MRC Molecular Haematology Unit

UNIVERSITY OF OXFORD

wellcome

Deep Neural Networks in Genomics

Ron Schwessinger

03/12/2019

MRC Weatherall Institute of Molecular Medicine

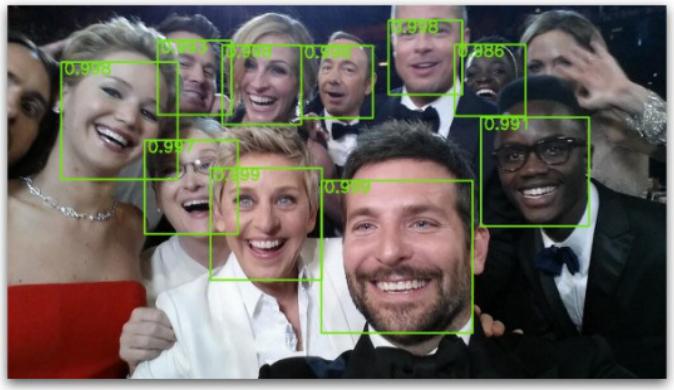
The MRC Weatherall Institute of Molecular Medicine is a strategic alliance between the Medical Research Council and the University of Oxford

Outline

1) Introduction to Deep Neural Networks
    1) Neural Networks oversimplified
    2) How do we learn?
    3) Training Process
2) Introduction to Convolutional Neural Networks
    1) Convolutions for Image Analysis
    2) Convolutions for DNA
3) Convolutional Neural Networks in Genomics
    1) Chromatin Feature Networks
    2) Utility, Strategies and Interpretation
    3) More Examples, More Architectures
4) Basic Practical Aspects
    1) Overfitting / Underfitting
    2) Train, Validation & Test Sets (& Cross-Validation)
    3) Learning Rate & Optimizers

1) Introduction to Deep Neural Networks

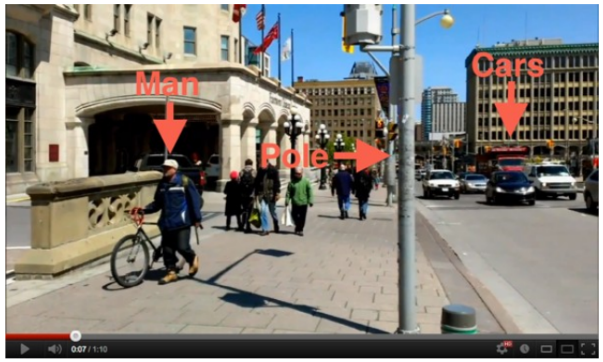
Deep Learning Out There



Google lands patent for automatic object recognition in videos, leaves no stone untagged

Jon Fingas , @jonfingas
08.28.12

0
Shares



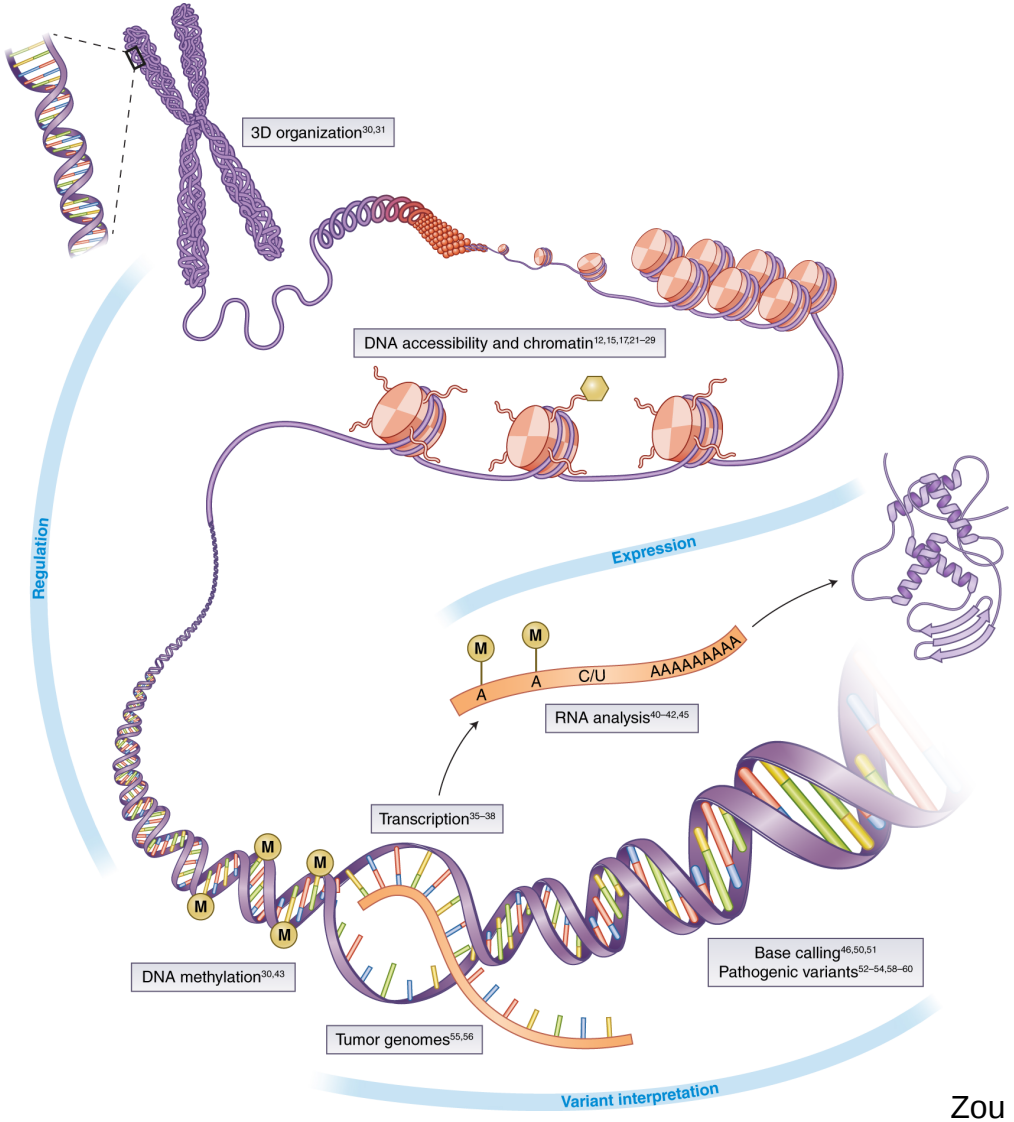
Sponsored Links by Taboola

Not All Mobile Phones Are Created Equal - Discover the OnePlus
OnePlus

The Gmail Trick That Google Doesn't Talk About
Boomerang

15 Most Powerful Email Subject Lines
Campaign Monitor

Deep Learning in Genomics



Zou *et al.* Nature Genetics 2019

Neural Networks oversimplified

rain probability from weather forecast

x

0.6
0.4
0.2
0.8

tomorrow is a rainy day

y

1
1
0
1

y truth: is tomorrow a rainy day

$\hat{y}$ prediction: is tomorrow a rainy day?

x local forecast rain prob.

Neural Networks oversimplified

rain probability from weather forecast

x

0.6
0.4
0.2
0.8

tomorrow is a rainy day

y

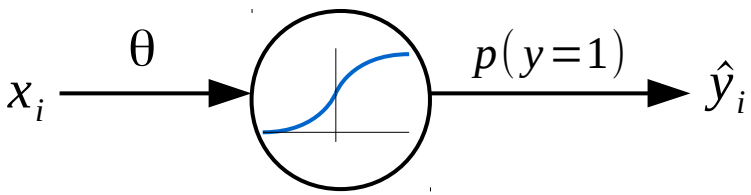
1
1
0
1

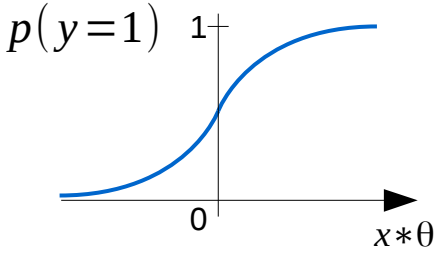
y truth: is tomorrow a rainy day

$\hat{y}$ prediction: is tomorrow a rainy day?

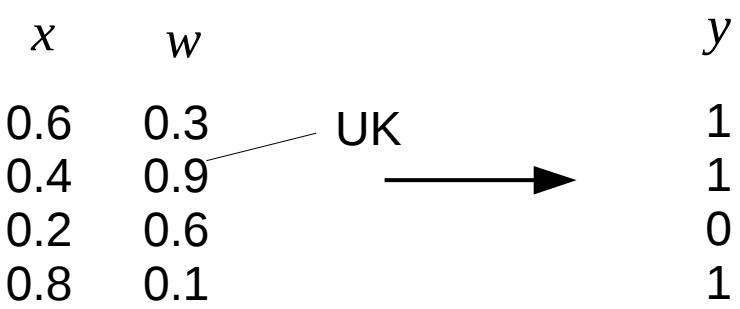
x local forecast rain prob.

Single Neuron





Neural Networks oversimplified



- y truth is tomorrow a rainy day
- $\hat{y}$ prediction: is tomorrow a rainy day?
- x local forecast rain prob.
- w how rainy is my current location in general?

Neural Networks oversimplified

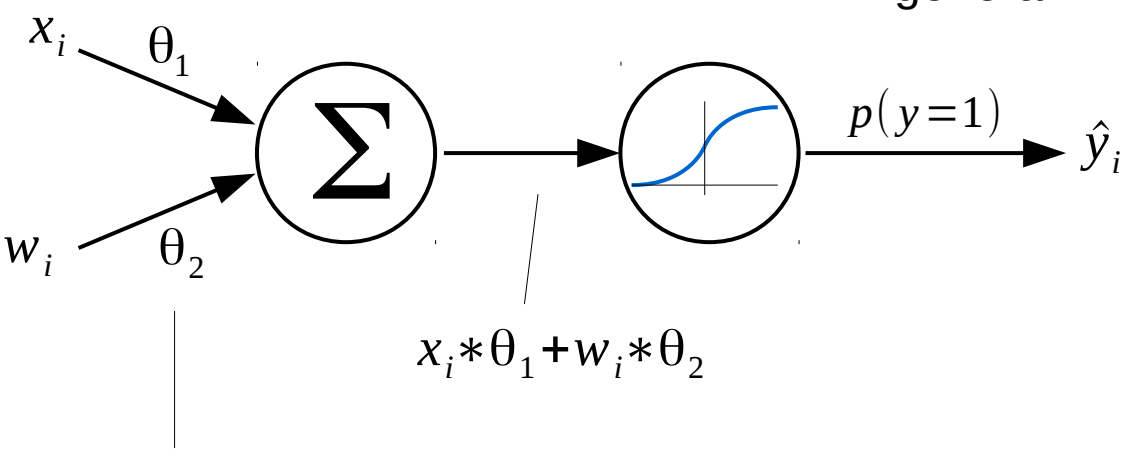
x	w		y
0.6	0.3	UK →	1
0.4	0.9		1
0.2	0.6		0
0.8	0.1		1

y truth is tomorrow a rainy day

$\hat{y}$ prediction: is tomorrow a rainy day?

x local forecast rain prob.

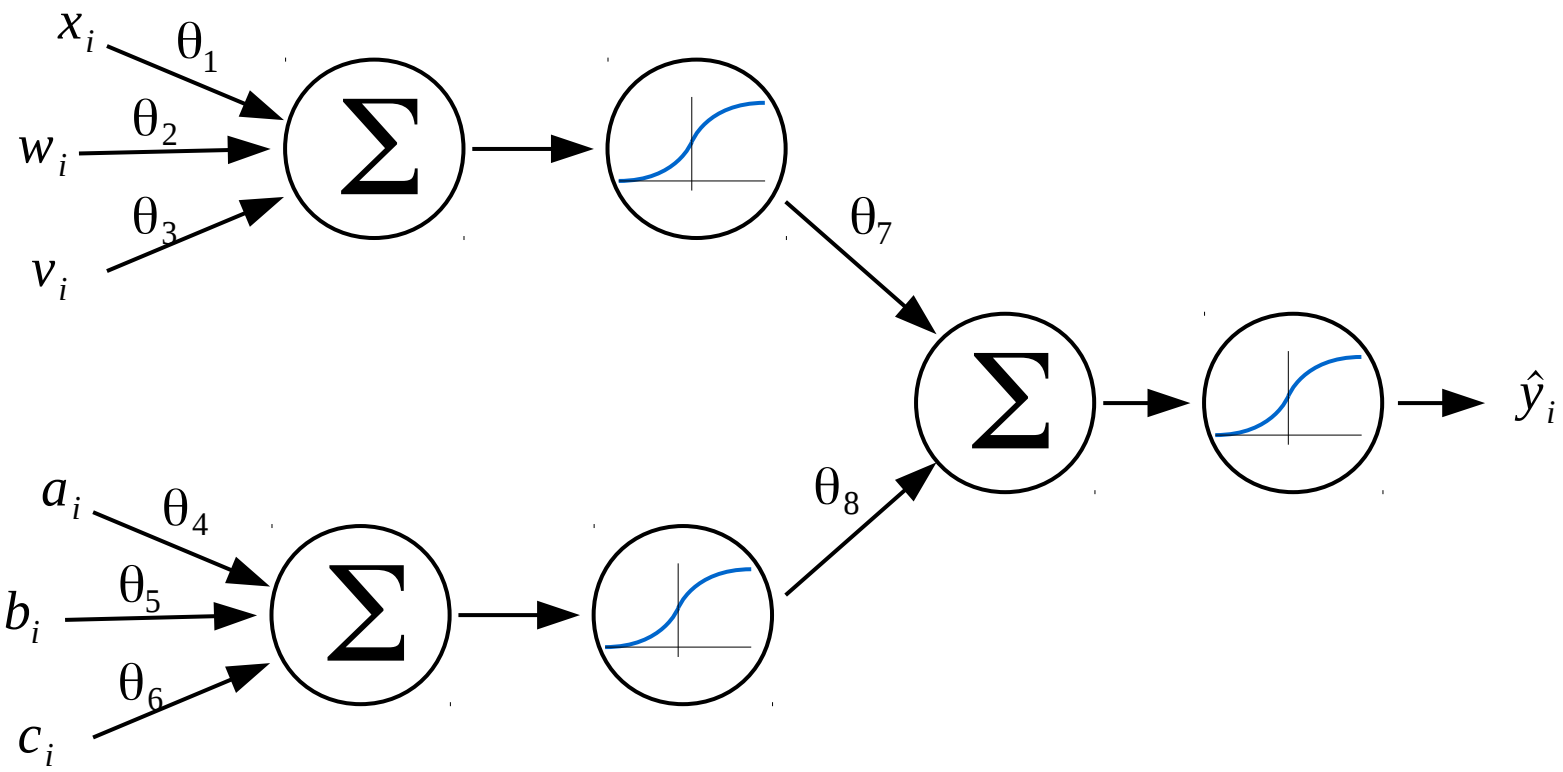
w how rainy is my current location in general?



