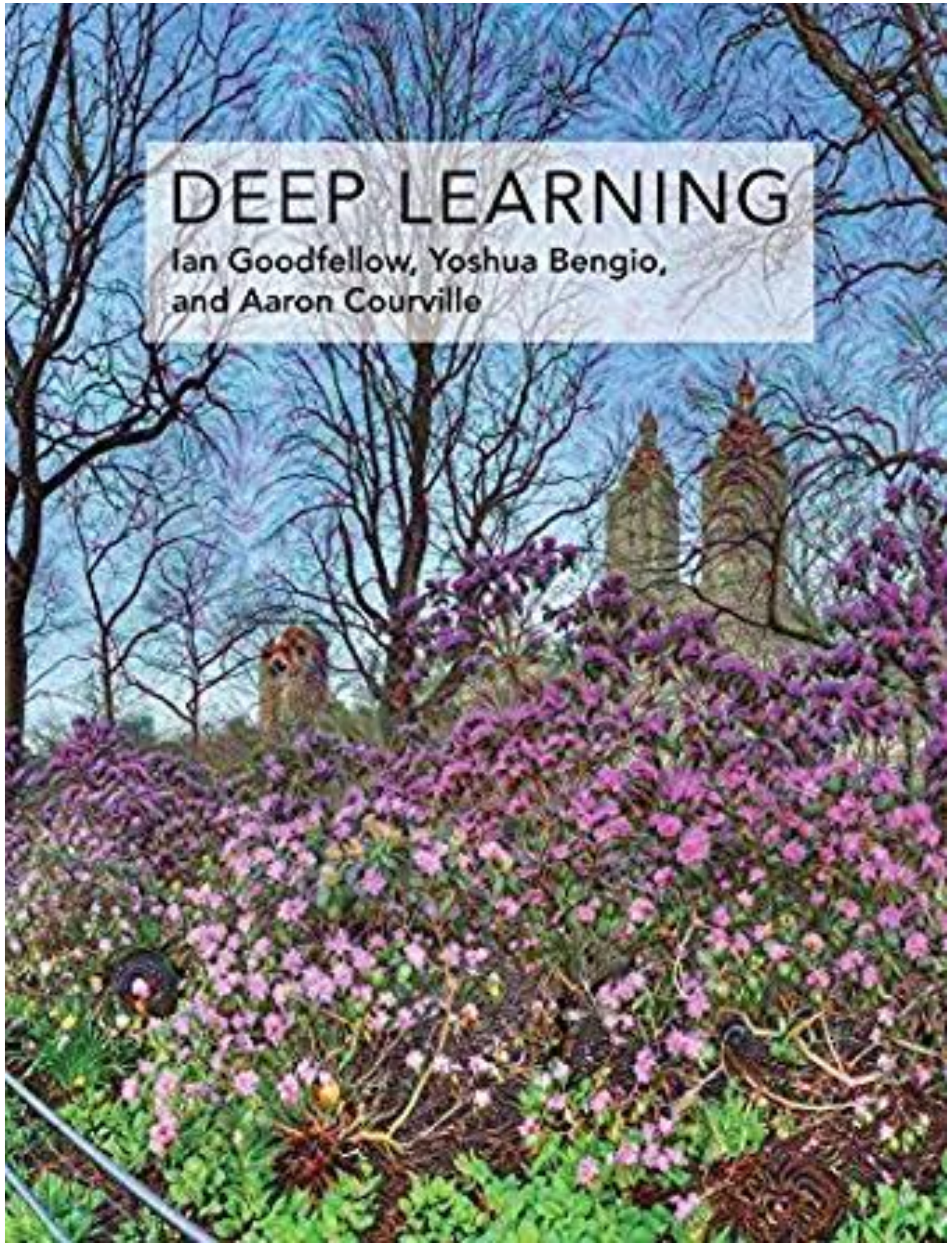
Neural networks

William Ratcliff

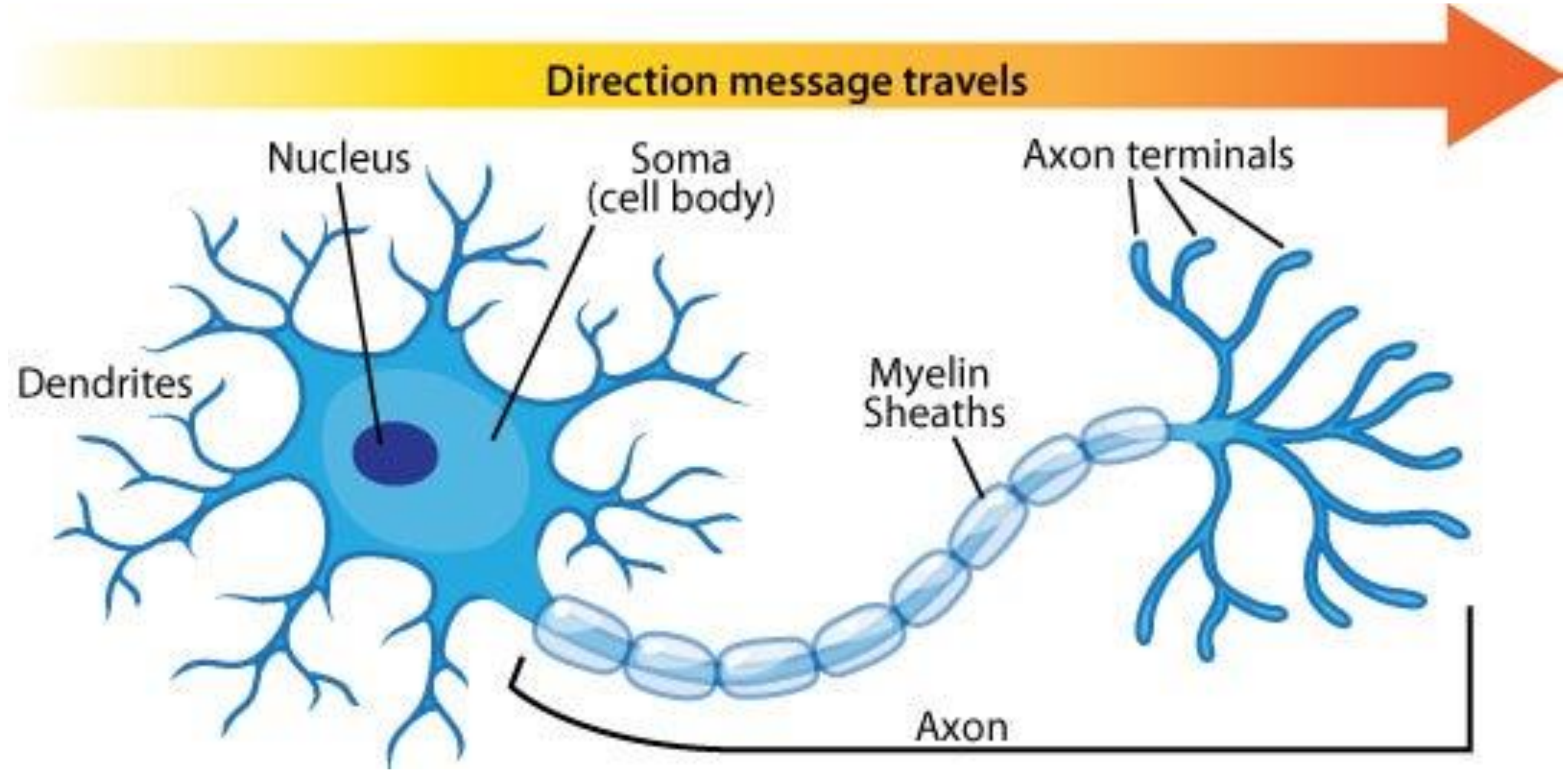
NIST; University of Maryland


DEEP LEARNING
Ian Goodfellow, Yoshua Bengio,
and Aaron Courville

