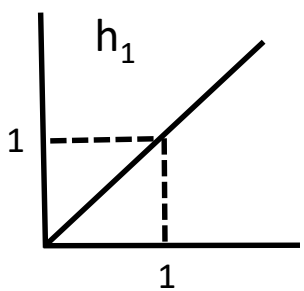
Function	Gradient	DeepLIFT multiplier
Linear $A_y = b + \sum_i w_i A_{x_i}$	$\frac{\partial_{A_y}}{\partial_{A_{x_i}}} = w_i$	$C_{x_i y} = w_i \Delta_{x_i}$ $m_{x_i y} = \frac{C_{x_i y}}{\Delta_{x_i}} = w_i$
ReLU $A_y = \max(0, A_x)$	$\frac{\partial_{A_y}}{\partial_{A_x}} = 1 \text{ if } x > 0$ $= 0 \text{ if } x \leq 0$	$C_{xy} = \Delta_y$ $m_{xy} = \frac{\Delta_y}{\Delta_x}$

Example failure-mode 2: thresholds



Contributions from i_1 & i_2 are 2 and 1 respectively, but gradient*Δinput is 2 and 4 respectively

Solution with DeepLIFT



ref = 0
$\Delta h_1 = 1$

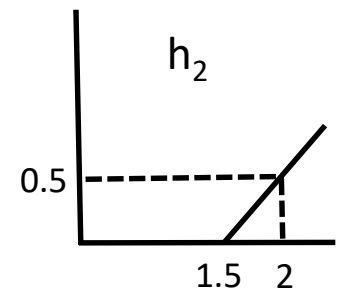
3 = 2 + 1

$y = 2h_1 + 2h_2$

ref = 0
$\Delta y = 3$

$m_{h_1} y = 2$

$m_{h_2} y = 2$



ref = 0
$\Delta h_2 = 0.5$

$h_1 = \max(0, i_1)$

$h_2 = \max(0, i_2 - 1.5)$

$$m_{i_1 h_1} = \frac{\Delta h_1}{\Delta i_1} = 1$$

$$m_{i_2 h_2} = \frac{\Delta h_2}{\Delta i_2} = \frac{1}{4}$$

$i_1 = 1$

ref = 0
$\Delta i_1 = 1$

$i_2 = 2$

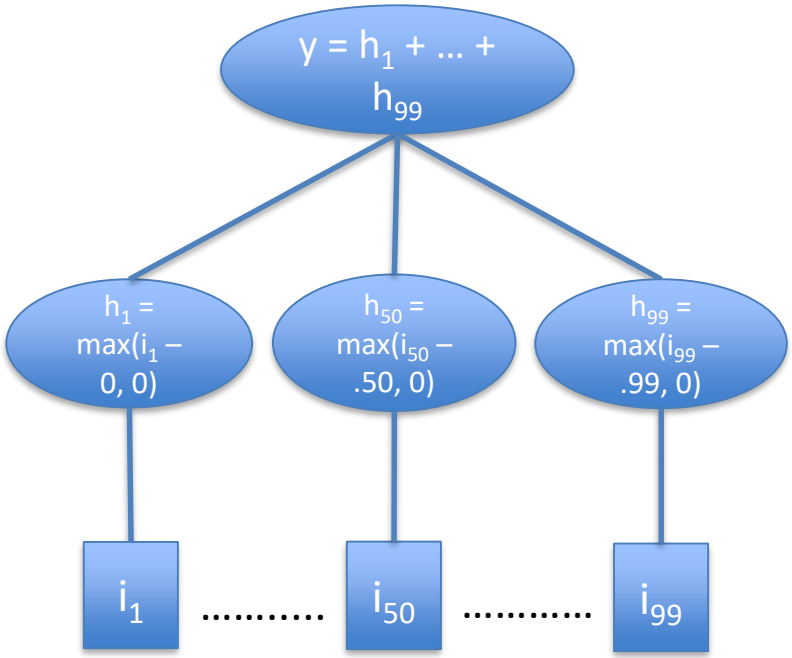
ref = 0
$\Delta i_2 = 2$

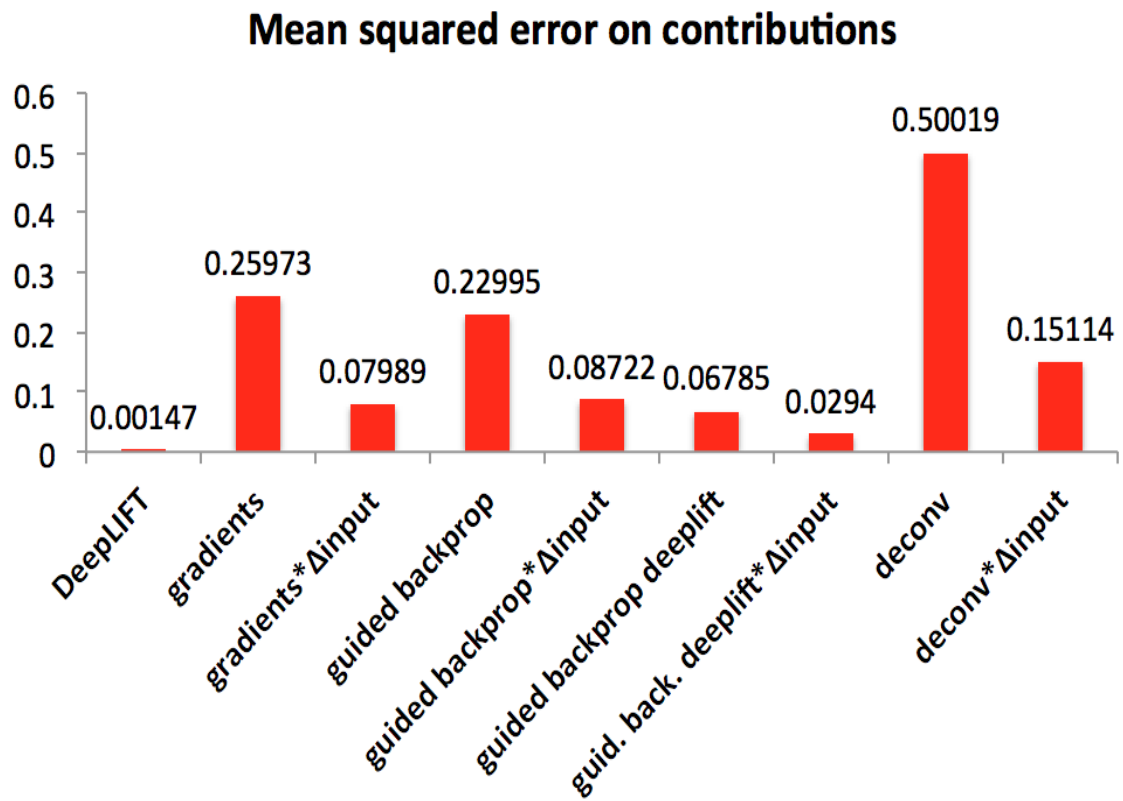
$$m_{i_1 y} = m_{i_1 h_1} m_{h_1 y} = 2$$
$$C_{i_1 y} = m_{i_1 y} \Delta_{i_1} = 2$$

$$m_{i_2 y} = m_{i_2 h_2} m_{h_2 y} = 0.5$$
$$C_{i_2 y} = m_{i_2 y} \Delta_{i_2} = 1$$

Threshold simulation

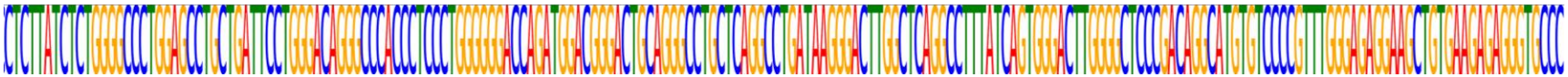
- 100 inputs x_i, uniformly sampled from 0 to 1
- Output is $\sum_i \max\left(x_i - \frac{i}{100}, 0\right)$