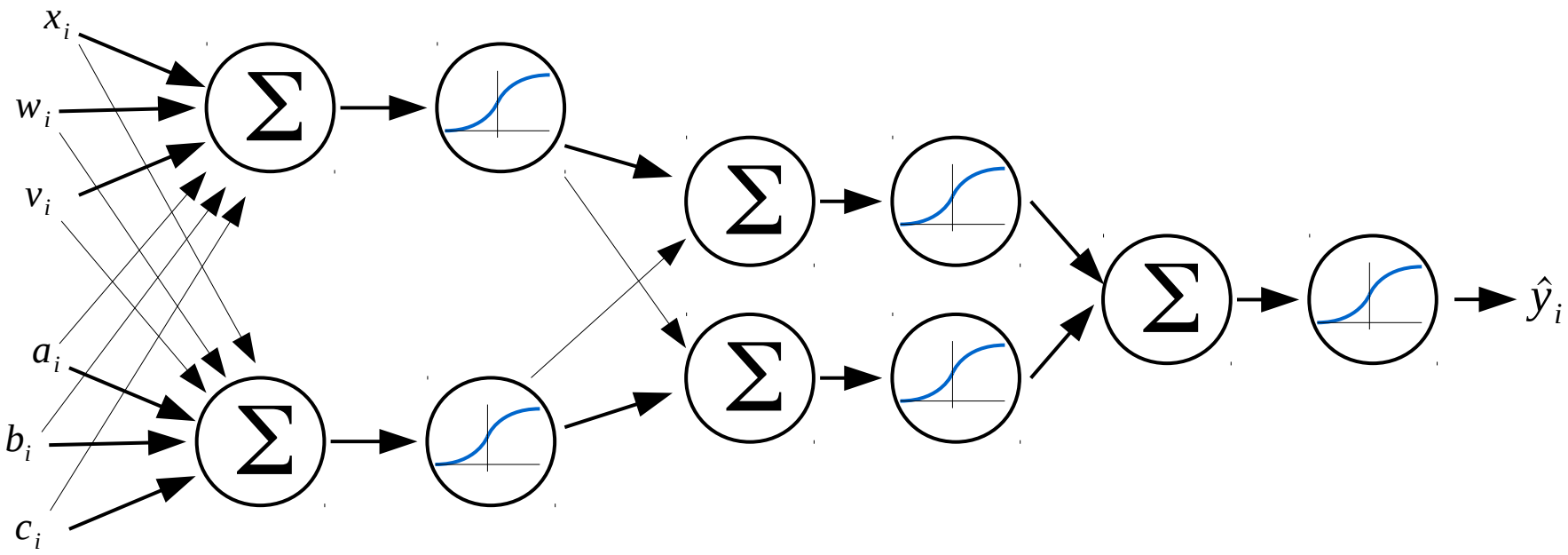
parameters

aim: tweak the parameters to get the best predictions

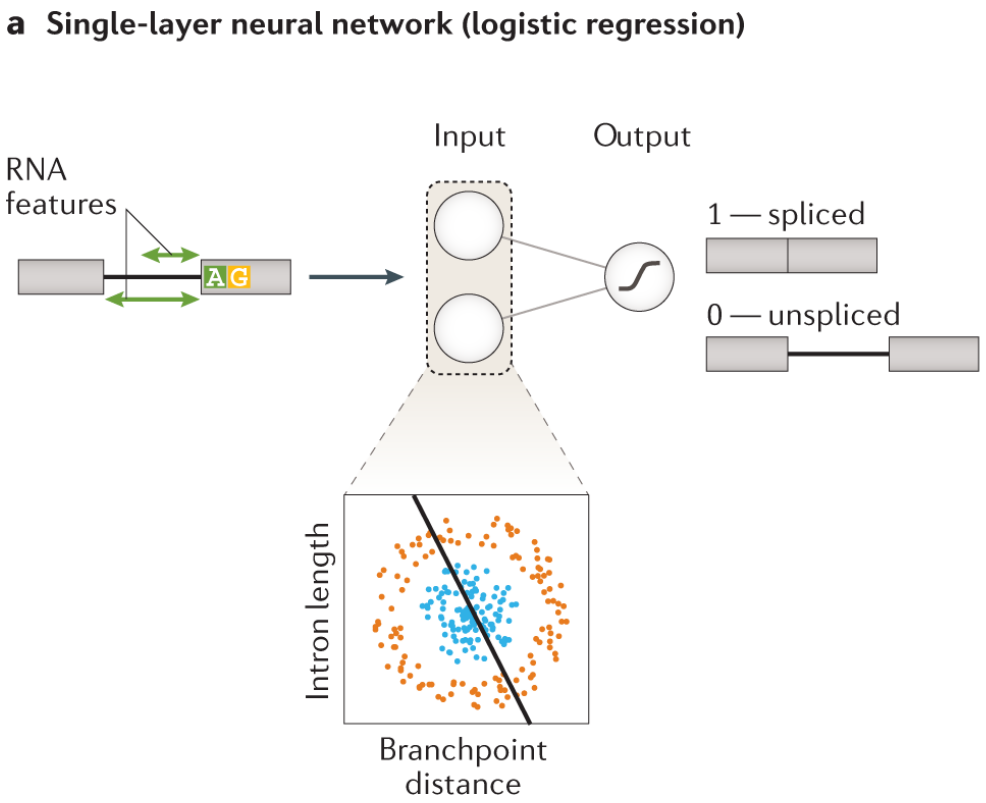
Neural Networks oversimplified

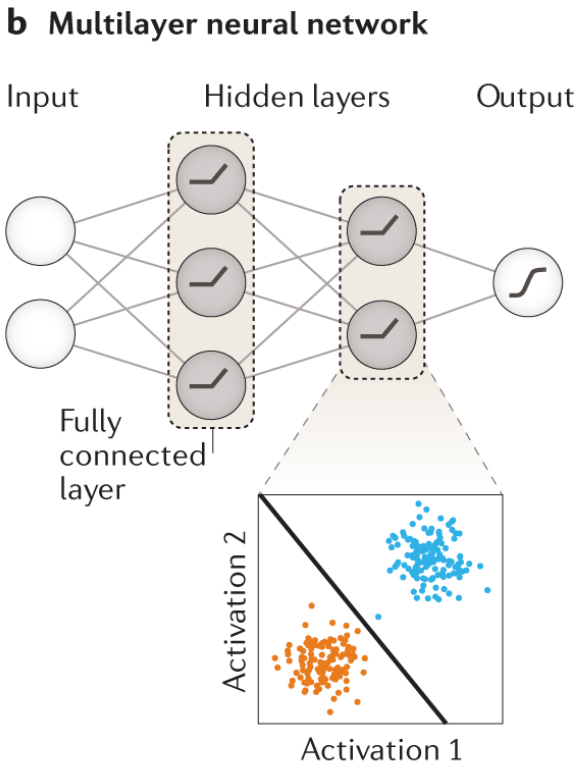


Neural Networks oversimplified



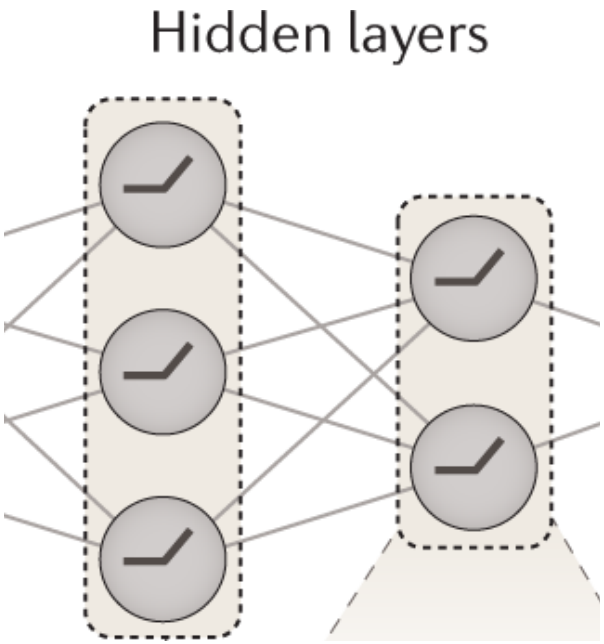
The Power of Non Linear Activations





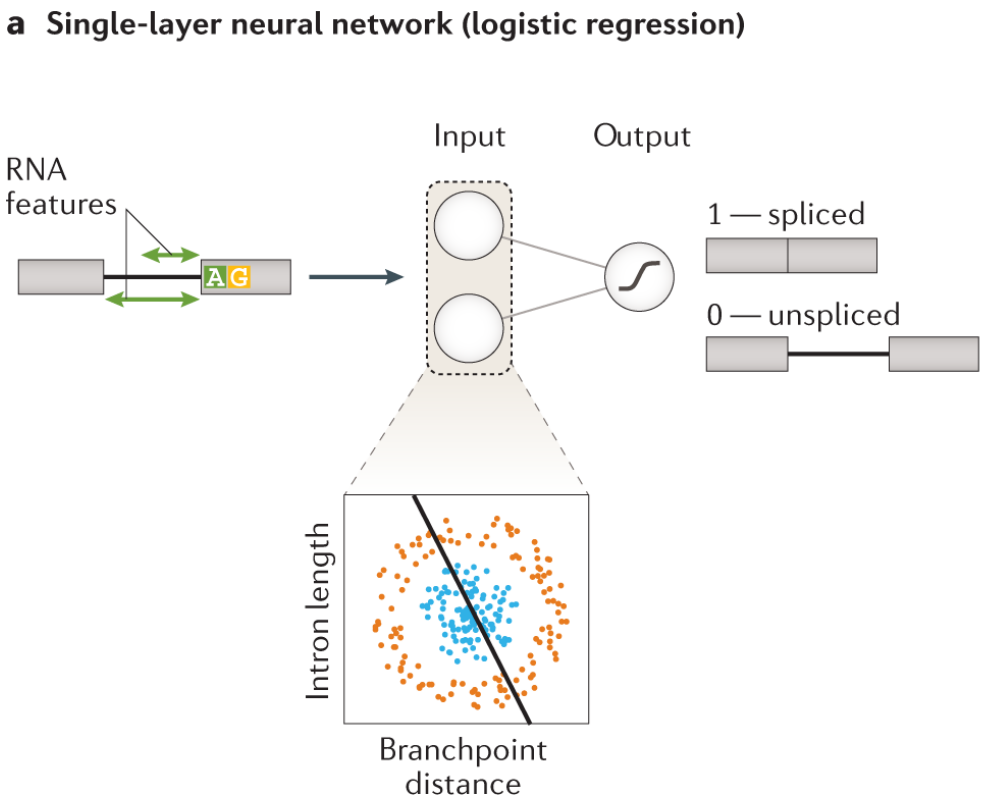
Gökcen *et al.* Nature Reviews Genetics 2019

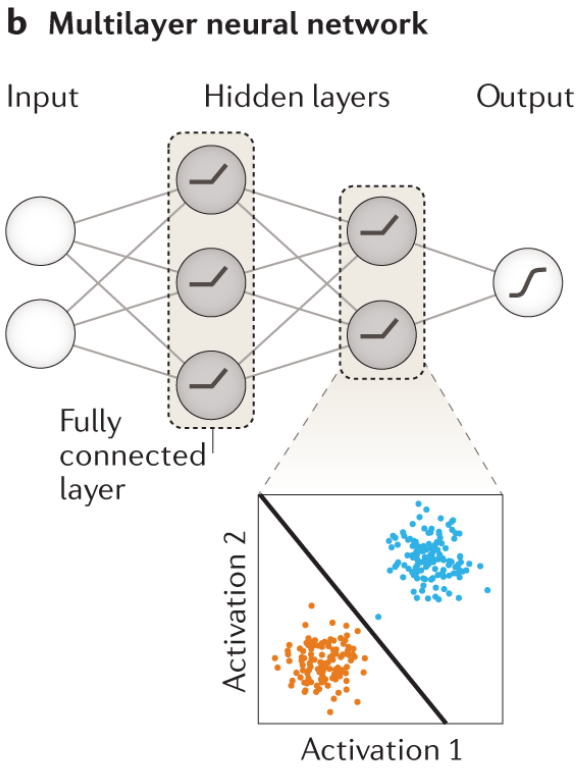
Non Linear Activations



Gökcen *et al.* Nature Reviews Genetics 2019

The Power of Non Linear Activations





Gökcen *et al.* Nature Reviews Genetics 2019

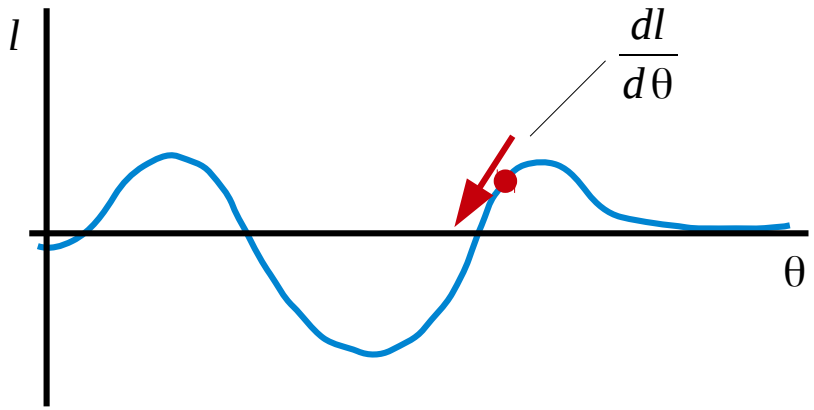
How do we learn?

Optimise parameters θ :

loss function l → how bad do we perfom? → minimize l

calculate **gradients!**

Optimization



How do we learn?

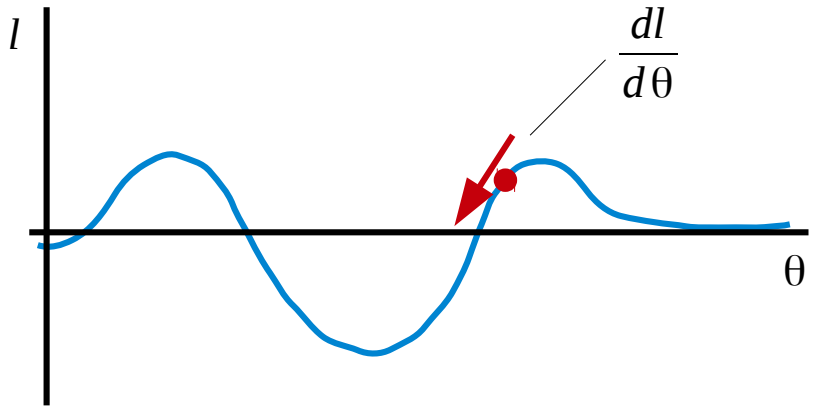
Optimise parameters θ :

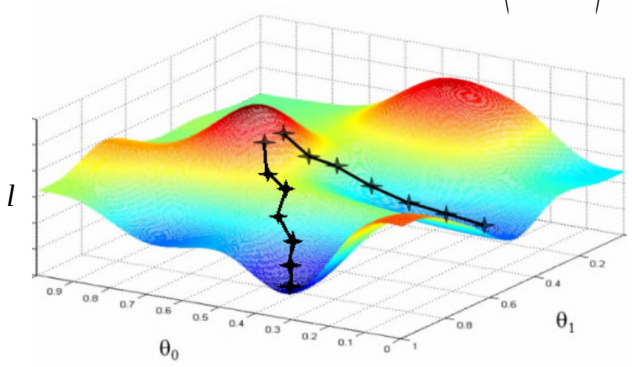
loss function l → how bad do we perfom? → minimize l

calculate **gradients!**

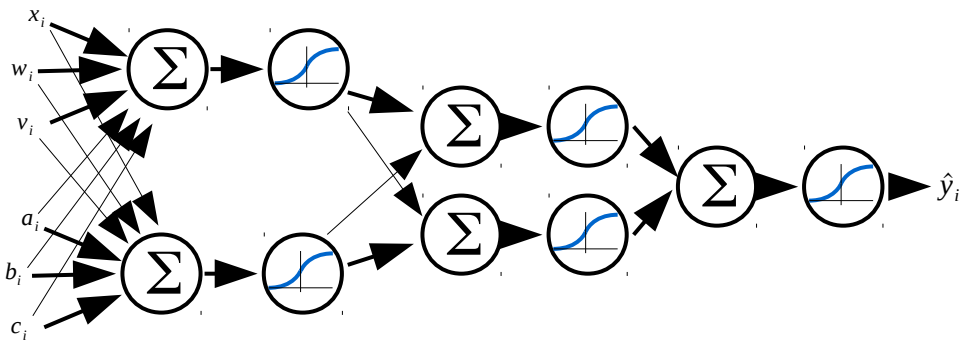
$$\nabla J = \begin{pmatrix} \frac{\partial l}{\partial \theta_0} \\ \frac{\partial l}{\partial \theta_1} \end{pmatrix}$$

Optimization





How do we learn?