Where to find the Code

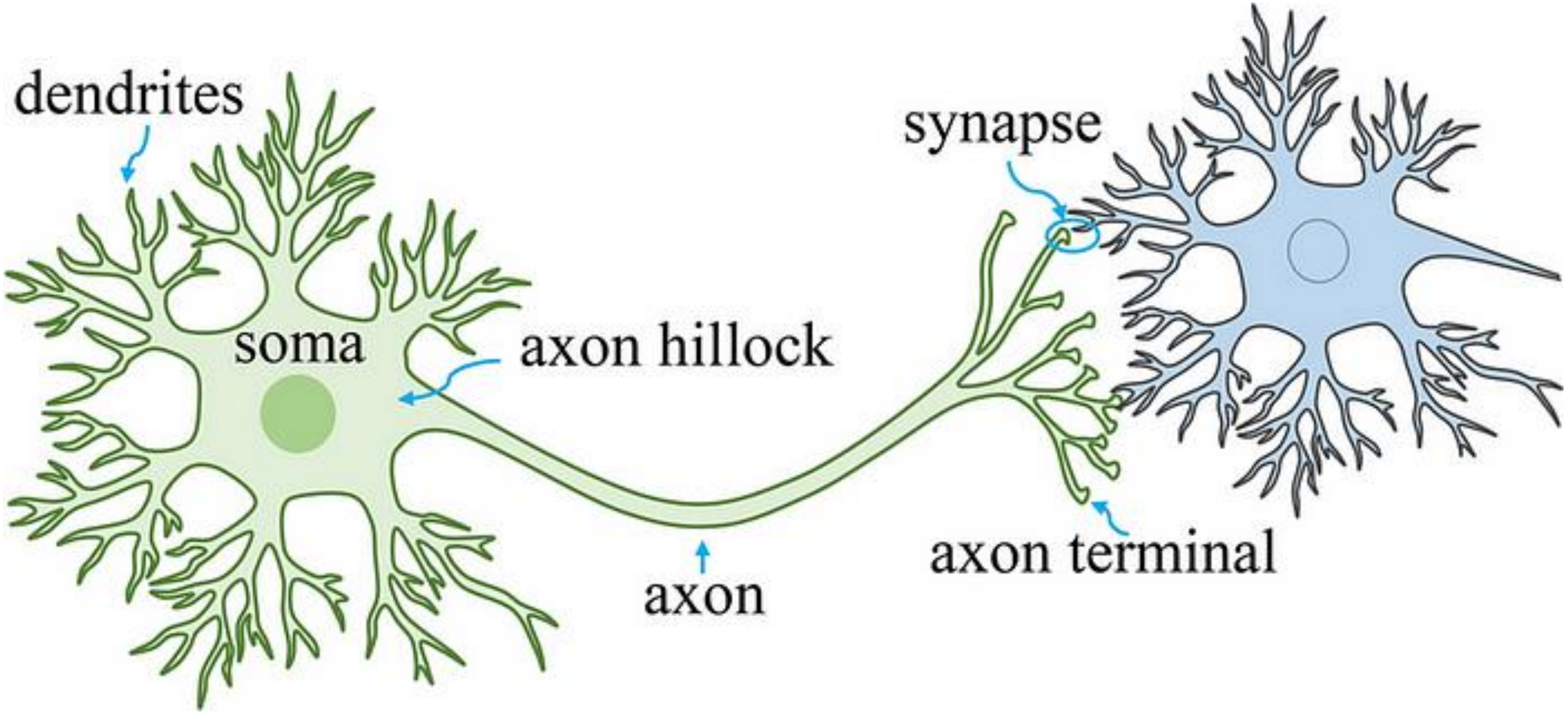
https://bit.ly/3R9UdYz

Neurons!



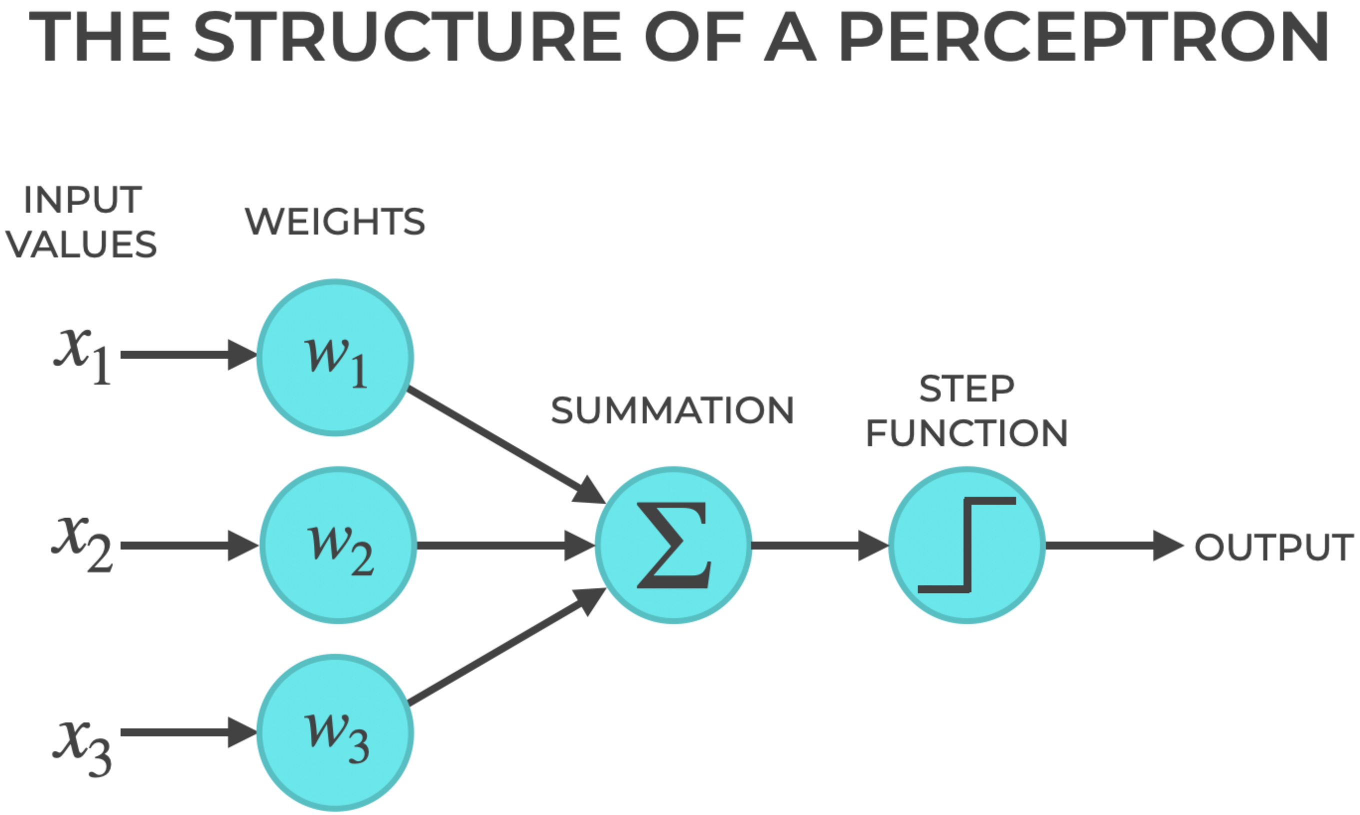
https://askabiologist.asu.edu/neuron-anatomy

Neurons!