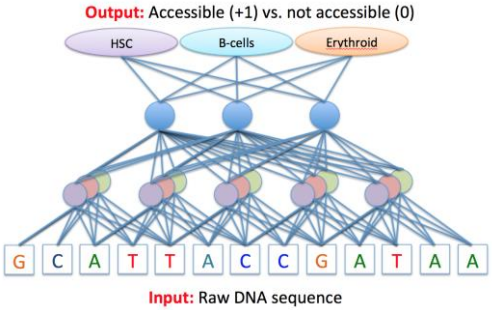


Genomics: DeepLIFT highlights essential DNA bases

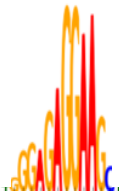
Input



Model + DeepLIFT



Output

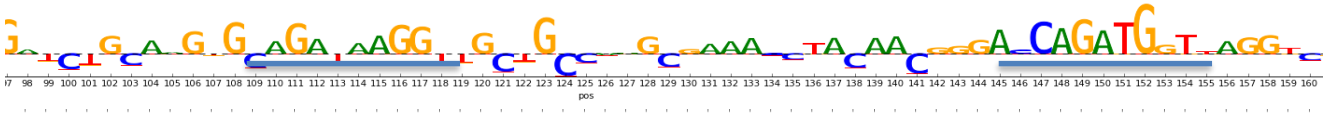


DeepLIFT captures essential features that gradients miss

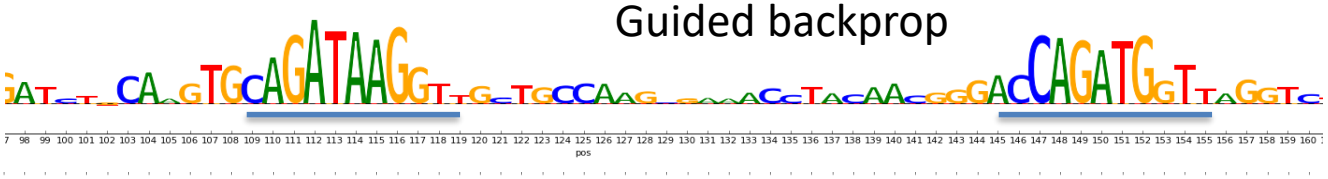
Simulation: random background sequence with and/or