- for simple NN → gradients can be calculated by hand
- for larger architectures by computers
- but for very large/complex NNs → normal CPU computation takes ages …

Two major developments:

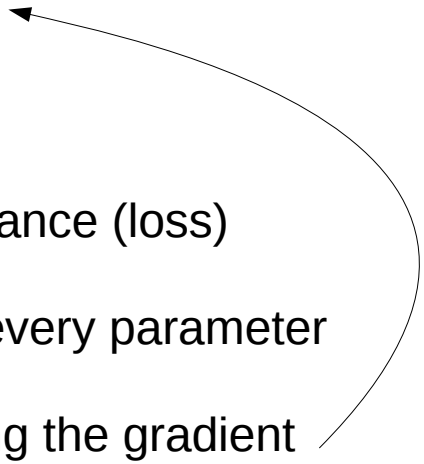
- better algorithms to calculate partial derivatives along large networks
- using graphics cards (GPU) computation

GPUs are great for massive parallelization of simply computational tasks

Training Process

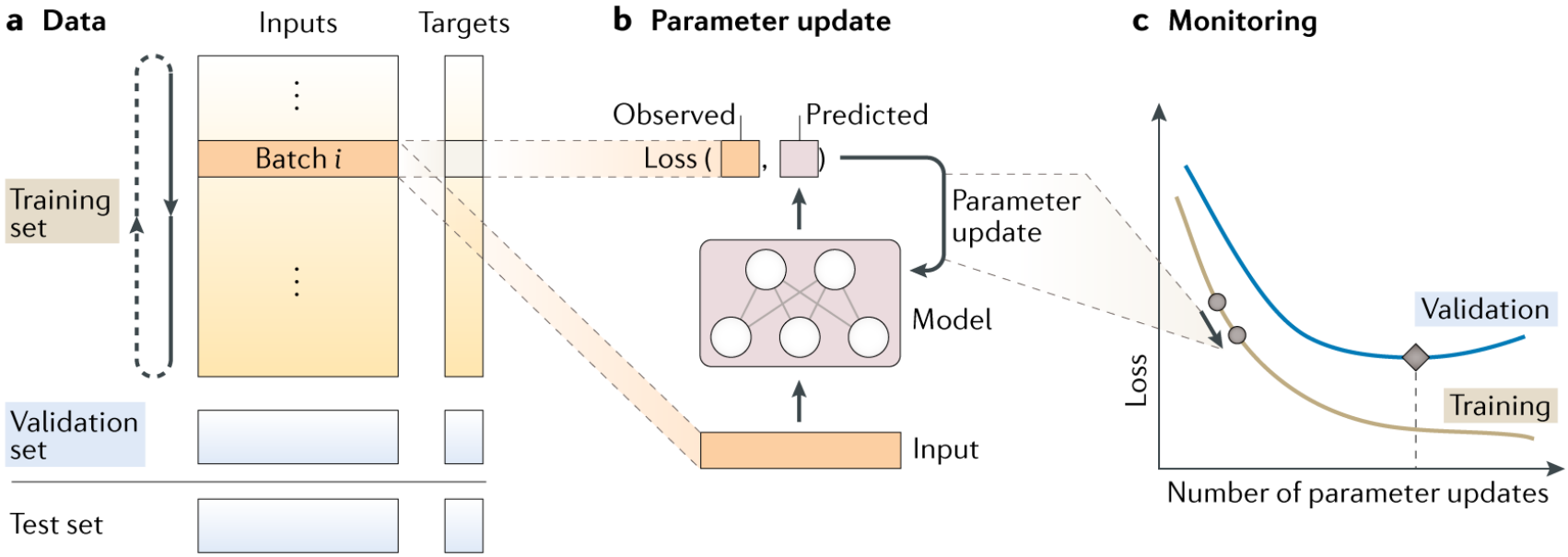
Training:

1) Start with random parameters
2) Take a batch of inputs
3) Calculate predictions
4) Calculate your performance (loss)
5) Calculate gradient for every parameter
6) Adjust parameters along the gradient



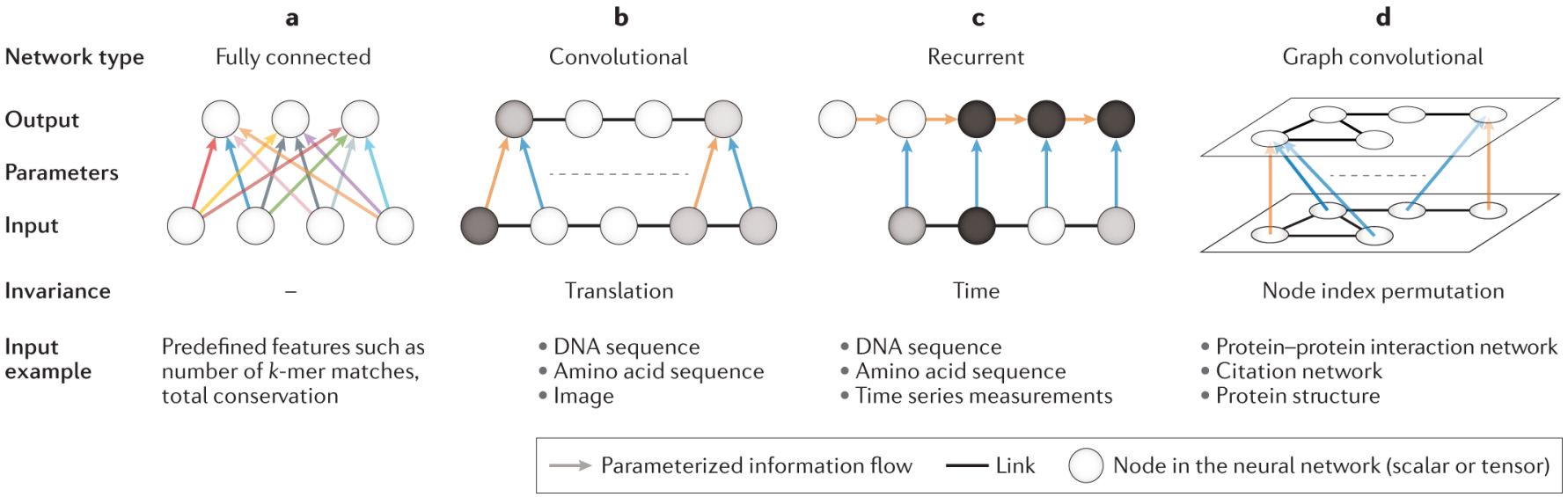
Until no improvement in performance noticable...

Training Process



Gökcen *et al.* Nature Reviews Genetics 2019

Overview of Network Types



Gökcen *et al.* Nature Reviews Genetics 2019

2) Introduction to Convolutional Neural Networks

Convolutions for Image Analysis

Image

Grey values per pixels

104	111	101	101
90	88	91	90
39	41	35	44
51	49	48	52

Convolutions for Image Analysis

Image

Grey values per pixels

104	111	101	101
90	88	91	90
39	41	35	44
51	49	48	52

192	202	191
131	123	135
88	89	87

Filter

$$\sum \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) = x$$

1*a + 0*b + 0*c + 1*d = x

Filter = $\begin{bmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{bmatrix}$ → parameters to optimize

Edge detection – Sobel operator

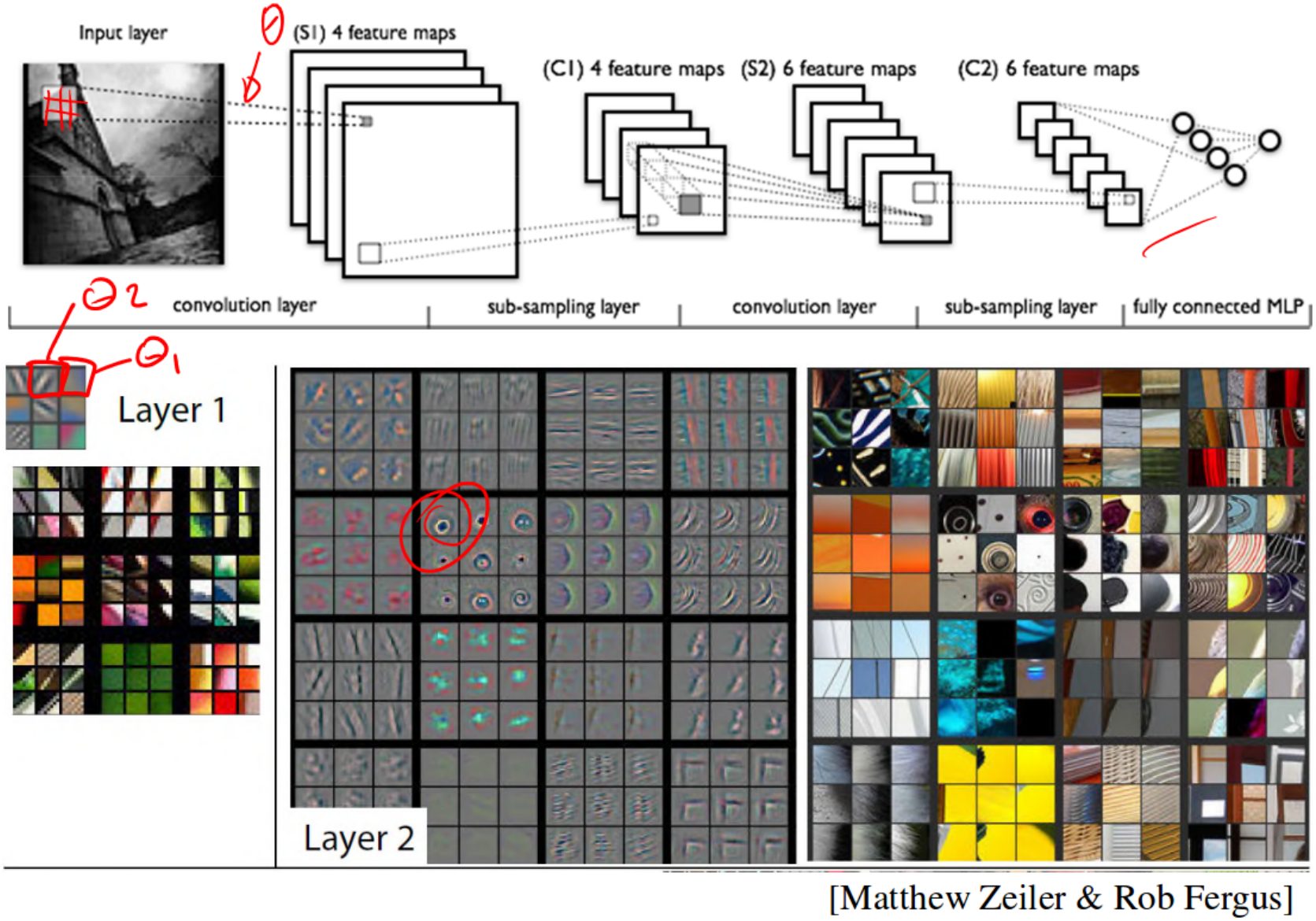
-1	0	1
-2	0	2
-1	0	1

→

1	2	1
0	0	0
-1	-2	-1

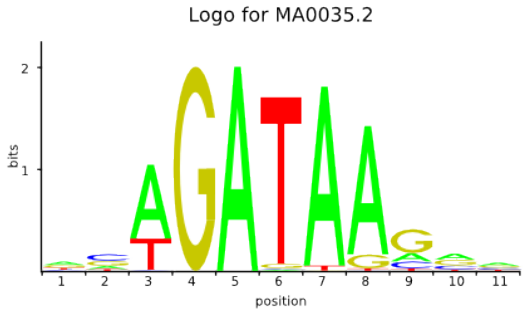
↑

Convolutional networks



[Matthew Zeiler & Rob Fergus]