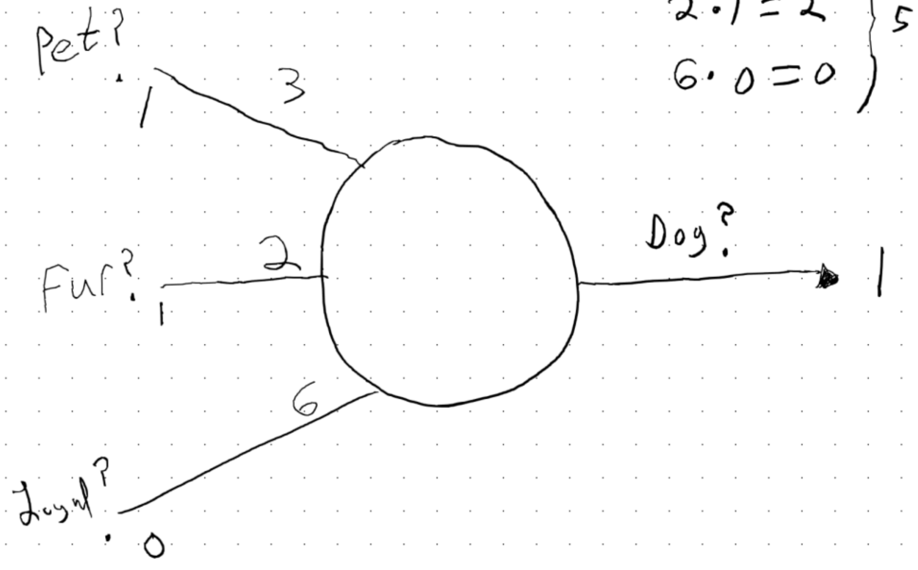
https://www.researchgate.net/figure/Diagram-of-neuron-and-synapse-Information-transfer-occurs-at-the-synapse-a-junction_fig1_368474455

Perceptron

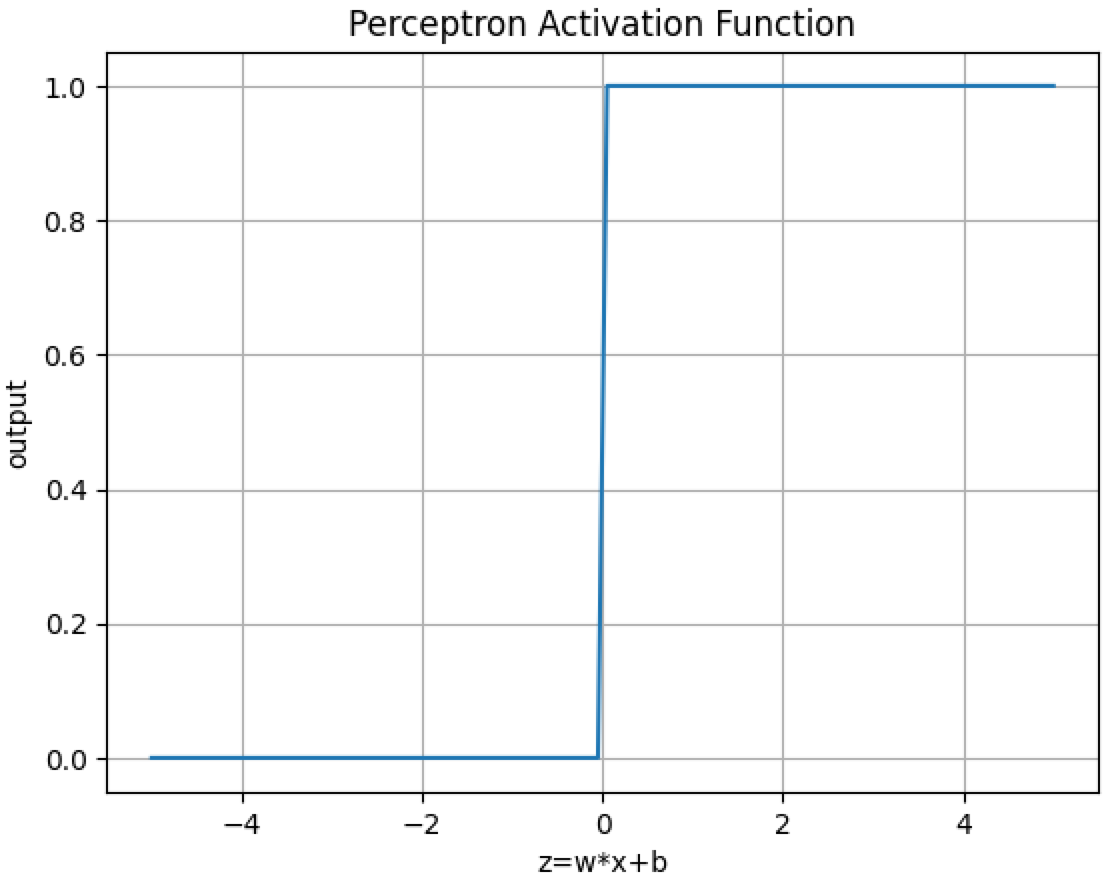


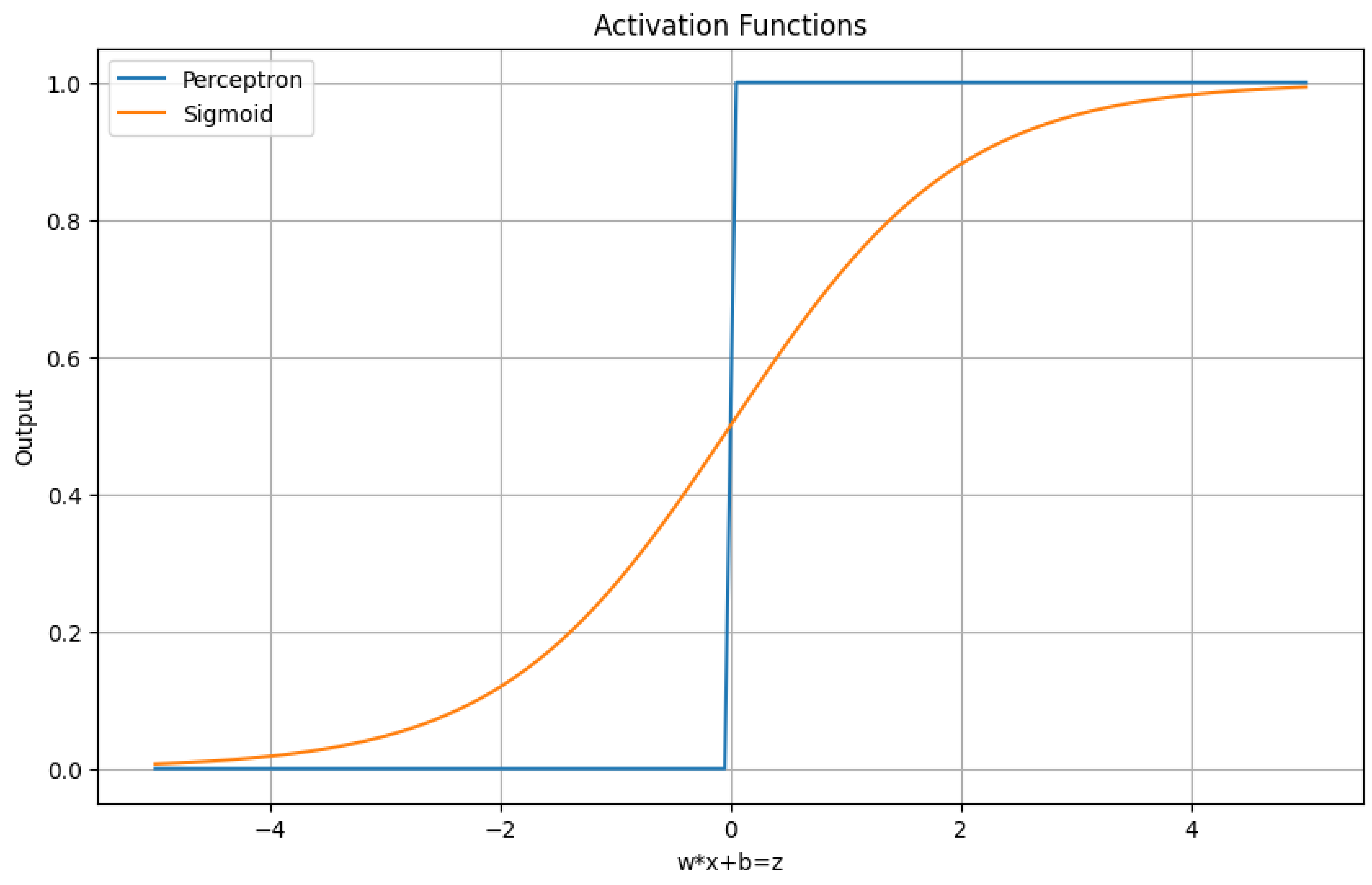
https://www.sharpsightlabs.com/blog/python-perceptron-from-scratch/