Positive set: BOTH AND

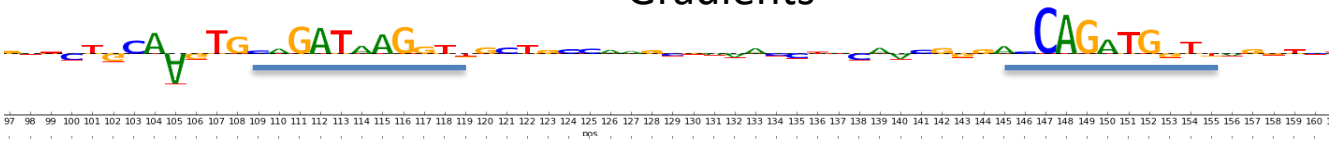
Deconv nets



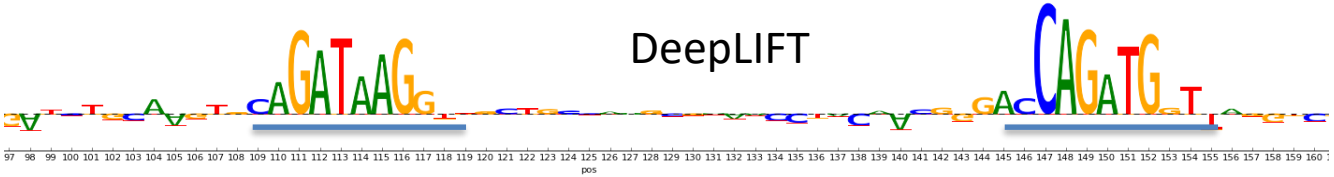
Guided backprop



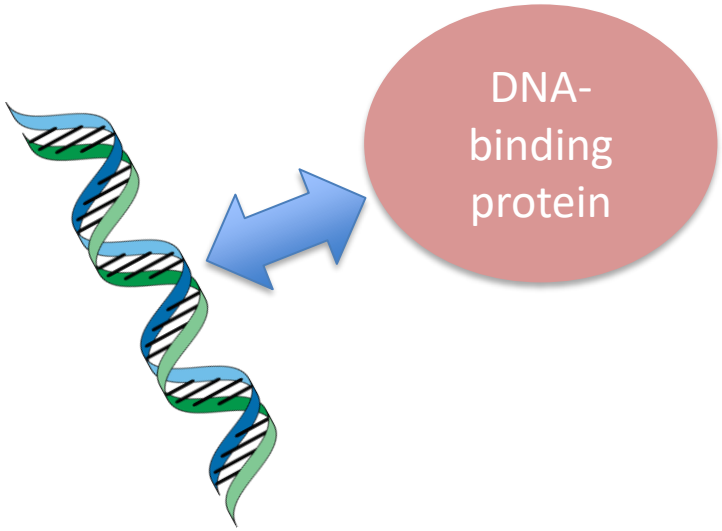
Gradients



DeepLIFT



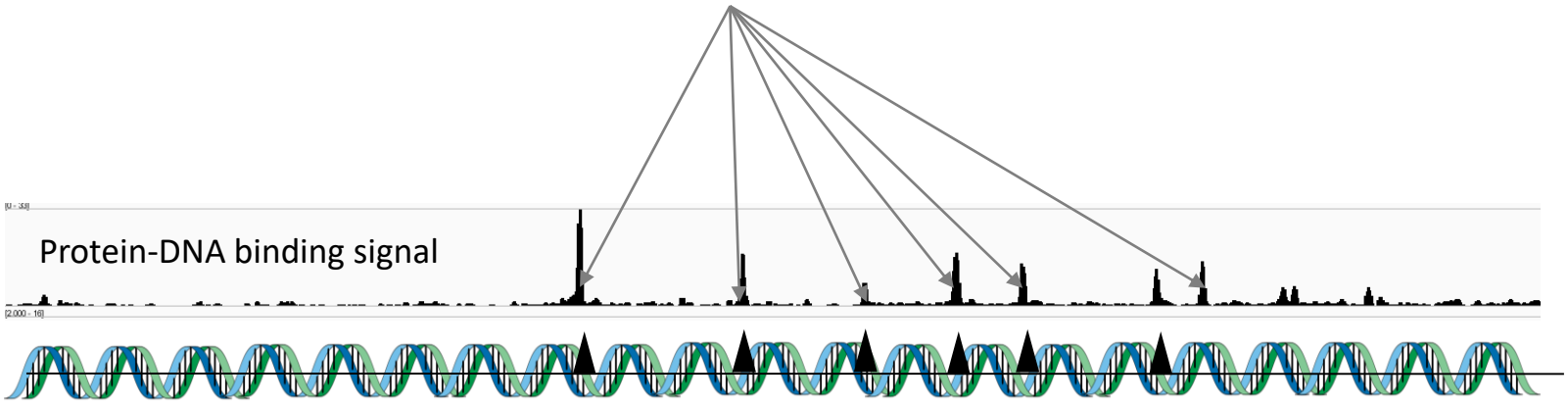
Pattern Discovery



Training data: 1000s of ‘positive examples’ (sequences) bound by protein

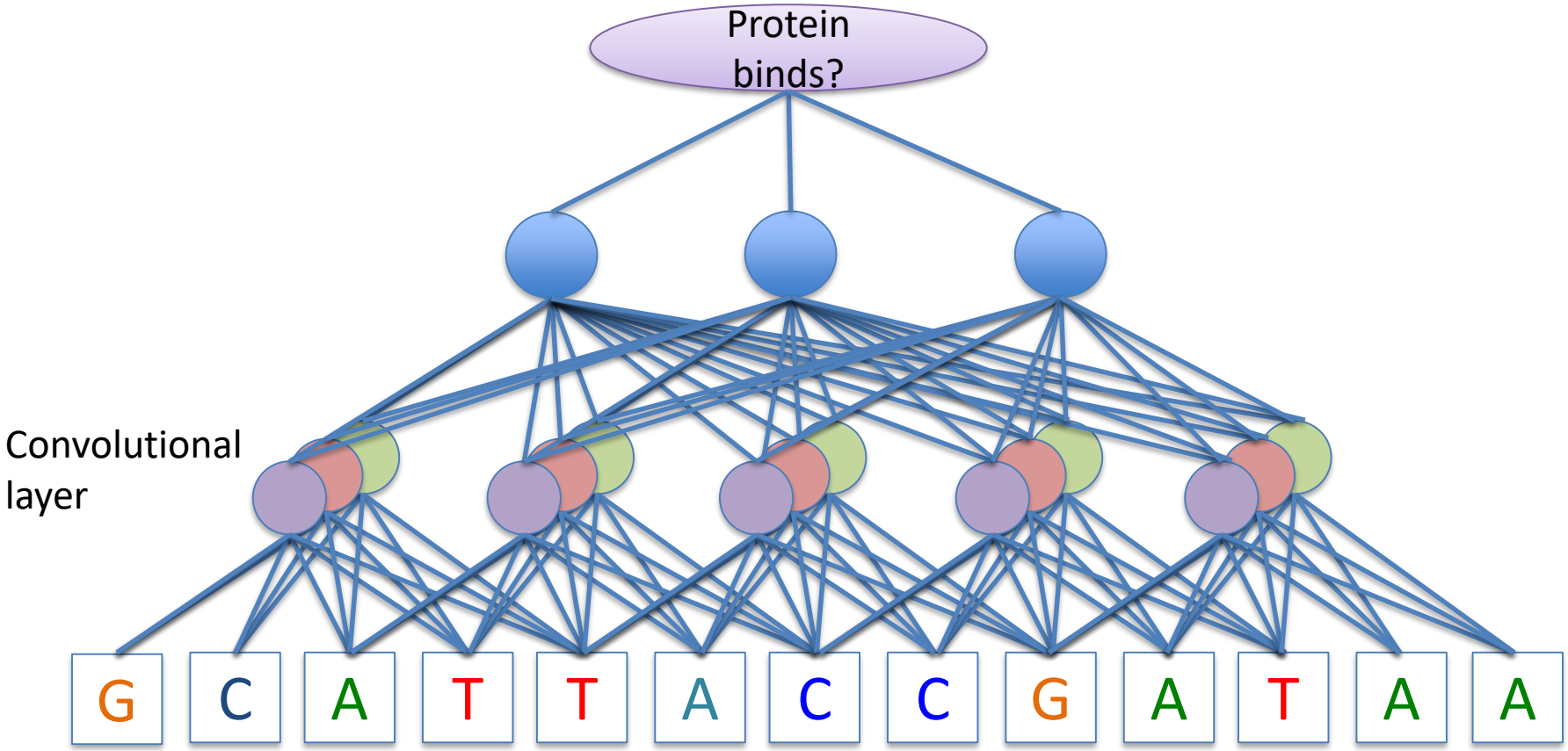
vs.

millions of ‘negative examples’ i.e. rest of the genome



Convolutional neural networks to predict binding of proteins to DNA sequence

Output: Bound (+1) vs. not bound (0)



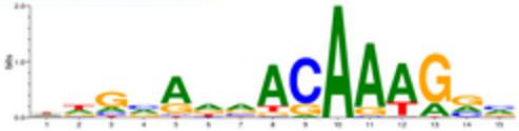
Input: Raw DNA sequence, one-hot encoded

Learning reoccurring patterns

ENCODE PWMs
(Kheradpour et al.)

NANOG_disc1:

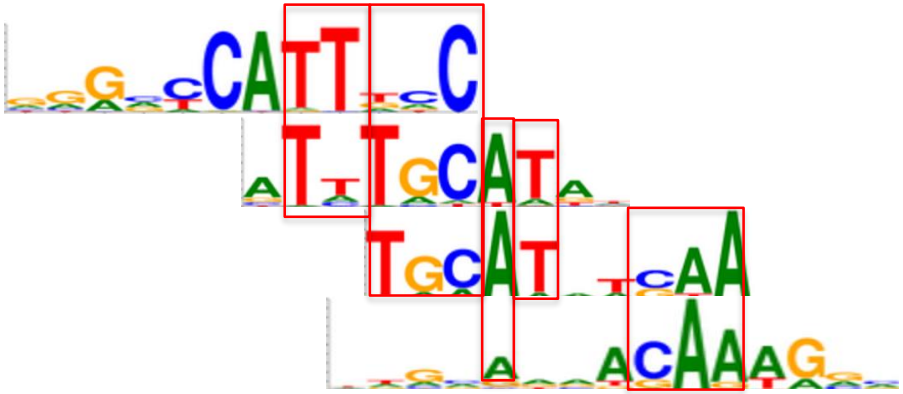
NANOG_disc2:



NANOG_disc3:

NANOG_disc4:

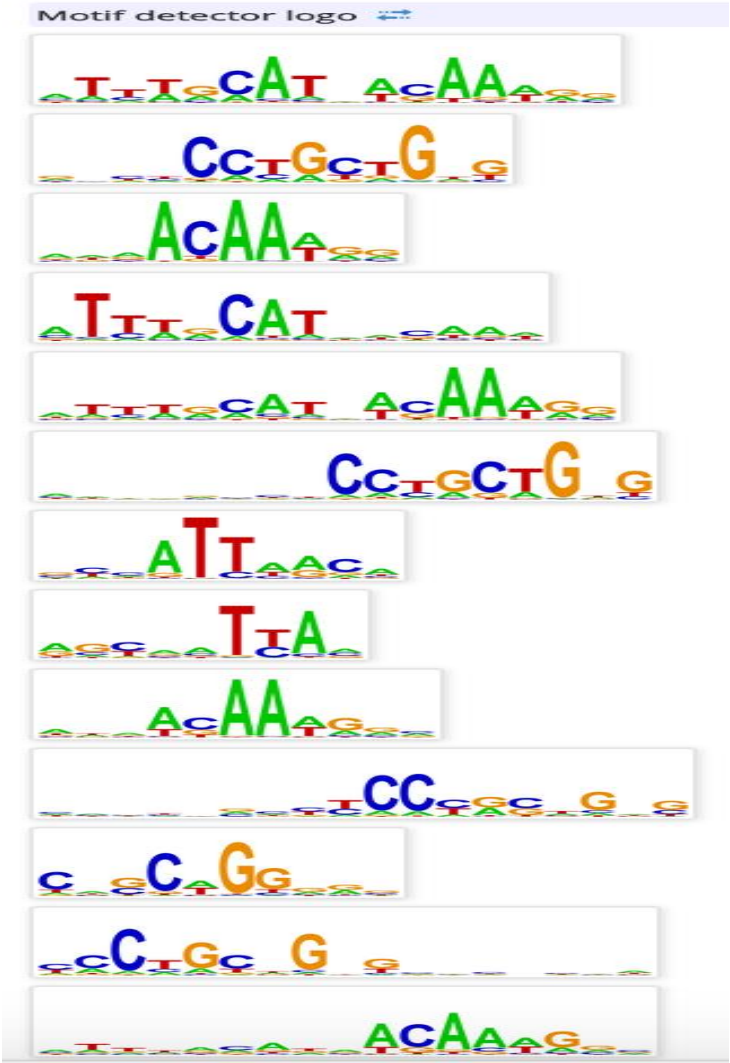
NANOG_known1:



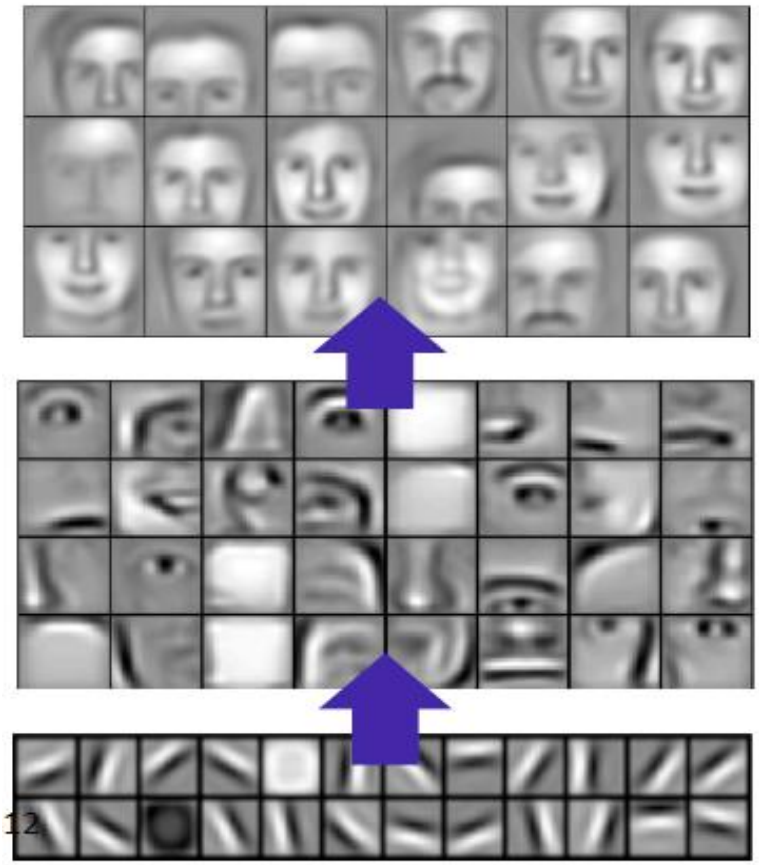
(Spoiler)
Result of MoDISco

Visualizing individual convolutional neurons gives highly distributed representations

DeepBind (Alipanahi et al.) for Nanog

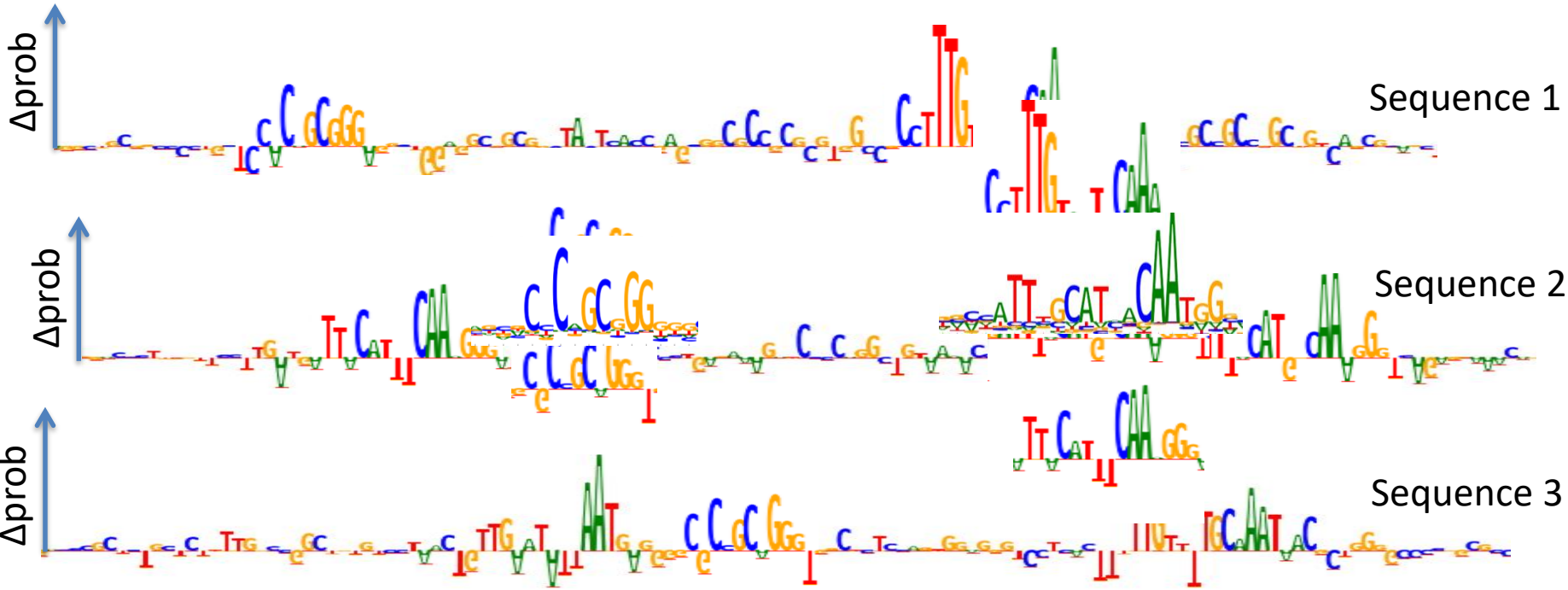


In computer vision



How do we combine the aggregate contribution of multiple pattern detectors at individual sequences?

Insight: filter contributions are resolved at the input level



MoDISco: Motif Discovery from Importance Scores

MoDISco uncovers broader patterns than existing approaches

MoDISco

Known pattern
(Heinz et al.)

Oct-Sox-Nanog

MoDISco

Known pattern
(Heinz et al.)

Nanog

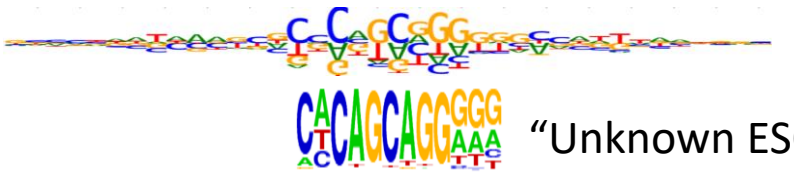
MoDISco

Known pattern
(Heinz et al.)

Sox2

MoDISco

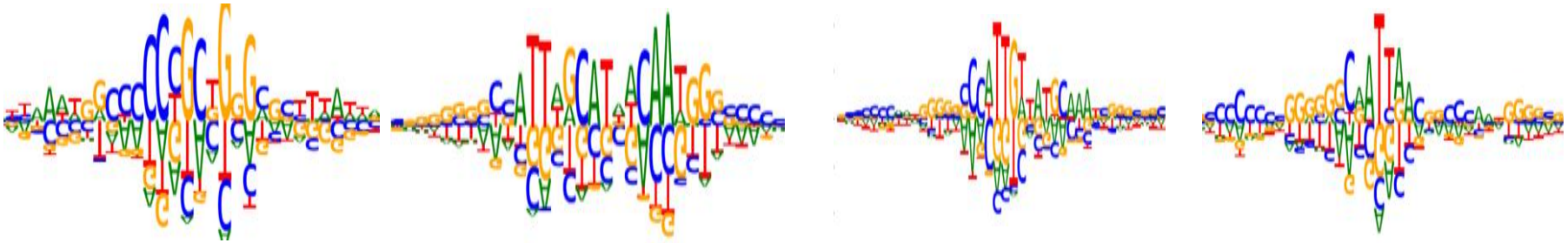
Known pattern
(Heinz et al.)