Convolutions for DNA



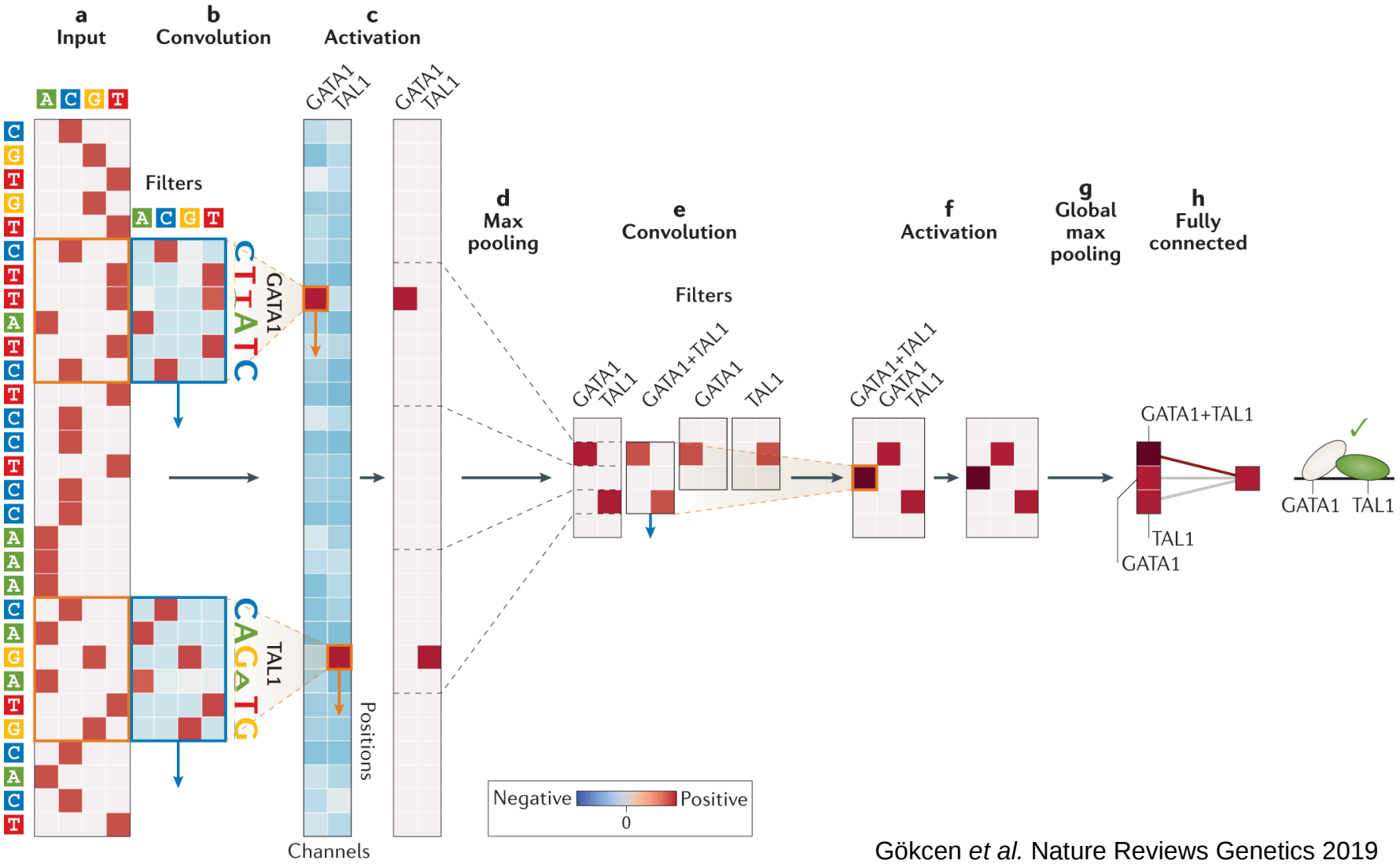

```
A  [1423  708 2782    0 4000   27 3887 3550  799 1432 1487 ]
C  [ 560 1633   31    0    0   29    0    4  681  897  829 ]
G  [1242 1235   10 4000    0  109    6  383 2296 1360 1099 ]
T  [ 775  424 1177    0    0 3835  107   63  224  311  585 ]
```

hot coded sequence

```
   A C A G A T A A G T A G A G G C T A T T C C ...
A  [1 0 1 0 1 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 ...
C  [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 ...
G  [0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 ...
T  [0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 ...
```

```
1005 506 501 501 40 55 501 ...
```

Convolutions of Convolutions for DNA

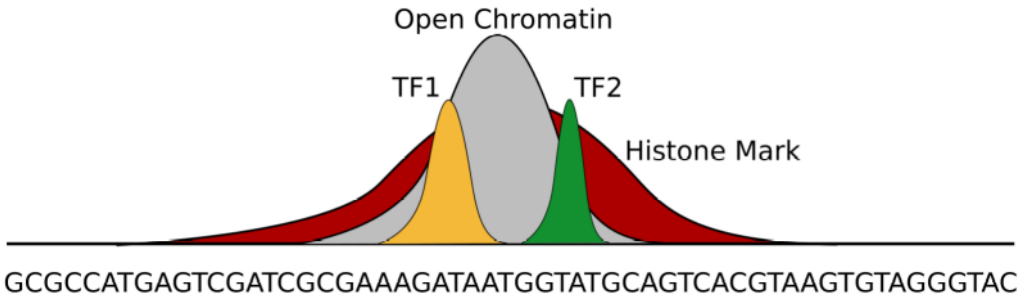


Gökcen *et al.* Nature Reviews Genetics 2019

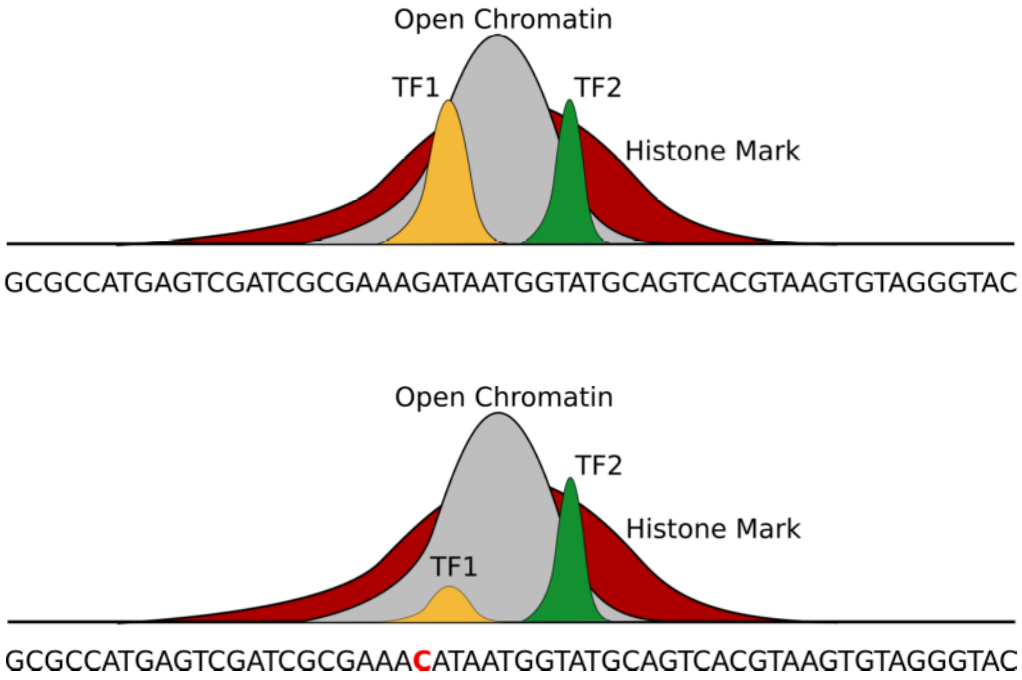
Coffee Break!

3) Convolutional Neural Networks in Genomics

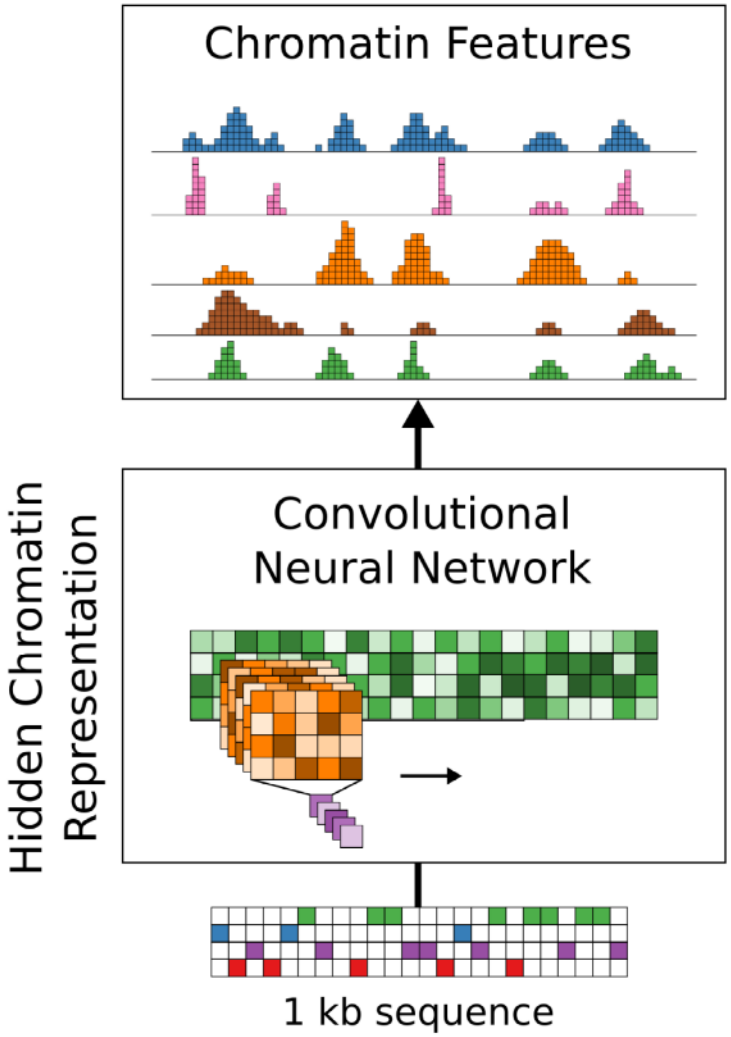
Chromatin Feature Networks



Chromatin Feature Networks



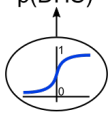
Chromatin Feature Networks



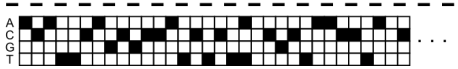
Chromatin Feature Networks

Output

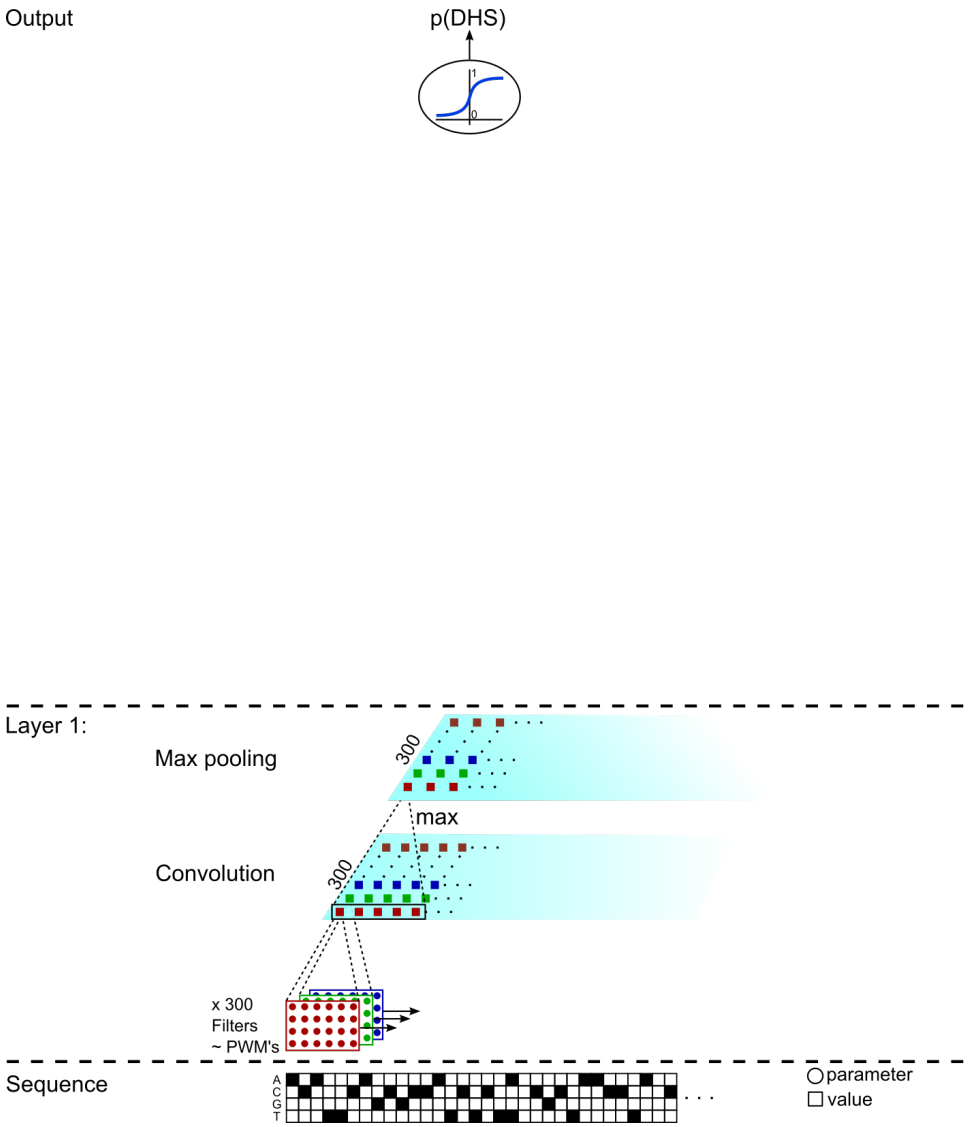
p(DHS)