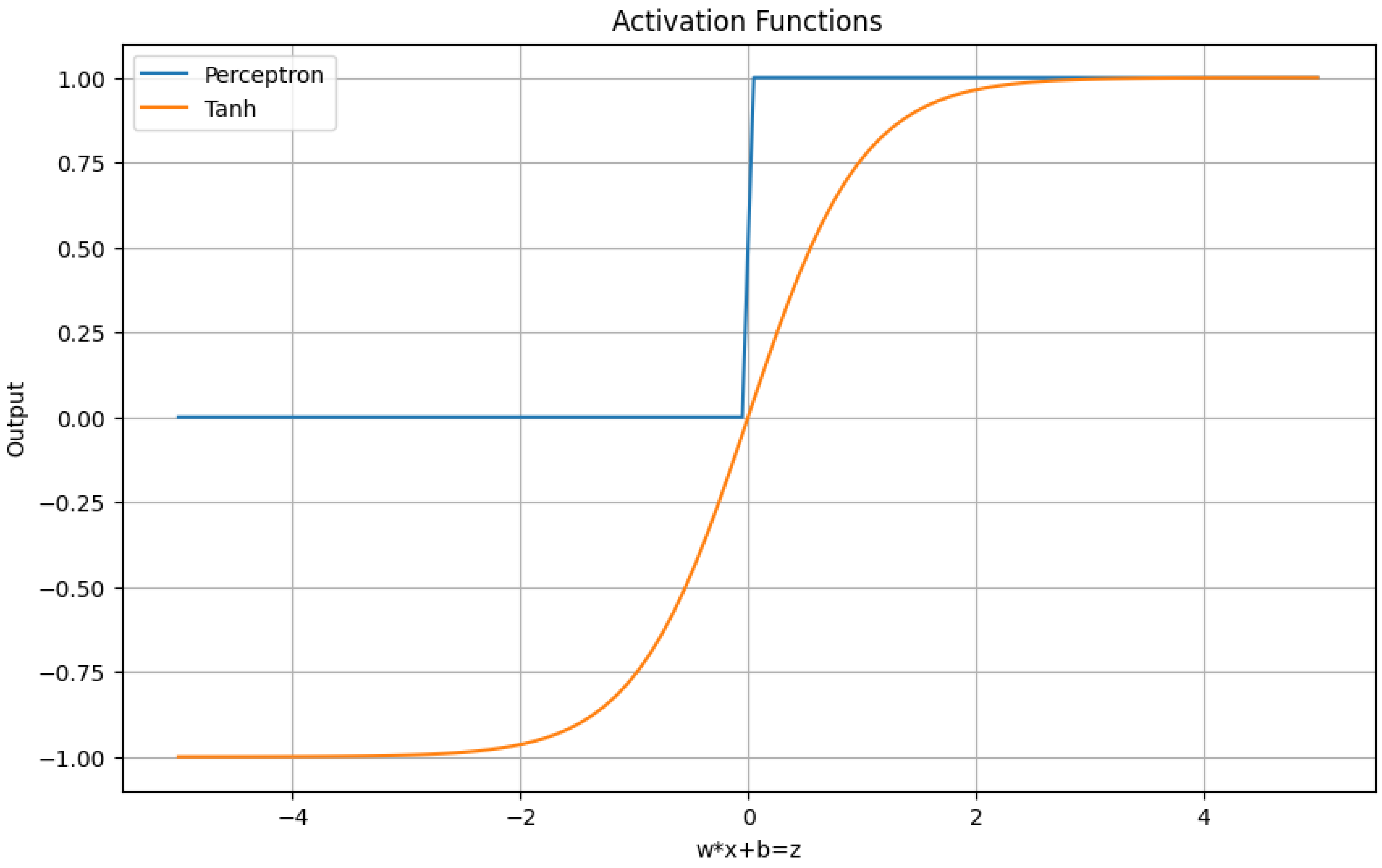
$$\left.\begin{array}{l}3\cdot 1=3\\ 2\cdot 1=2\\ 6\cdot 0=0\end{array}\right\} 5>4$$



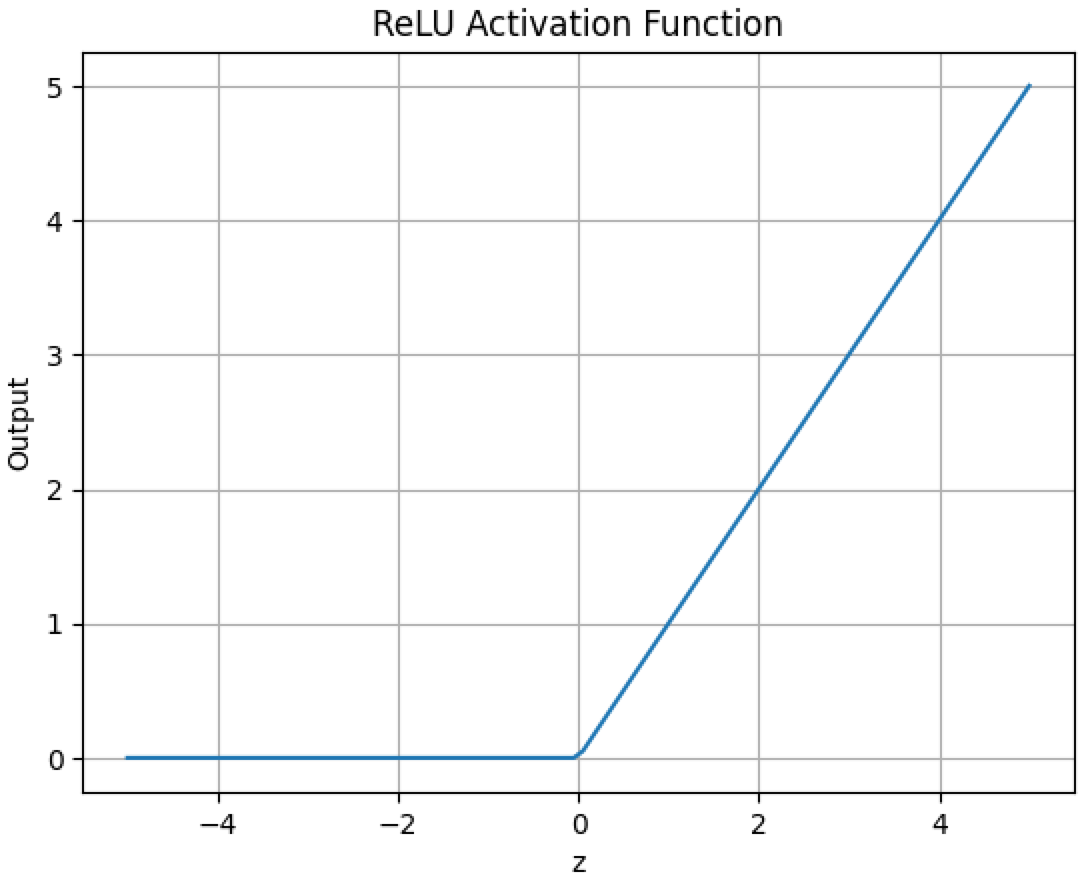
$$\sum_i w_i x_i \begin{array}{ll} > \text{threshold}, & \text{output } 1 \\ \leq \text{threshold}, & \text{output } 0 \end{array}$$


Perceptron Activation Function
1.0
0.8
0.6
0.4
0.2
0.0
output
−4
−2
0
2
4
z=w*x+b