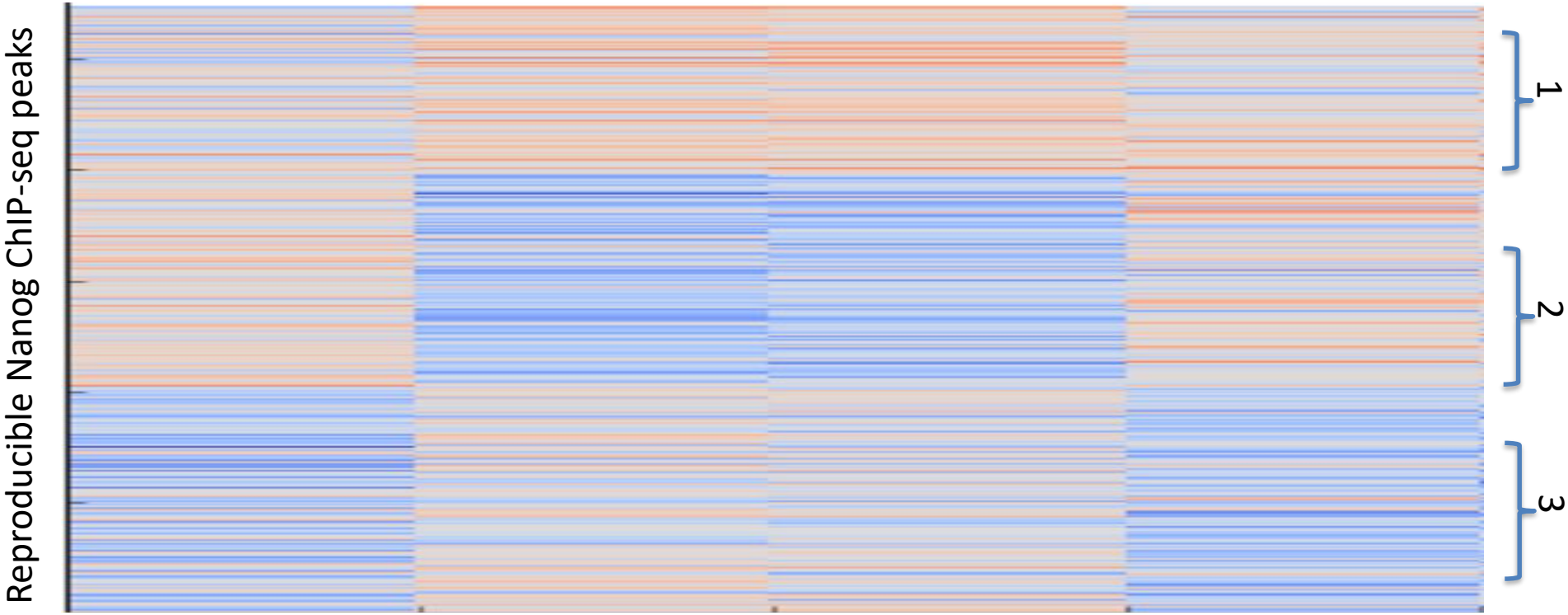
Zic3 (Jaspar; Zhao et al.)

"Unknown ESC element"

At least 3 distinct classes of Nanog sites

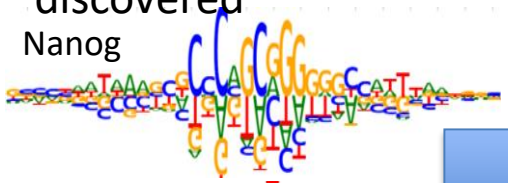


Zic3 motif | Oct4-Sox2-Nanog fusion motif | Sox2 motif | Nanog motif

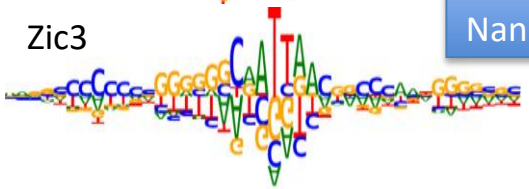


Potential novel relationship between Nanog and Zic3 discovered

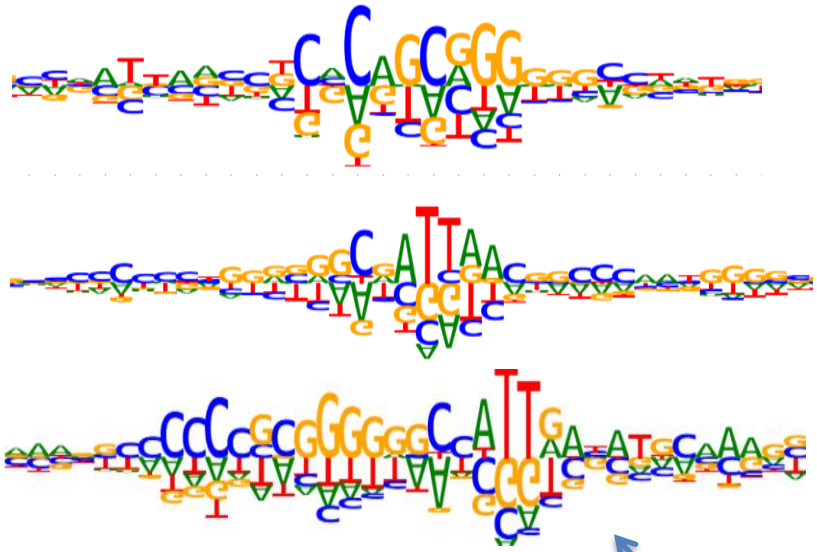
Nanog



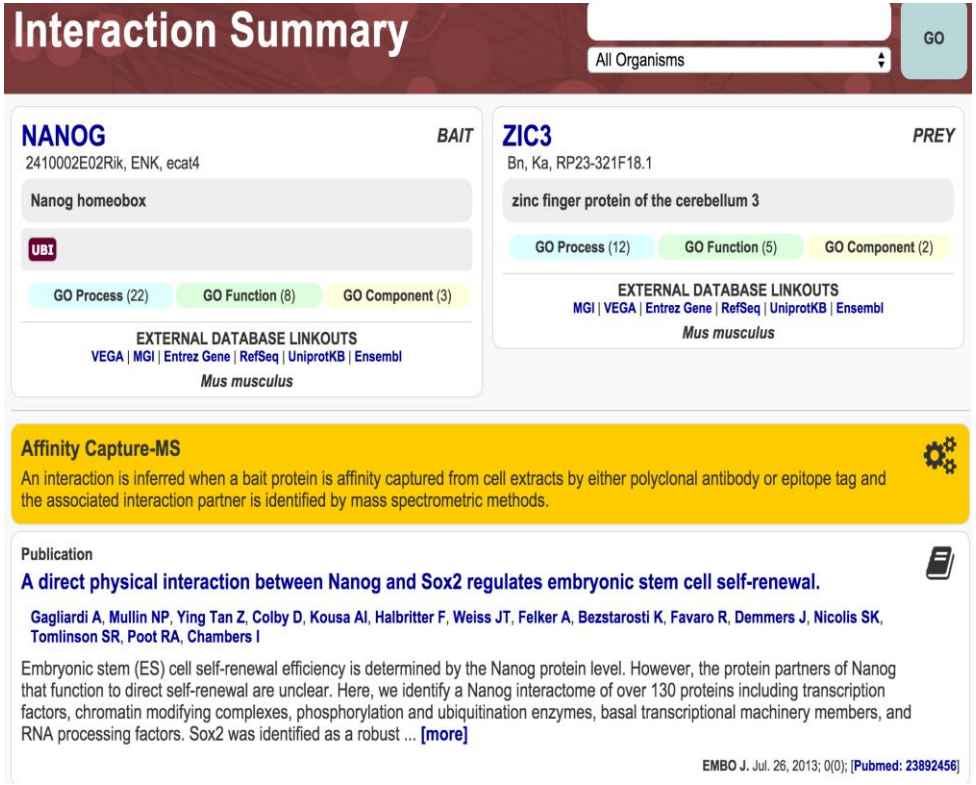
Zic3



Re-cluster with Nanog/Zic3 patterns only



Novel combination of Nanog and Zic3 patterns



Interaction Summary

All Organisms

GO

NANOG *BAIT*

2410002E02Rik, ENK, ecat4

Nanog homeobox

UBI

GO Process (22) | GO Function (8) | GO Component (3)