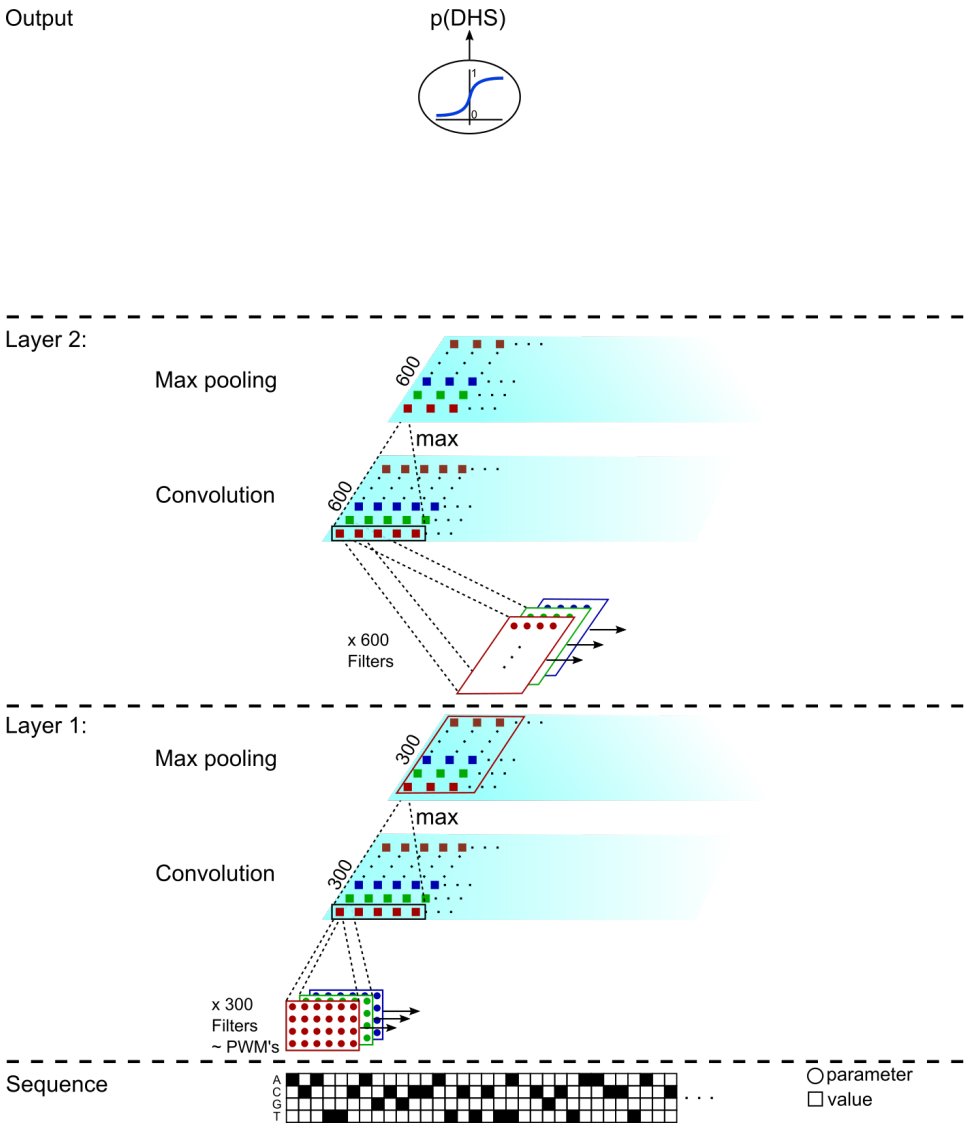
Sequence



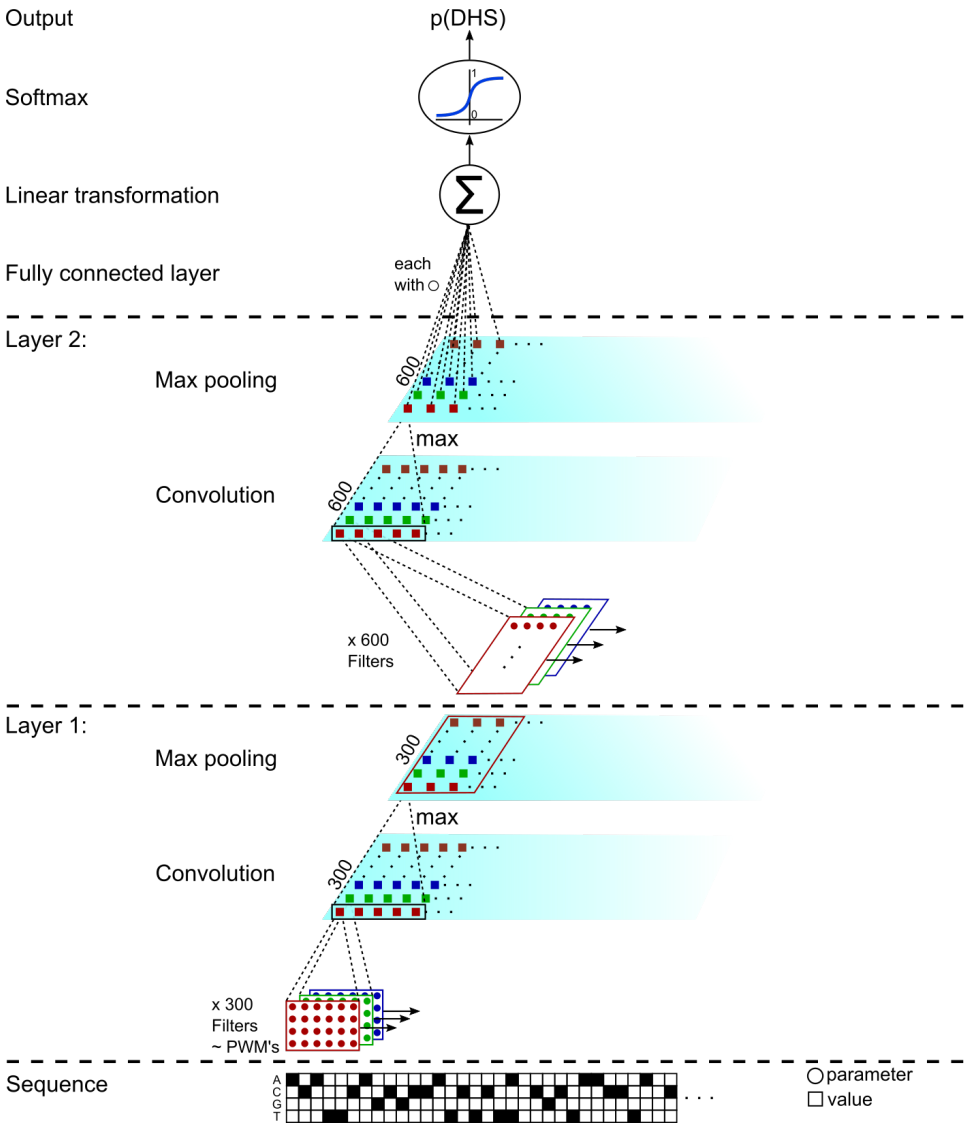
Chromatin Feature Networks



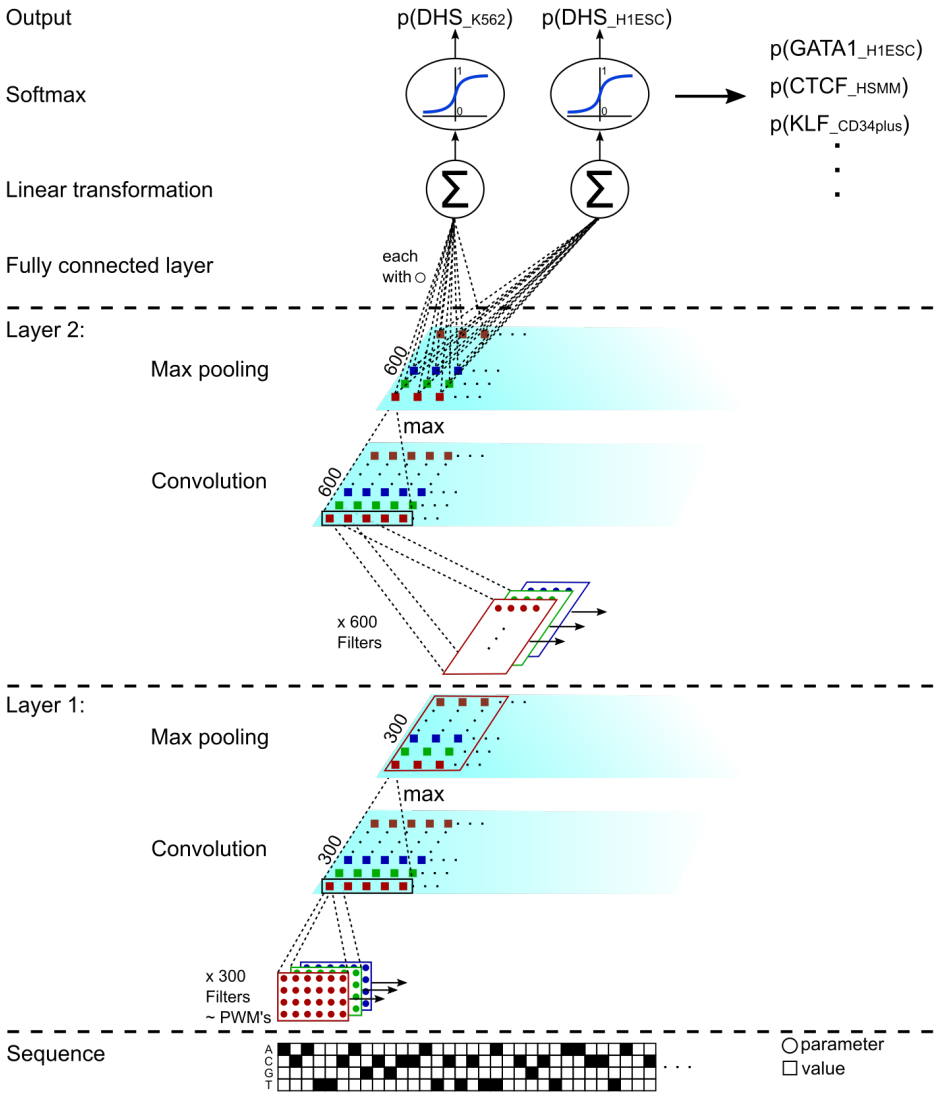
Chromatin Feature Networks



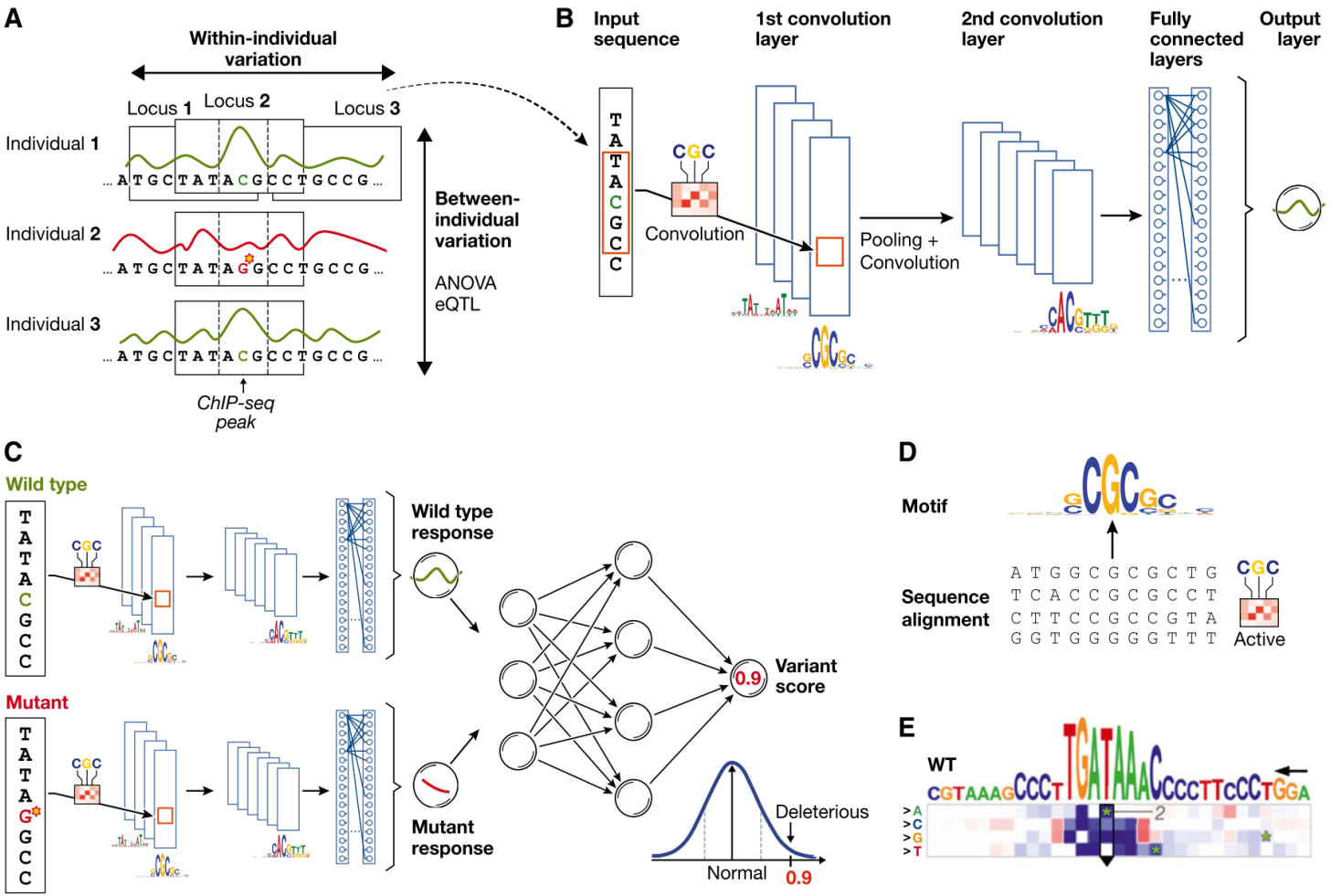
Chromatin Feature Networks



Chromatin Feature Networks

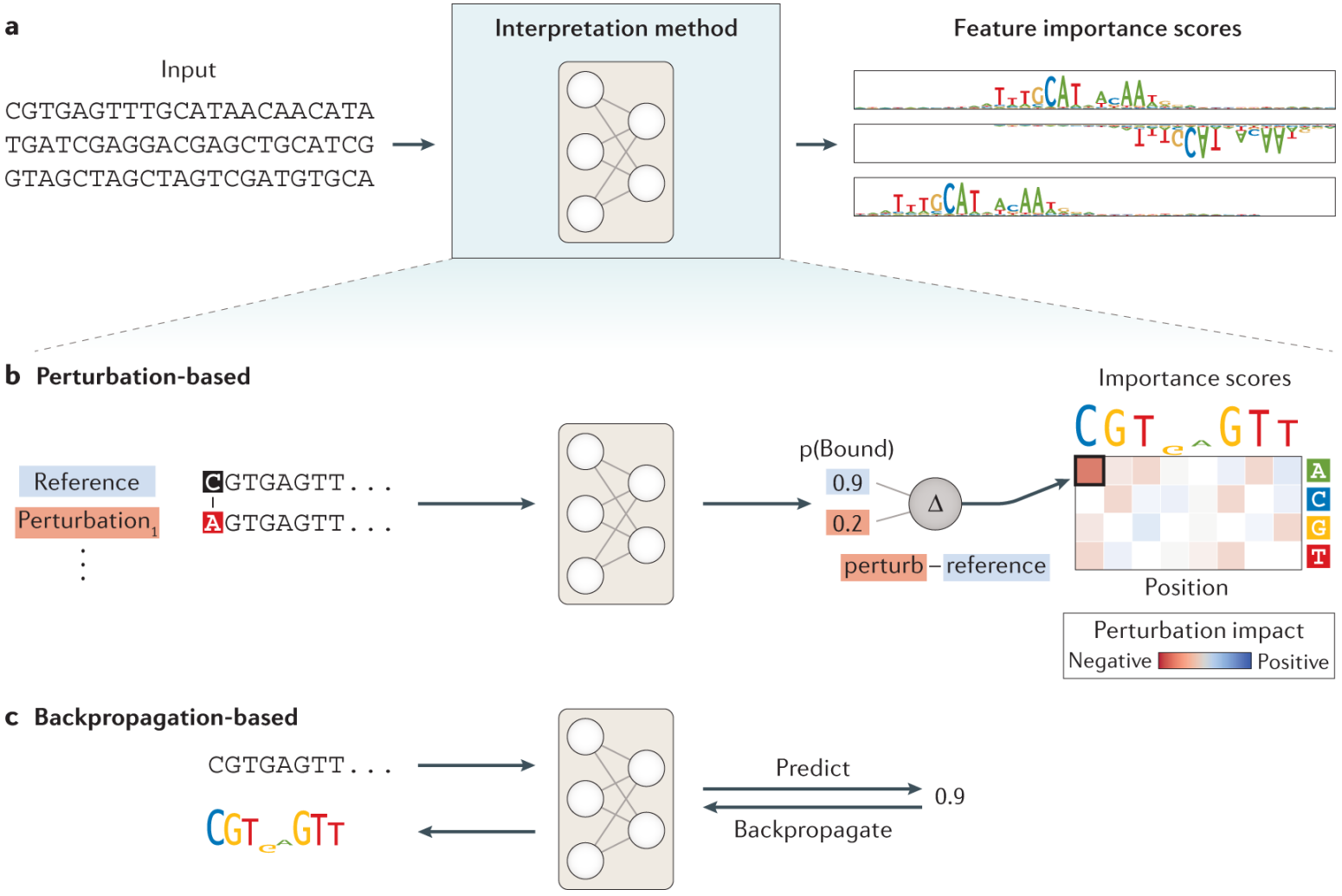


Utility



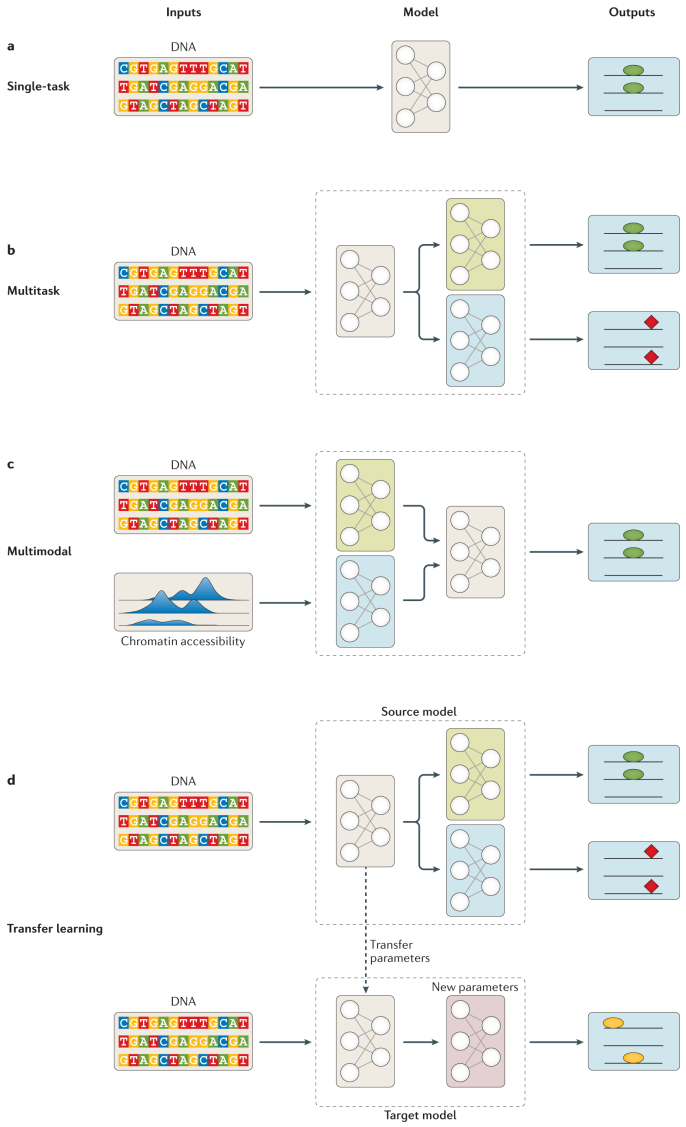
Angermueller *et al.* 2016