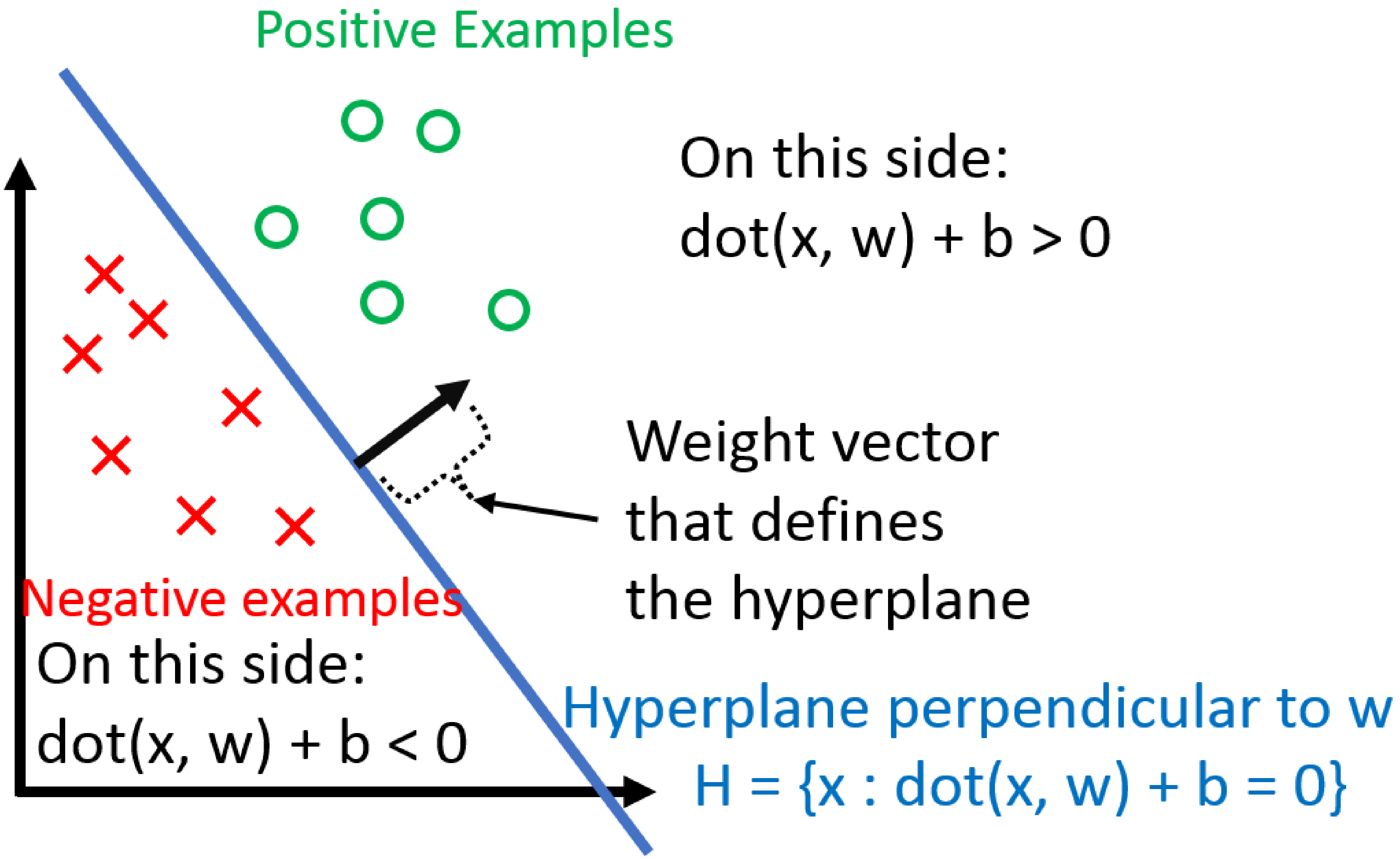

Activation Functions
Perceptron
Sigmoid
Output
w*x+b=z
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4



Activation Functions
Perceptron
Tanh
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Output
−4
−2
0
2
4
w*x+b=z


ReLU Activation Function
Output
z
0
1
2
3
4
5
−4
−2
0
2
4

Perceptron learning



https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html

Perceptron Algorithm

Now that we know what the $\mathbf{w}$ is supposed to do (defining a hyperplane the separates the data), let's look at how we can get such $\mathbf{w}$.

Perceptron Algorithm

Initialize $\vec{w} = \vec{0}$	// Initialize $\vec{w}$. $\vec{w} = \vec{0}$ misclassifies everything.
while TRUE **do**	// Keep looping
$m = 0$	// Count the number of misclassifications, m
for $(x_i, y_i) \in D$ **do**	// Loop over each (data, label) pair in the dataset, D
if $y_i(\vec{w}^T \cdot \vec{x}_i) \leq 0$ **then**	// If the pair $(\vec{x}_i, y_i)$ is misclassified
$\vec{w} \leftarrow \vec{w} + y\vec{x}$	// Update the weight vector $\vec{w}$
$m \leftarrow m + 1$	// Counter the number of misclassification
end if	
end for	
if $m = 0$ **then**	// If the most recent $\vec{w}$ gave 0 misclassifications
break	// Break out of the while-loop
end if	
end while	// Otherwise, keep looping!

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html

Geometric Intuition

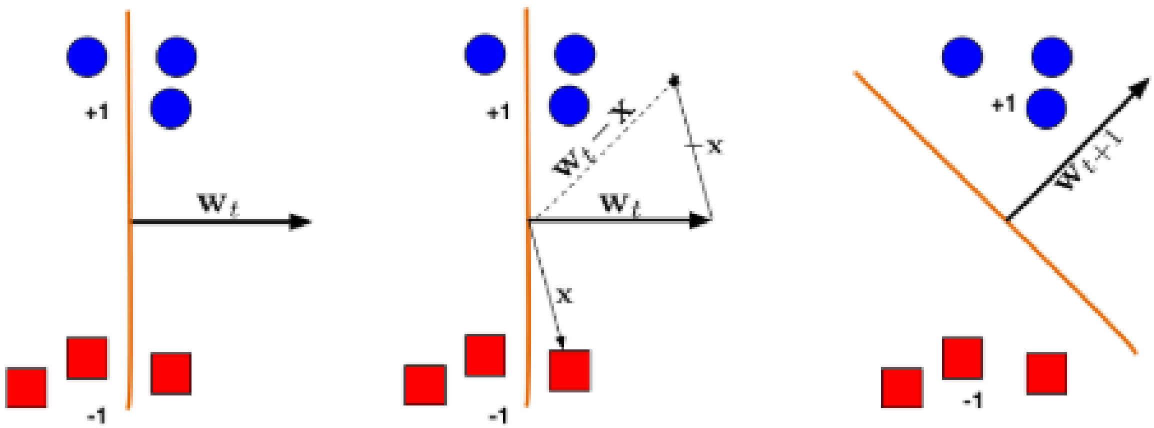
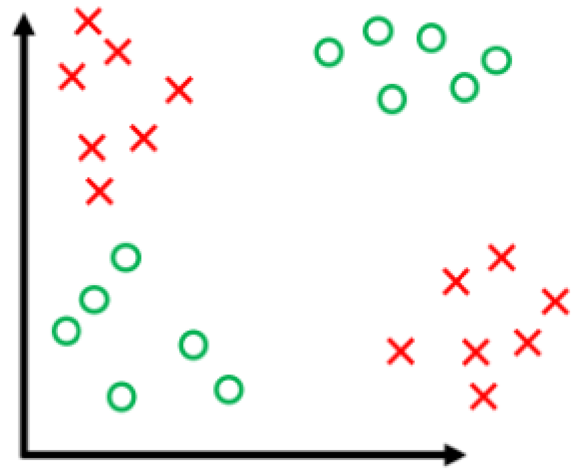
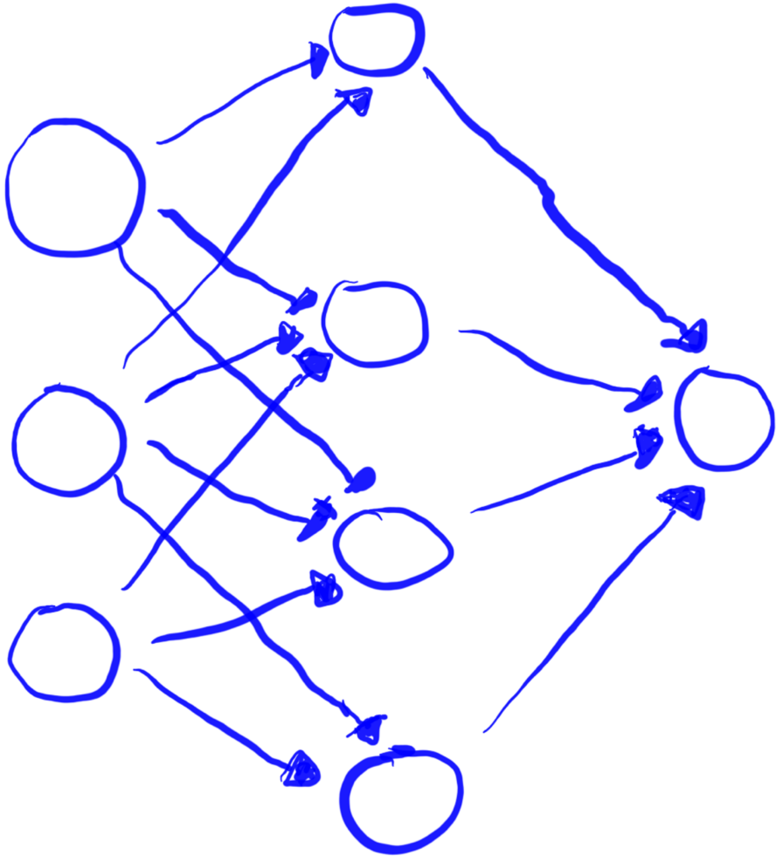