EXTERNAL DATABASE LINKOUTS

VEGA | MGI | Entrez Gene | RefSeq | UniprotKB | Ensembl

Mus musculus

ZIC3 *PREY*

Bn, Ka, RP23-321F18.1

zinc finger protein of the cerebellum 3

GO Process (12) | GO Function (5) | GO Component (2)

EXTERNAL DATABASE LINKOUTS

MGI | VEGA | Entrez Gene | RefSeq | UniprotKB | Ensembl

Mus musculus

Affinity Capture-MS

An interaction is inferred when a bait protein is affinity captured from cell extracts by either polyclonal antibody or epitope tag and the associated interaction partner is identified by mass spectrometric methods.

Publication

A direct physical interaction between Nanog and Sox2 regulates embryonic stem cell self-renewal.

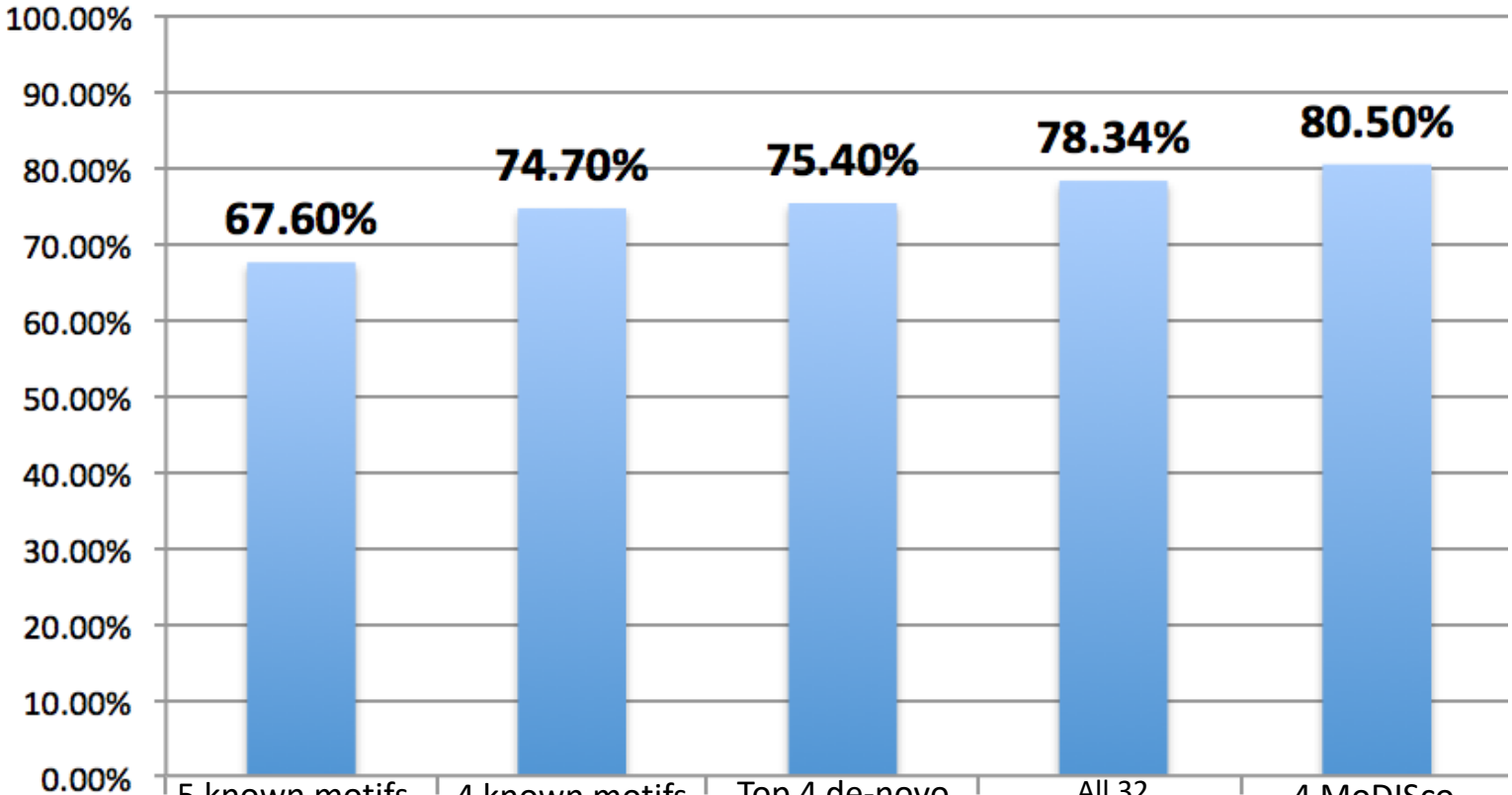
Gagliardi A, Mullin NP, Ying Tan Z, Colby D, Kousa AI, Halbritter F, Weiss JT, Felker A, Bezstarosti K, Favaro R, Demmers J, Nicolis SK, Tomlinson SR, Poot RA, Chambers I

Embryonic stem (ES) cell self-renewal efficiency is determined by the Nanog protein level. However, the protein partners of Nanog that function to direct self-renewal are unclear. Here, we identify a Nanog interactome of over 130 proteins including transcription factors, chromatin modifying complexes, phosphorylation and ubiquitination enzymes, basal transcriptional machinery members, and RNA processing factors. Sox2 was identified as a robust ... [more]

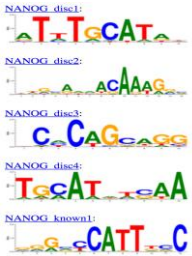
EMBO J. Jul. 26, 2013; 0(0); [Pubmed: 23892456]

MoDISco patterns outperform traditional motif discovery methods

Logistic regression on top hits to each pattern, auROC

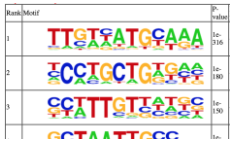


5 known motifs (ENCODE db)



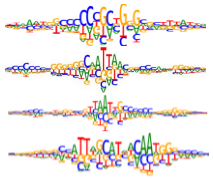
4 known motifs (HOMER db)

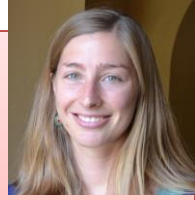
Top 4 de-novo from traditional method (HOMER)



All 32 de-novo from traditional method (HOMER)