Interpretation



Gökcen *et al.* Nature Reviews Genetics 2019

Learning Strategy

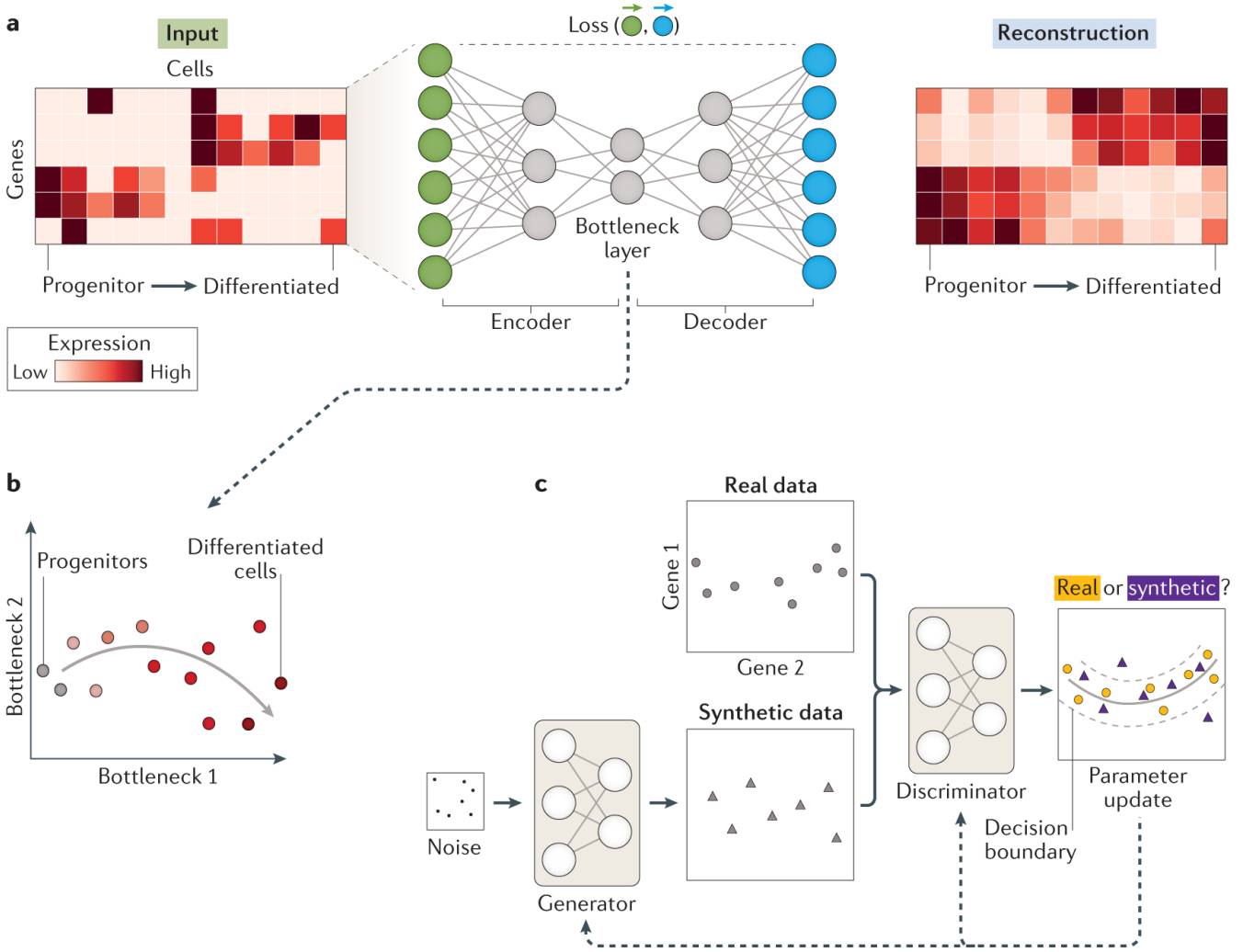


Gökcen *et al.* Nature Reviews Genetics 2019

More Examples, More Architectures

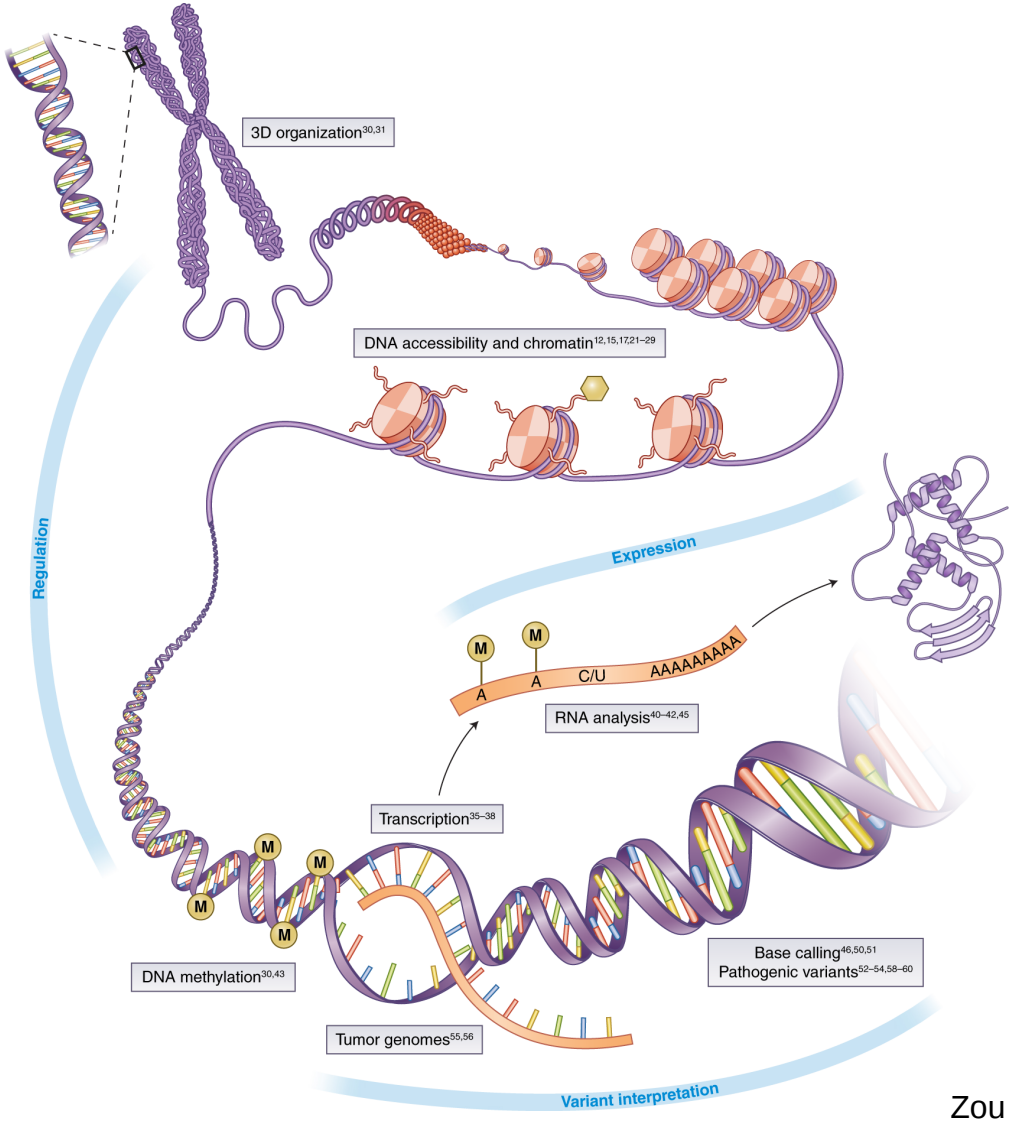
Gökcen *et al.* Nature Reviews Genetics 2019

AutoEncoders and GANs



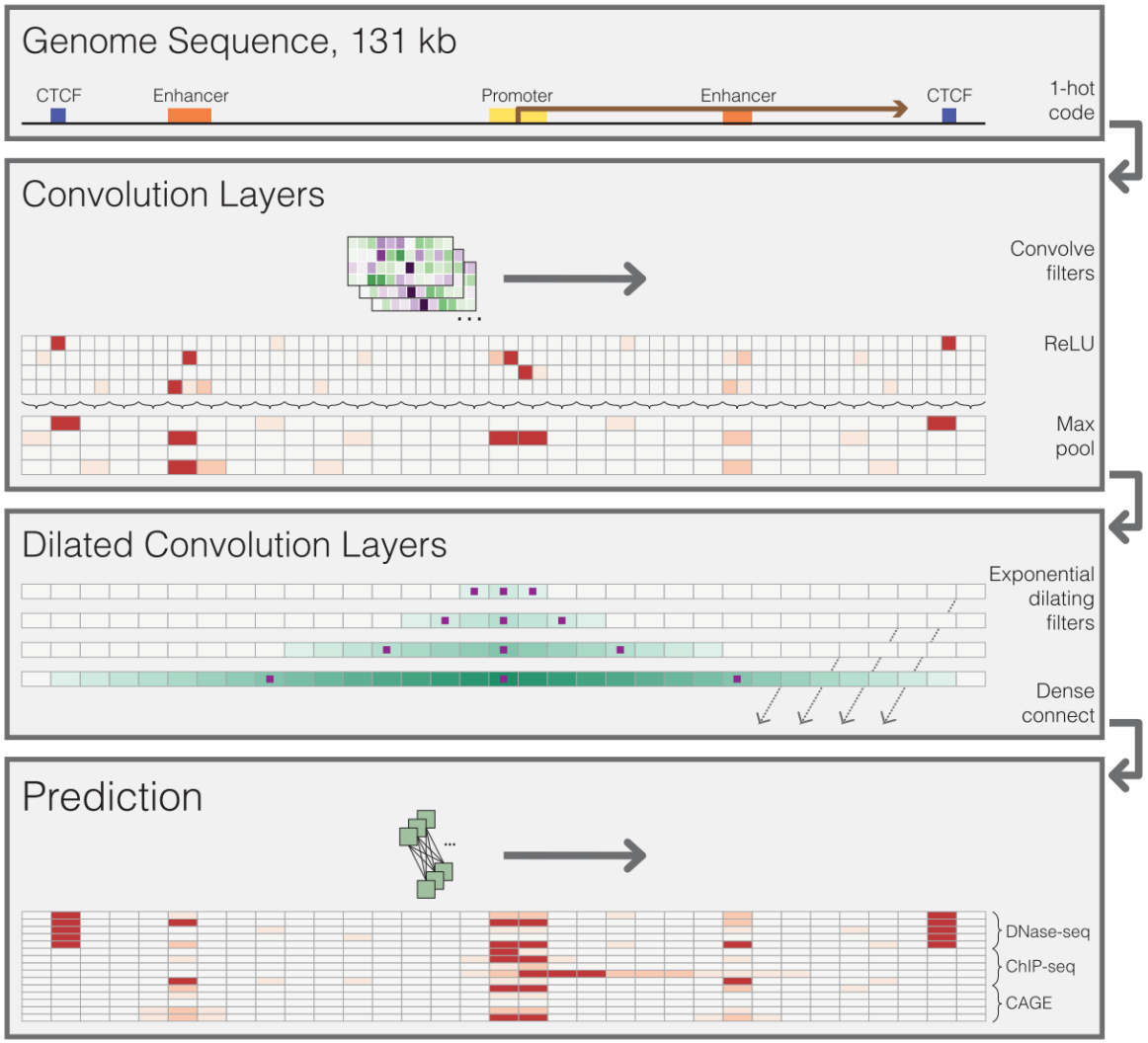
Gökcen *et al.* Nature Reviews Genetics 2019