Illustration of a Perceptron update. (Left:) The hyperplane defined by $\mathbf{w}_t$ misclassifies one red (-1) and one blue (+1) point. (Middle:) The red point $\mathbf{x}$ is chosen and used for an update. Because its label is -1 we need to ***subtract*** *$\mathbf{x}$ from $\mathbf{w}_t$. (Right:) The udpated hyperplane $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{x}$ separates the two classes and the Perceptron algorithm has converged.*

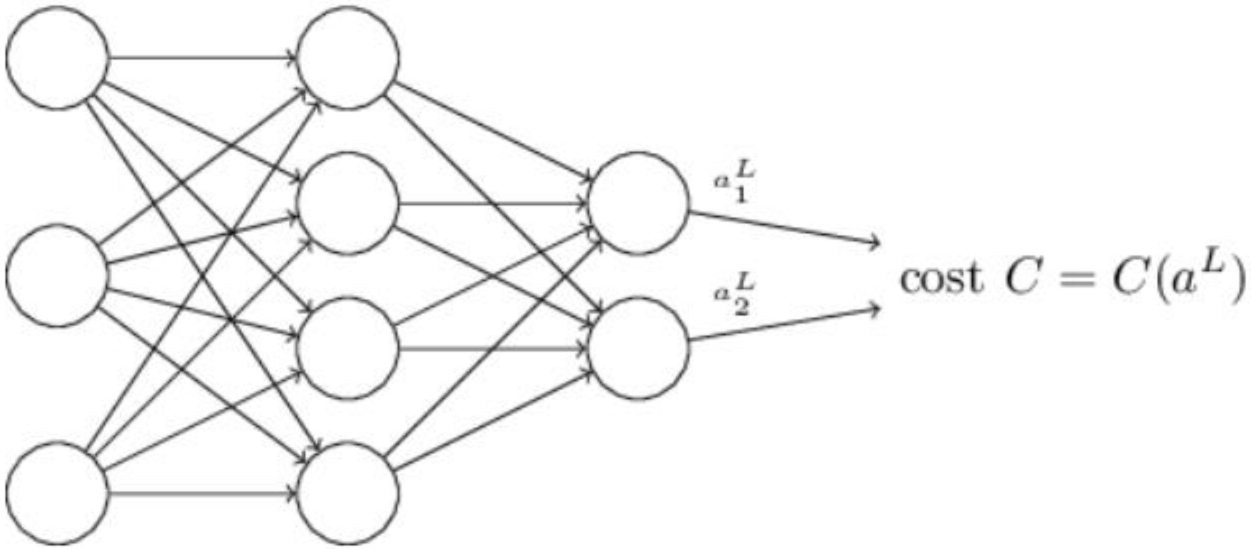
https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html