4 MoDISco Motifs discovered on validation set





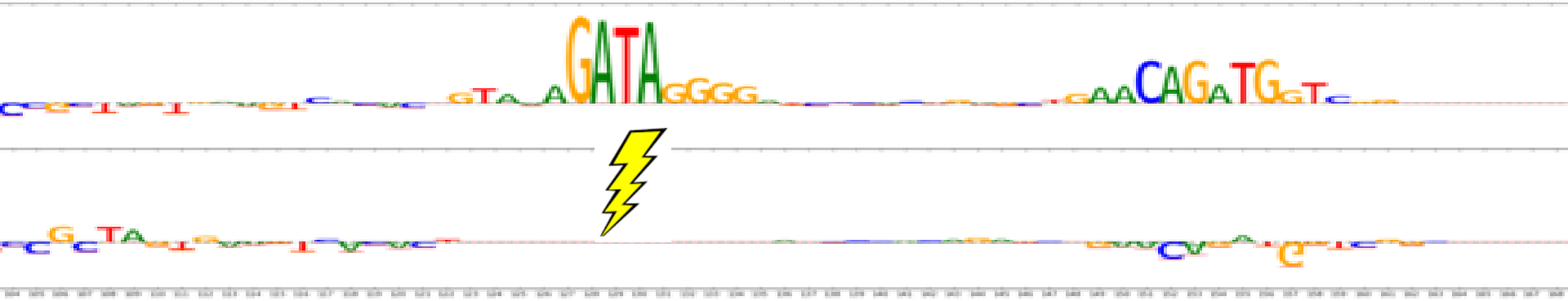
In development:
Discover dependencies with “Delta DeepLIFT”

Simulation: random background sequence with 0-2 0-2

Positive set: at least one and at least one

Identify sequences with one and one

Mutate the



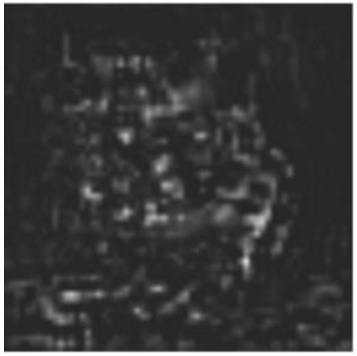
Caveats/Limitations

- You need a good reference!

Original

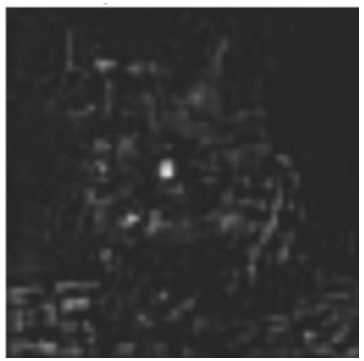
Taylor

DeepLIFT – flat ref



Ref =
Activations @
input of all zeros

DeepLIFT – learned ref



Ref =
Mean activations
(pre nonlinearity)

- Confidence scores
  - Ideas: test-time dropout, simulations

Summary

Find importance scores:

- Perturbation-based methods can find importance scores, but are inefficient and susceptible to saturation
- Backpropagation-based methods are efficient
- Examples of backpropagation methods include deconv nets, gradients, guided-backpropagation and DeepLIFT
- DeepLIFT works by looking at difference-from-reference, which addresses limitations of other methods
- However, DeepLIFT relies on having a good reference

Find patterns:

- Visualizing individual filters gives redundant representations
- MoDISco (Motif Discovery From Importance Scores): cluster and aggregate importance scores to find reoccurring patterns
- Can be extended to identify dependencies between features

Backup slides

Motif discovery works on continuous signals

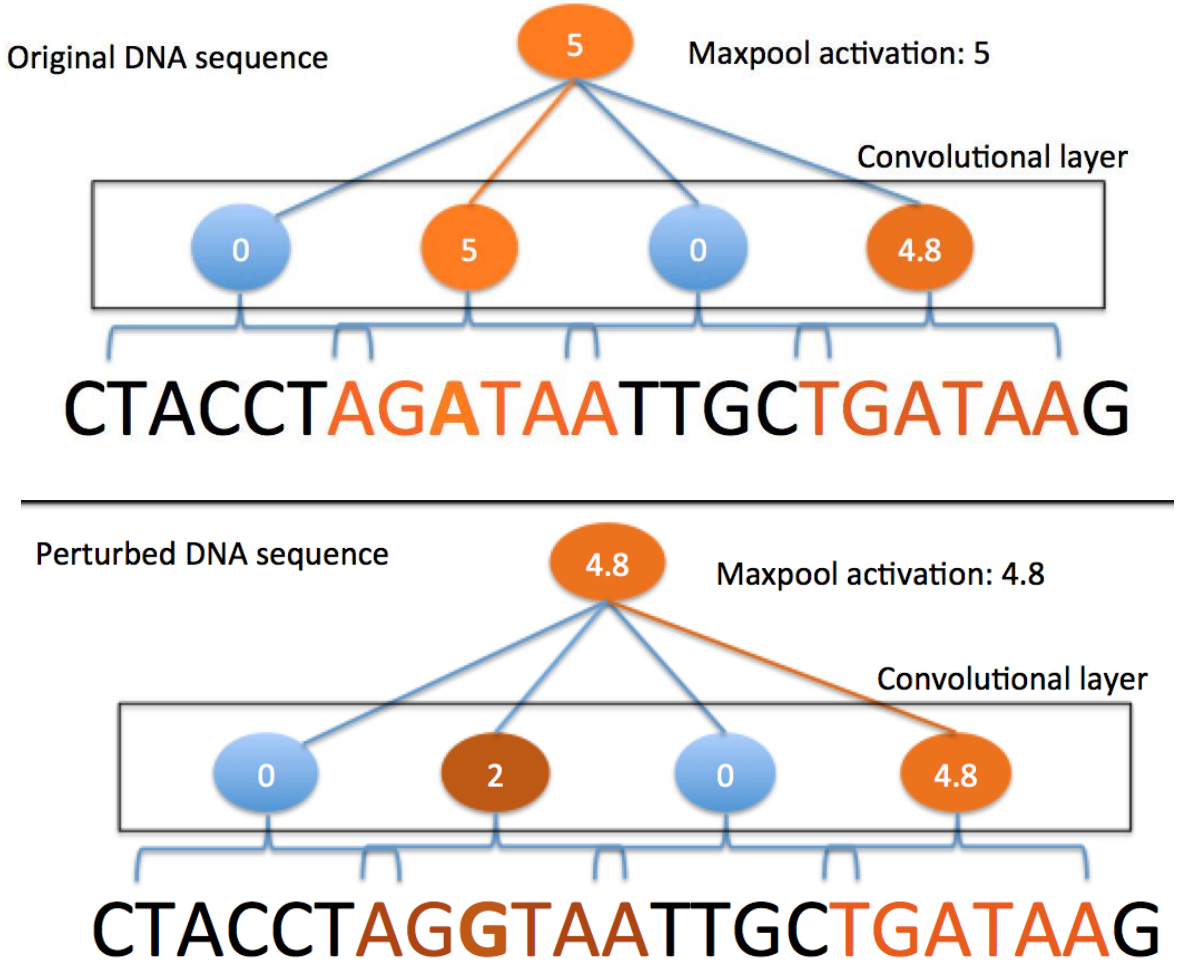
Model trained to predict binding of CTCF protein from sequence + accessibility signal (DNase)

DNA sequence pattern

"Footprint" pattern in accessibility signal from experimental assay (DNase)

(Backup slide)

Buffering and in-silico mutagenesis



(Backup)