Deep Learning in Genomics



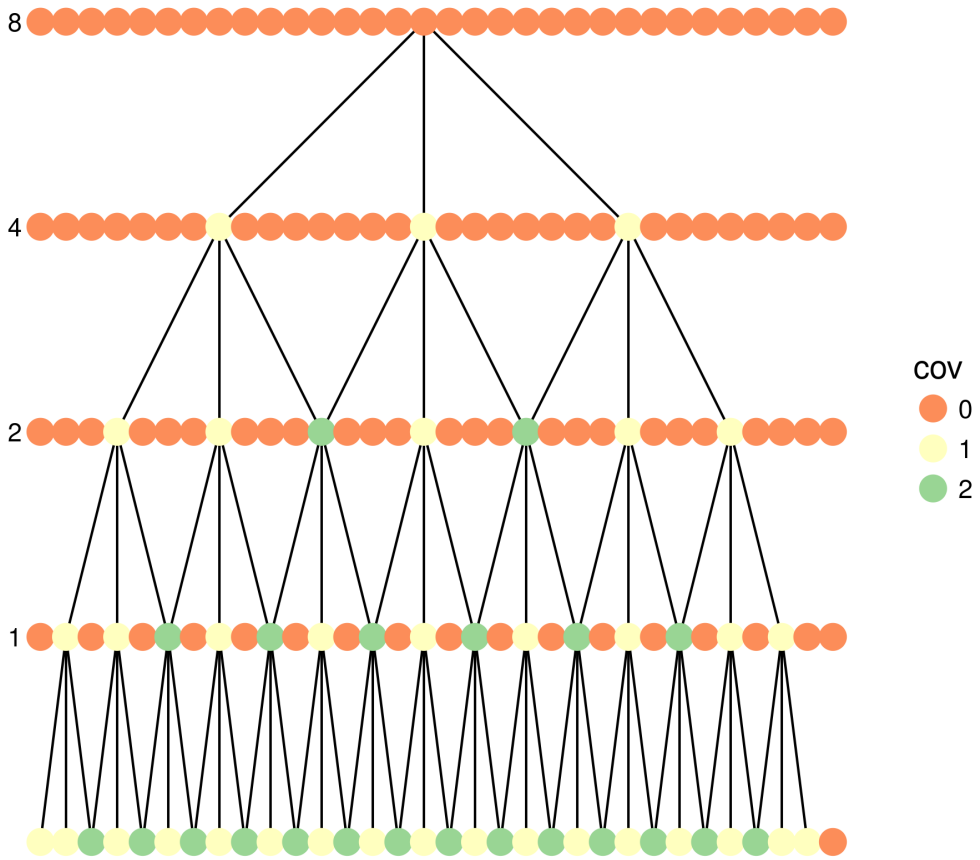
Zou *et al.* Nature Genetics 2019

Examples – Dilated Networks

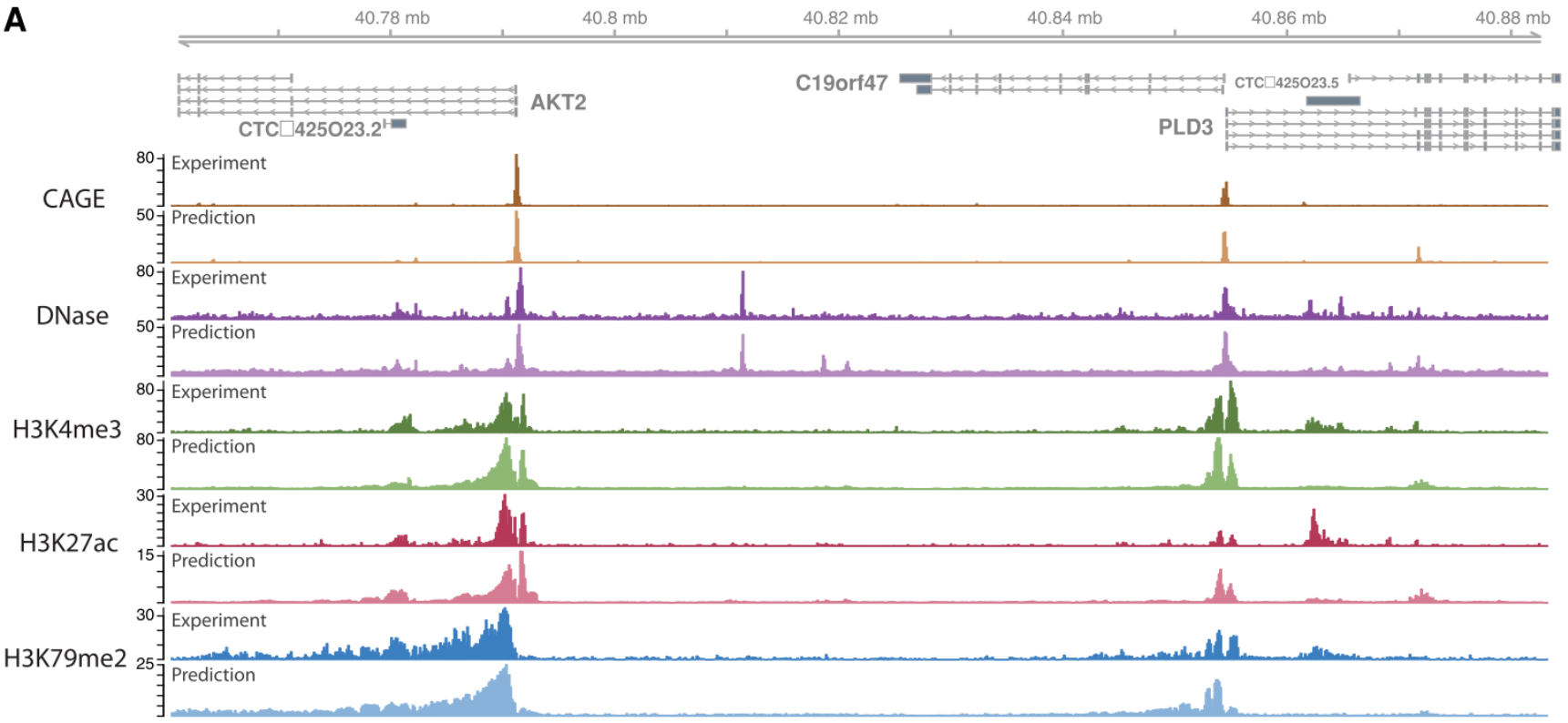


Kelley *et al.* Genome Research 2018

Examples – Dilated Networks

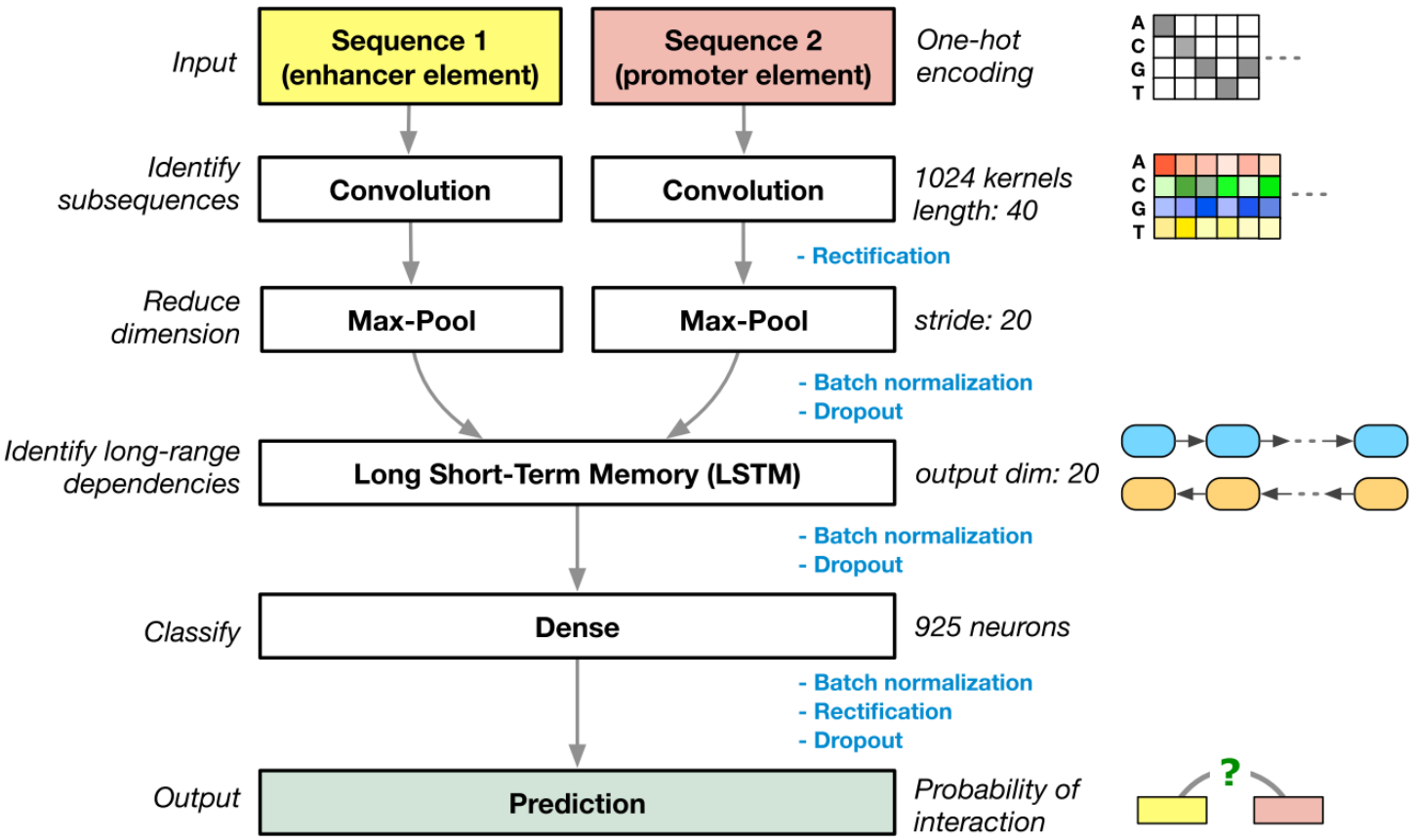


Examples – Dilated Networks



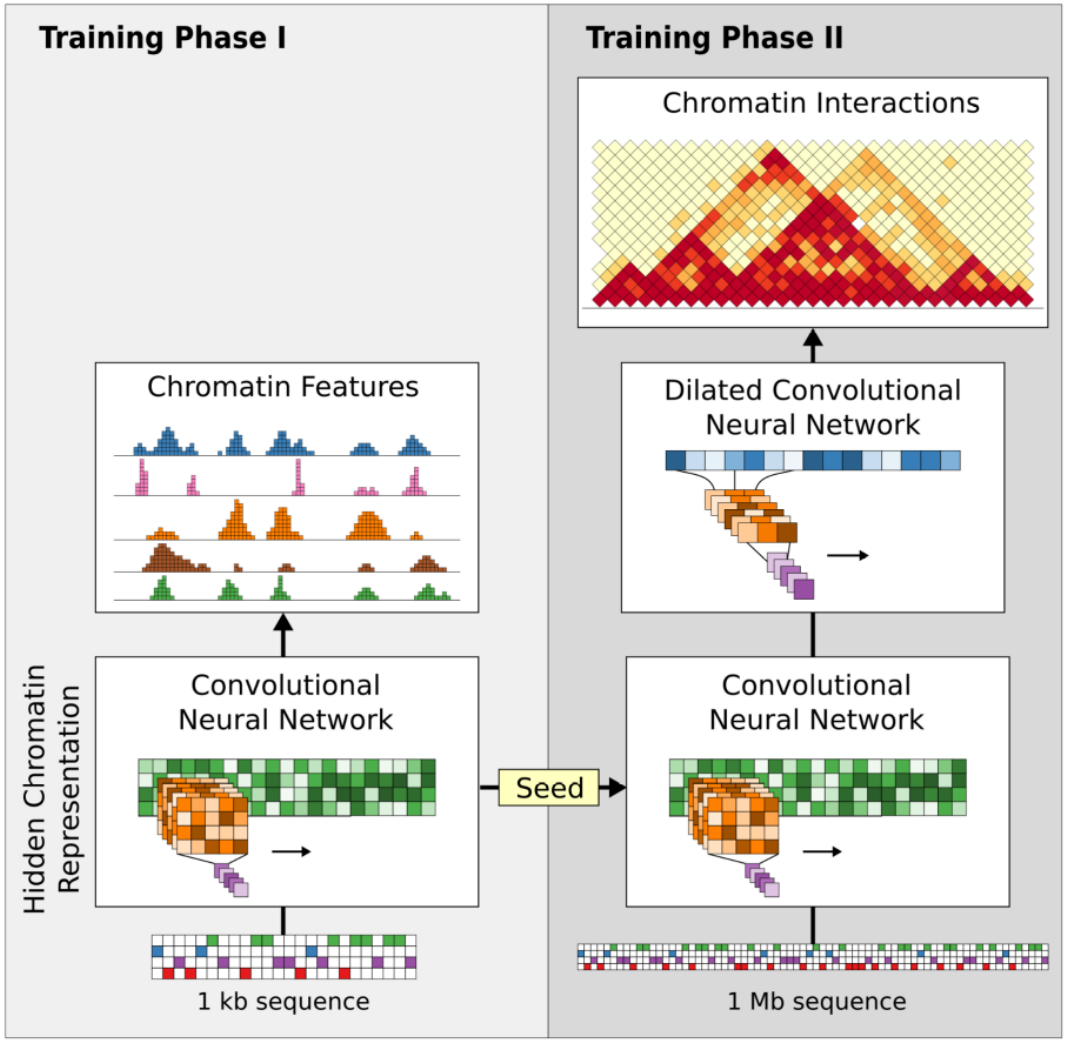
Kelley *et al.* Genome Research 2018

Examples – Enhancer – Promoter - Contacts



Singh *et al.* Quantitative Biology 2019

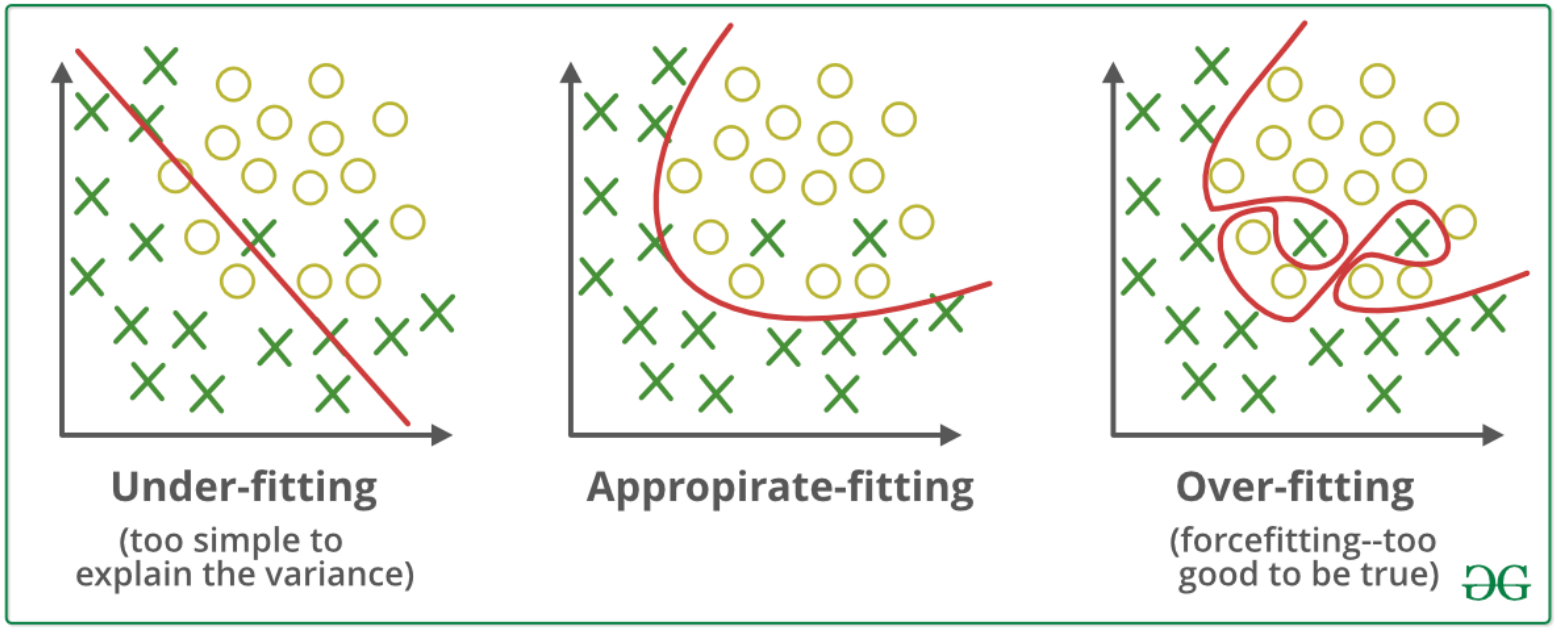
Examples – 3D Genome



Schwessinger *et al.* bioRxiv 2019

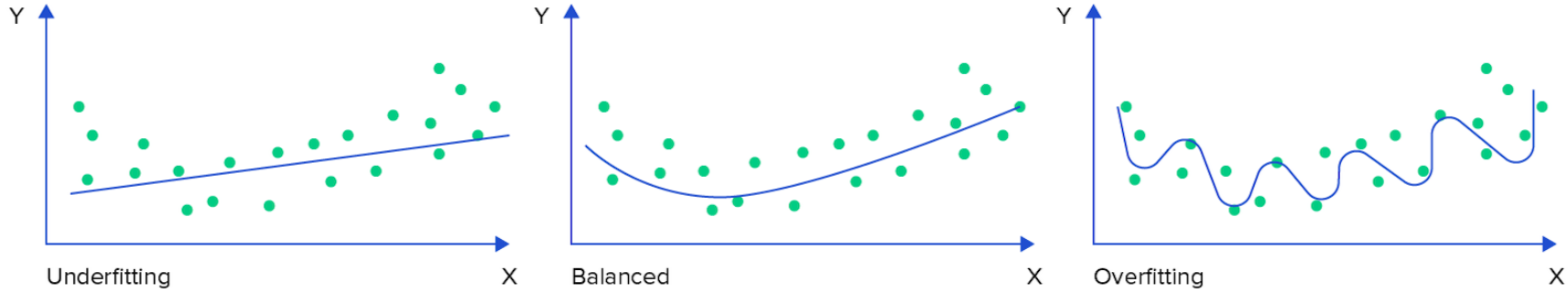
4) Basic Practical Aspects

Overfitting / Underfitting



Source towardsdatascience.com

Overfitting / Underfitting



Source towardsdatascience.com

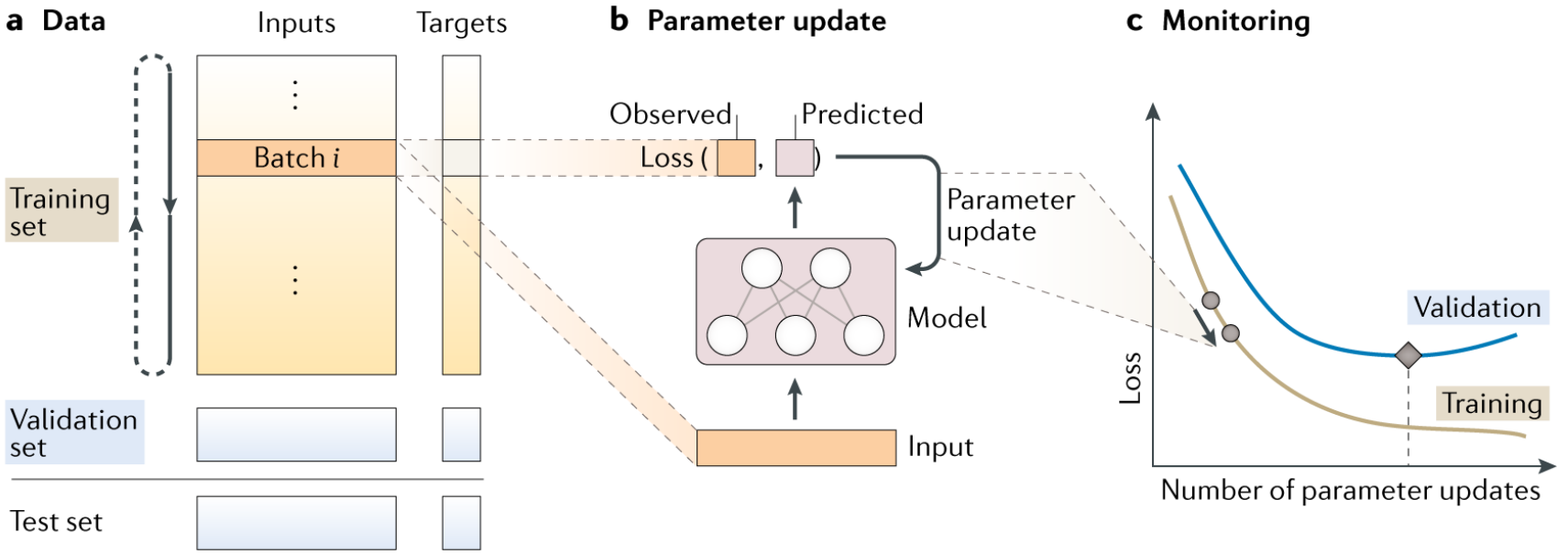
Overfitting / Underfitting

Model to simple underfitting

Model to complex overfitting

Source EDS

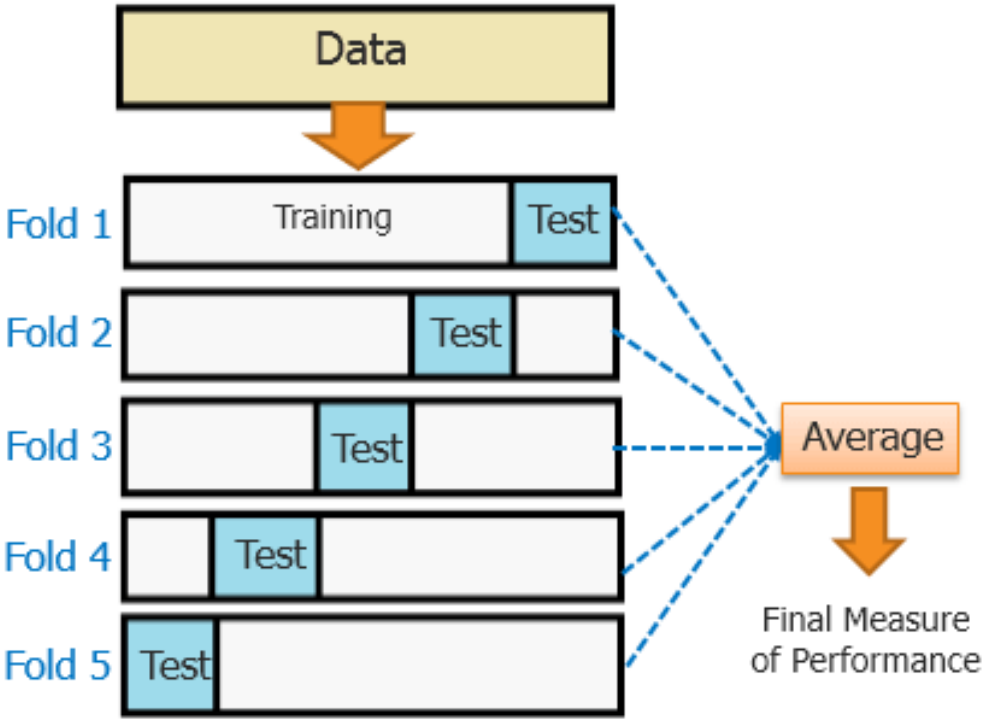
Train, Validation & Test Sets (& Cross-Validation)



Gökcen *et al.* Nature Reviews Genetics 2019

<u>In practice:</u> Split data into stratified train, valid. and test sets. Use cross-validation if low on data points.

Train, Validation & Test Sets (& Cross-Validation)



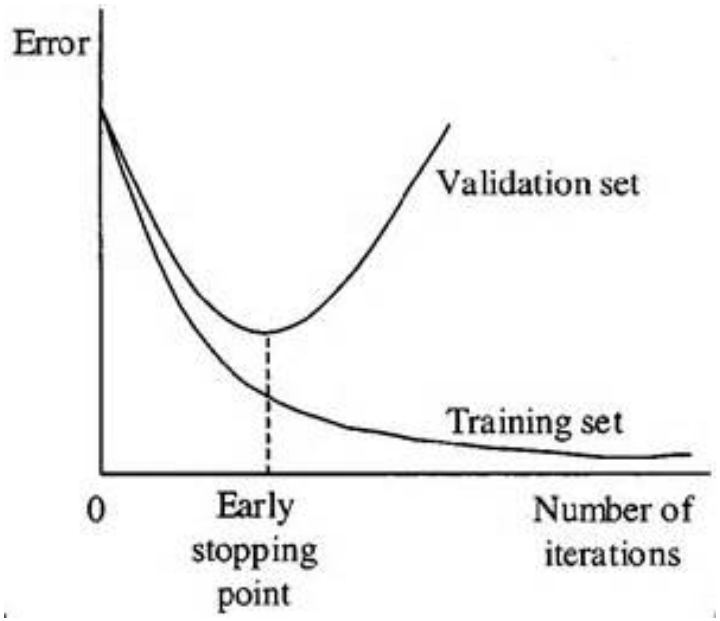
Source blog.contactsunny.com

In practice: Split data into stratified train, valid. and test sets. Use cross-validation if low on data points.