Multilayer Networks



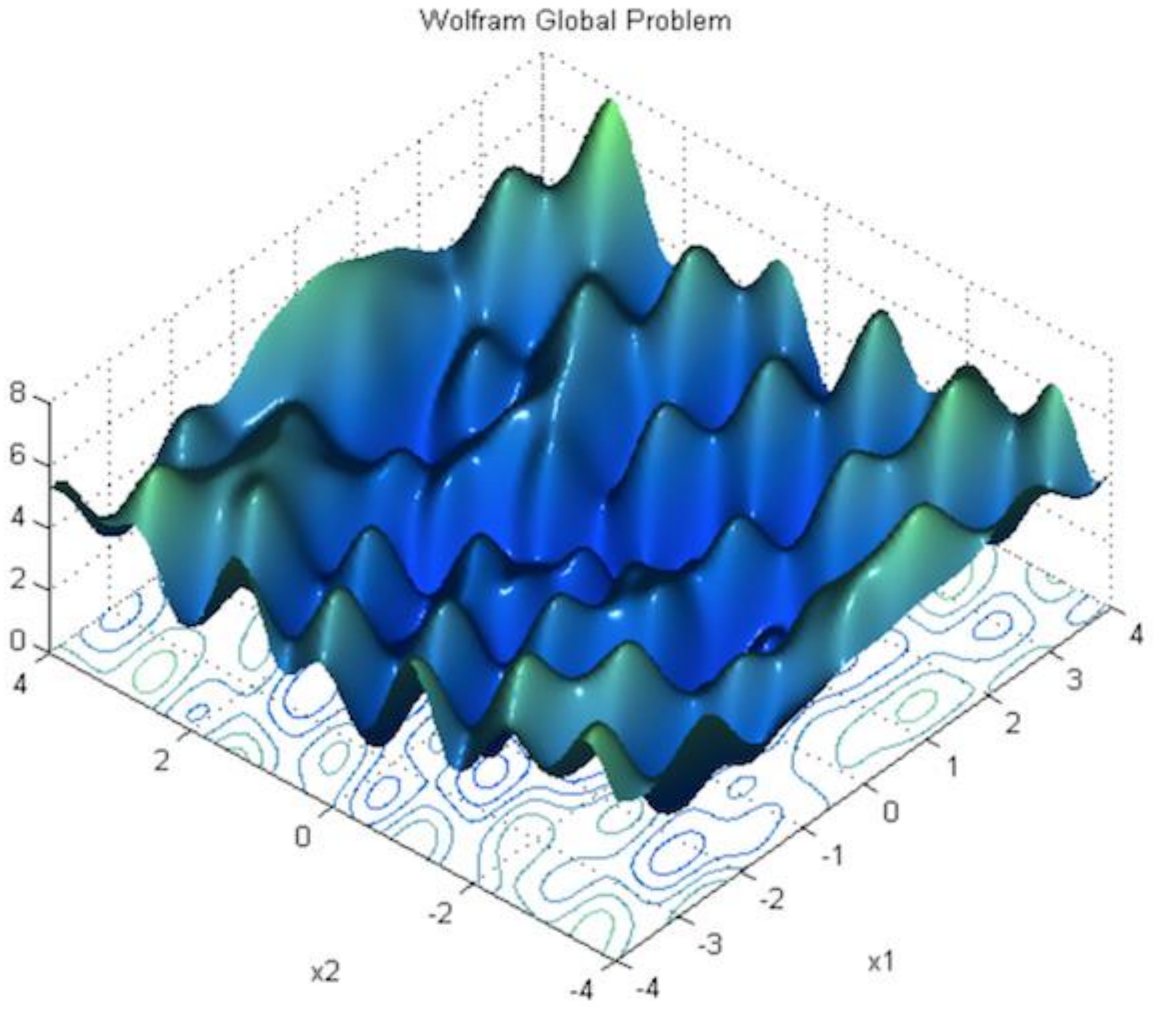
Cost



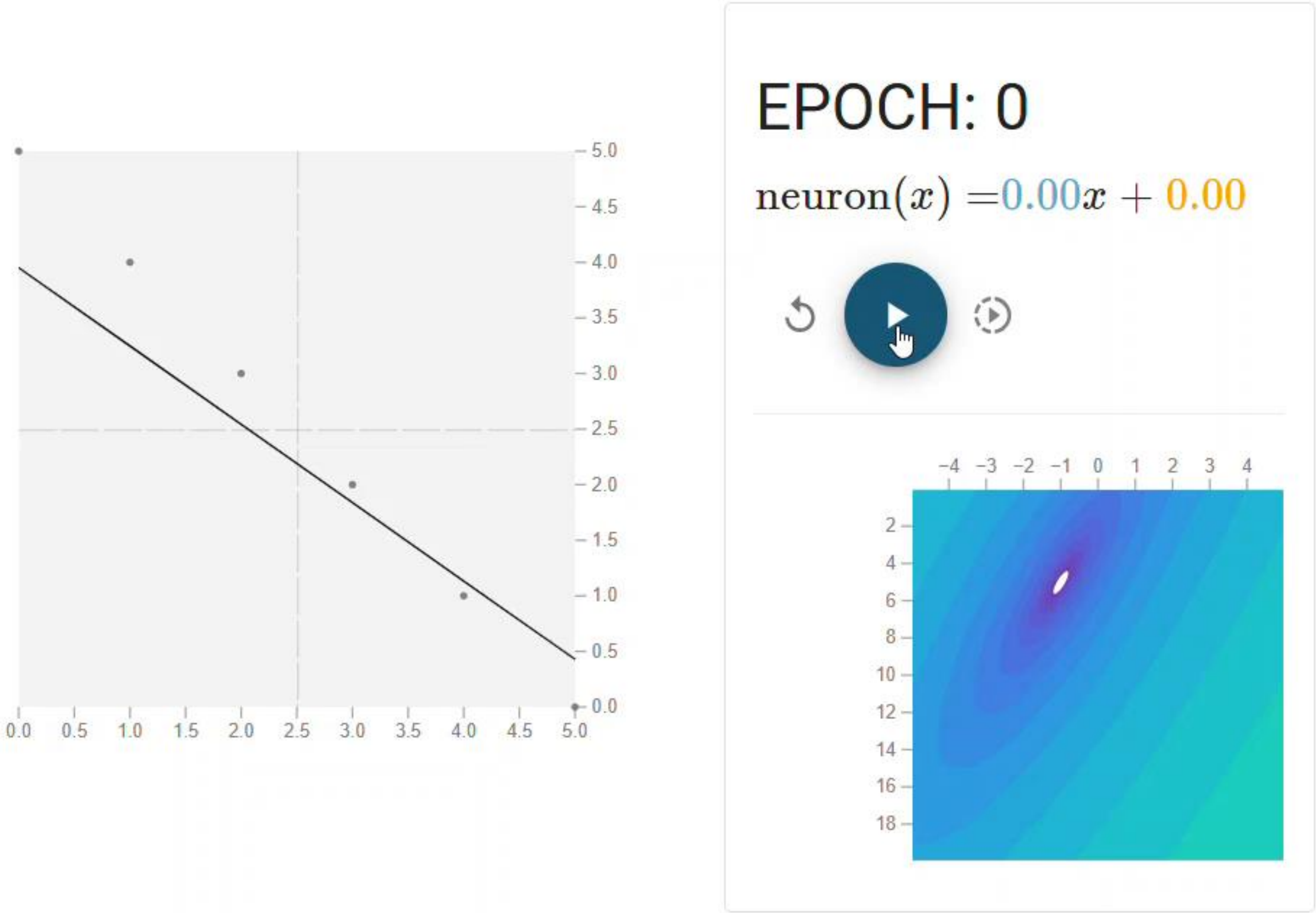
$$C = \frac{1}{2}\|y - a^L\|^2 = \frac{1}{2}\sum_j (y_j - a_j^L)^2$$

http://neuralnetworksanddeeplearning.com/chap2.html

Gradient Descent

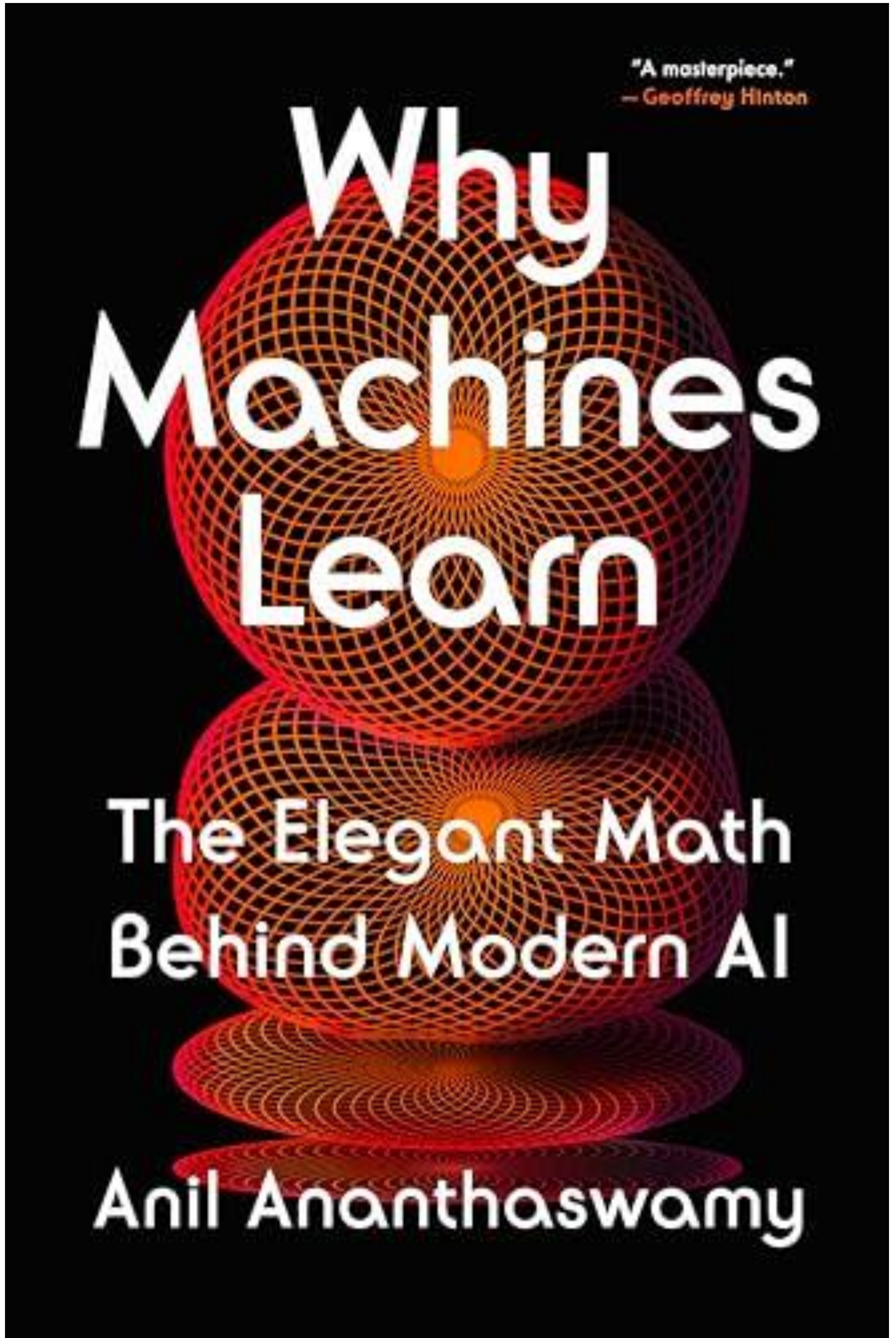


https://towardsdatascience.com/neural-network-optimization-7ca72d4db3e0

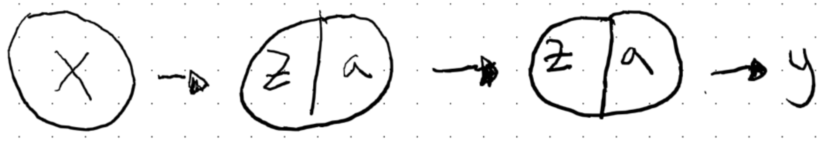


https://xnought.github.io/backprop-explainer/

Backpropagation



Back prop



$$z_1 = w_1 x + b_1 \qquad z_2 = w_2 a_1 + b_2$$

$$a_1 = \sigma(z_1) \qquad \hat{y} = \sigma(z_2)$$

$$e = y - \hat{y}$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Let $\mathcal{L} = e^2$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z))$$