Convolutional/Maxpooling Layers

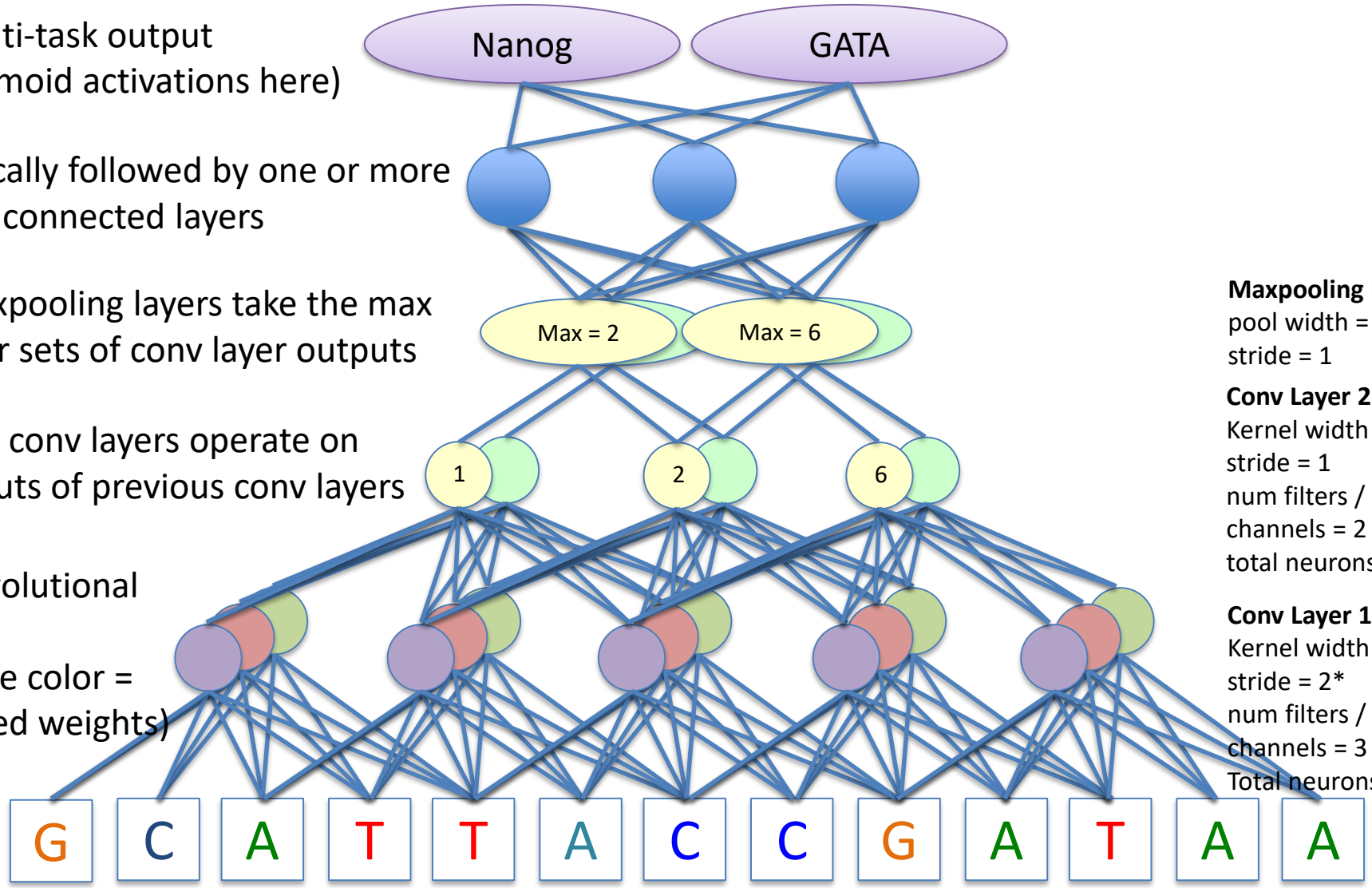
Multi-task output
(sigmoid activations here)

Typically followed by one or more fully connected layers

Maxpooling layers take the max over sets of conv layer outputs

Later conv layers operate on outputs of previous conv layers

Convolutional layer
(same color = shared weights)



Maxpooling layer
pool width = 2
stride = 1

Conv Layer 2
Kernel width = 3
stride = 1
num filters / num channels = 2
total neurons = 6

Conv Layer 1
Kernel width = 5
stride = 2*
num filters / num channels = 3
Total neurons = 15

Input: One-hot encoded raw DNA sequence (to be discussed)

***for genomics, a stride of 1 for conv layers is recommended**

(Backup slide)

Filter nullification

- Pattern-detector knockout
  - Convolutional layer has pattern detectors, called “filters”
  - Basset:
    - “We nullified each filter by setting its output to the mean value, which obstructs it from passing any information forward through the network. We computed the L2 norm of the change in predicted accessibility to measure its influence.”
    - Conceptually similar to “mutagenesis on the filters”
    - Computationally inefficient!
    - From the Basset tutorial:

This analysis requires considerably more compute time, so I separated it into a different script. To give it a sufficient number of sequences to obtain a good estimate influence, I typically run it overnight. If your computer is using too much memory, decrease the batch size. I'm going to run here with 1,000 sequences, but I used 20,000 for the paper.

    - Buffering still a problem

(Backup slide)

Zic3 Is Required for Maintenance of Pluripotency in Embryonic Stem Cells

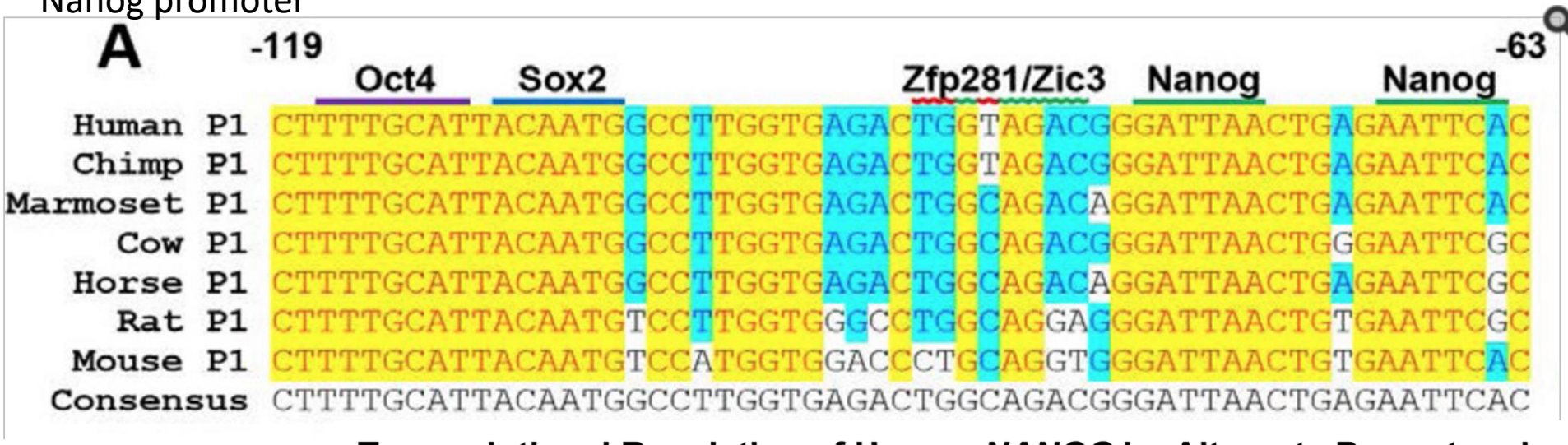
Linda Shushan Lim,[*†] Yuin-Han Loh,[†‡] Weiwei Zhang,[‡§] Yixun Li,[‖] Xi Chen,[‡§] Yinan Wang,[‡§] Manjiri Bakre,[*] Huck-Hui Ng,[‡§] and Lawrence W. Stanton[*§]

Stem Cells. 2010 Nov;28(11):1961-9. doi: 10.1002/stem.527.

The pluripotency regulator Zic3 is a direct activator of the Nanog promoter in ESCs.

Lim LS[1], Hong FH, Kunarso G, Stanton LW.

Nanog promoter



Transcriptional Regulation of Human *NANOG* by Alternate Promoters in Embryonic Stem Cells

Satyabrata Das,[1] Snehalata Jena,[1] Eun-Mi Kim,[1] Nicholas Zavazava,[1] and Dana N. Levasseur[1,2,*]