Overfitting / Underfitting

Tackle overfitting by in practice by:

- Optimizing regularization (L1 penalty, L2 penalty, Dropout)
- Use more data
- Early stopping
- Ensemble
- …

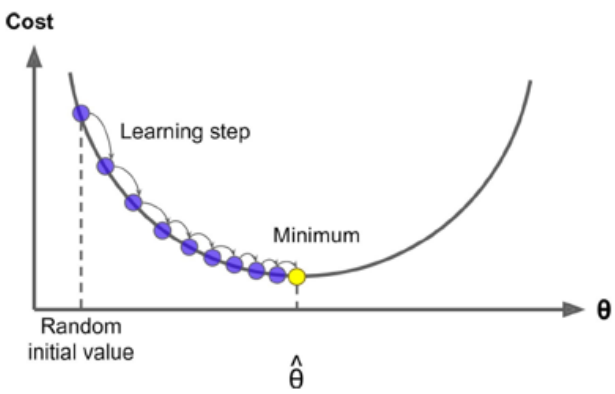


Overfitting / Underfitting

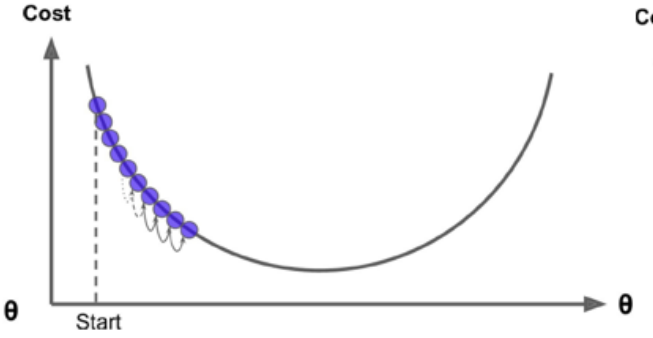
Tackle underfitting by in practice by:

- Increase the complexity / capacity of your model
- Use more data

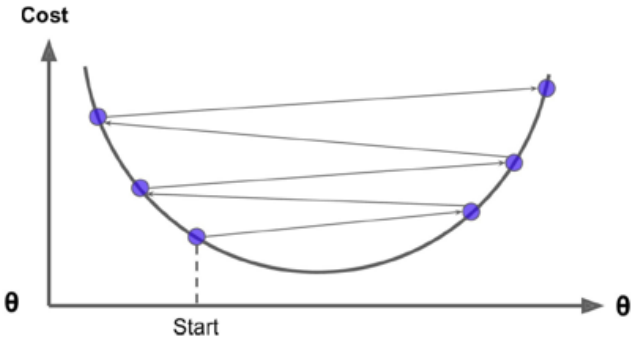
Learning Rate & Optimizer



Optimum learning rate : The model adjusts weights (θ) in subsequent training loops to arrive at cost minima.



Slow learning rate : Converges to cost minima but very slowly.



*Fast learning rate : **may** not converge to cost minima and the cost might keep increasing with further training loops.*

Image Credit : "Hands-on Machine Learning with Scikit-Learn and TensorFlow " by Aurelien Geron

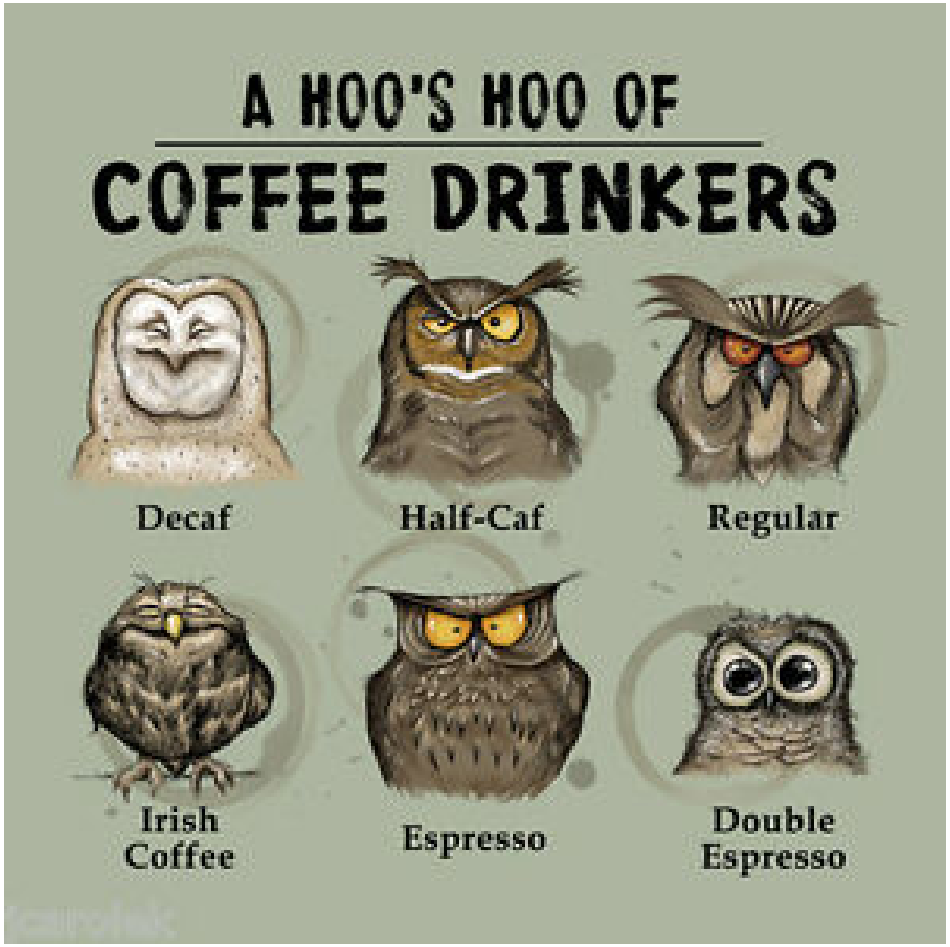
Disrupt4.0

CELLSTRAT

In practice: optimize the learning and pot. try different optimizers. But ADAM is very good default. Also use batches where you can.

Questions?

Coffee Break!



Practical

Head over to https://github.com/NGSchoolEU/ngs19

Scroll down to **Deep learning methods for genomics** and use the **Colab** link.

https://colab.research.google.com/drive/1SRHe_SXmKeXImNBR6tnhFQ3eThM4-iZu