Need:

$$\frac{\partial L}{\partial w_2}, \quad \frac{\partial L}{\partial b_2}, \quad \frac{\partial L}{\partial w_1}, \quad \frac{\partial L}{\partial b_1}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_2} \frac{\partial z_2}{\partial w_2}$$

$$L = e^2 \rightarrow \frac{\partial L}{\partial e} = 2e$$

$$e = y - \hat{y} \rightarrow \frac{\partial e}{\partial \hat{y}} = -1$$

$$\hat{y} = \sigma(z_2) \rightarrow \frac{\partial \hat{y}}{\partial z_2} = \sigma(z_2)(1 - \sigma(z_2))$$

$$= \hat{y}(1 - \hat{y})$$

$$z_2 = w_2 a_1 + b_2 \rightarrow \frac{\partial z_2}{\partial w_2} = a_1 \quad \& \quad \frac{\partial z_2}{\partial b_2} = 1$$

so: $$\frac{\partial L}{\partial w_2} = 2e(-)(\hat{y}(1-\hat{y}))(a_1)$$

$$\rightarrow \boxed{\frac{\partial L}{\partial w_2} = -2ea_1[\hat{y}(1-\hat{y})]}$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_2} \frac{\partial z_2}{\partial b_2}$$

$$\Rightarrow \frac{\partial L}{\partial b_2} = 2e(-1) \left[\hat{y} (1-\hat{y}) \right] (1)$$

$$= \boxed{-2e \left(\hat{y} (1-\hat{y}) \right)}$$

$$\Delta W_2 = -\alpha \frac{\partial L}{\partial W_2} \qquad W_2 \to W_2 + \Delta W_2$$

$$\Delta b_2 = -\alpha \frac{\partial L}{\partial W_2} \qquad b_2 \to b_2 + \Delta b_2$$

Your turn:

$$\frac{\partial L}{\partial w_1} = ?$$

$$\frac{\partial L}{\partial b_1} = ?$$

$$\Delta w_1 = ?$$

$$\Delta b_1 = ?$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_2} \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$= 2e(-1)\left(\hat{y}(1-\hat{y})\right) w_2 \left(a_1(1-a_1)\right) x$$

$$= \boxed{-2e\, x\, w_2[\hat{y}(1-\hat{y})]\ a_1(1-a_1)}$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_2} \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial b_1}$$

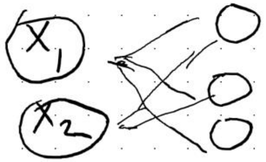
$$= 2e(-1)[\hat{y}(1-\hat{y})]\ w_2\ [a_1(1-a_1)]1$$

$$= \boxed{-2e\, w_2[\hat{y}(1-\hat{y})]\ a_1(1-a_1)}$$

Note: need values & weights!

$$\Delta w_1 = \alpha \frac{\partial L}{\partial w} \rightarrow w_1 \rightarrow w_1 + \Delta w_1$$

$$\Delta B_1 = \alpha \frac{\partial L}{\partial b_1} \rightarrow b_1 \rightarrow b_1 + \Delta b_1$$



$$\vec{x} \to [x_1, x_2]$$

$$z_1 = w_1^{11} x_1 + w_1^{21} x_2 + b_1^1$$

$$z_2 = w_1^{12} x_1 + w_1^{22} x_2 + b_1^2$$

$$z_3 = w_1^{13} x_1 + w_1^{23} x_2 + b_1^3$$

$$z_i = \begin{bmatrix} w_1^{11} & w_1^{21} \\ w_1^{12} & w_1^{22} \\ w_1^{13} & w_1^{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1^1 \\ b_1^2 \\ b_1^3 \end{bmatrix} = W_1^T x + b$$

$$W_1 = \begin{bmatrix} w_1^{11} & w_1^{12} & w_1^{13} \\ w_1^{21} & w_1^{22} & w_1^{23} \end{bmatrix}$$

Then similar to before, but w/ matrices

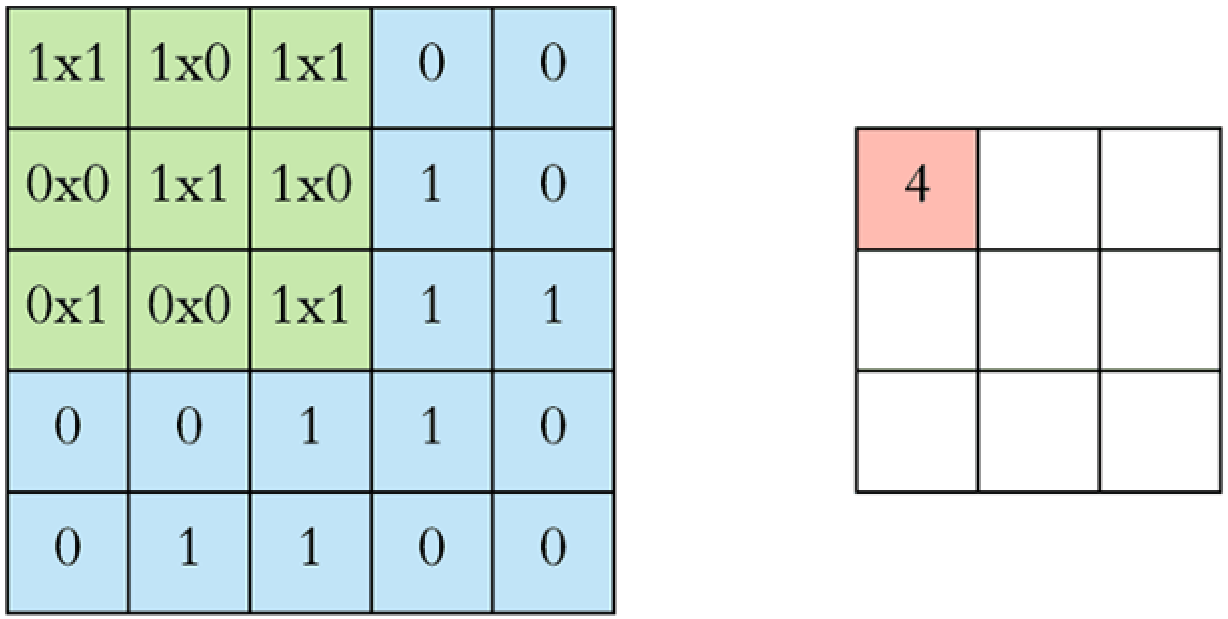
Convolutional Neural Networks

Hands On

https://tinyurl.com/3mmvny2r

Convolutional Layers

- Apply filter to image
- Extract high level features
- Reduce dimensionality

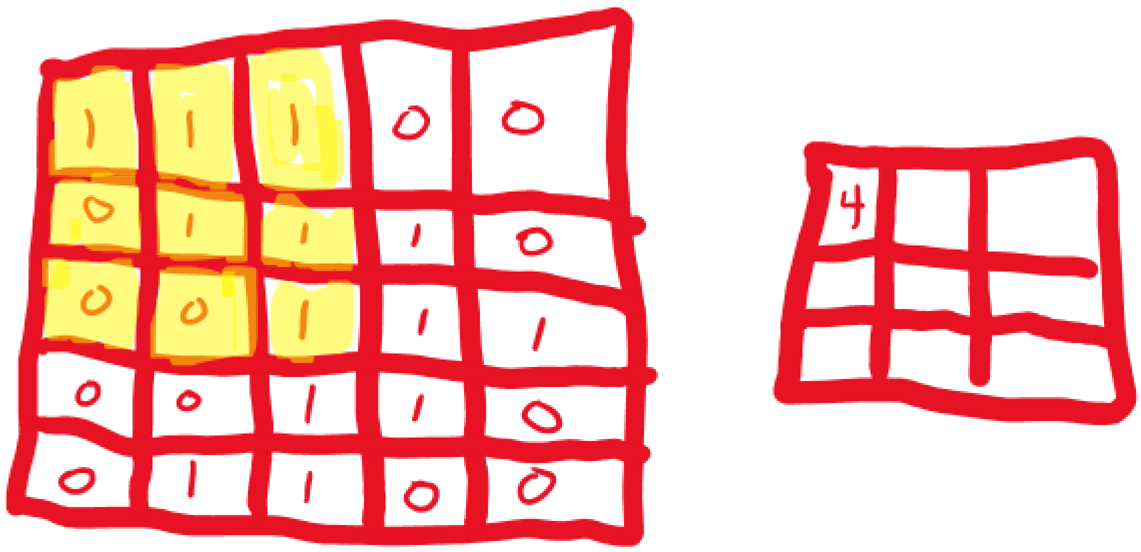


Source: https://towardsdatascience.com/pytorch-basics-how-to-train-your-neural-net-intro-to-cnn-26a14c2ea29

CNN

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

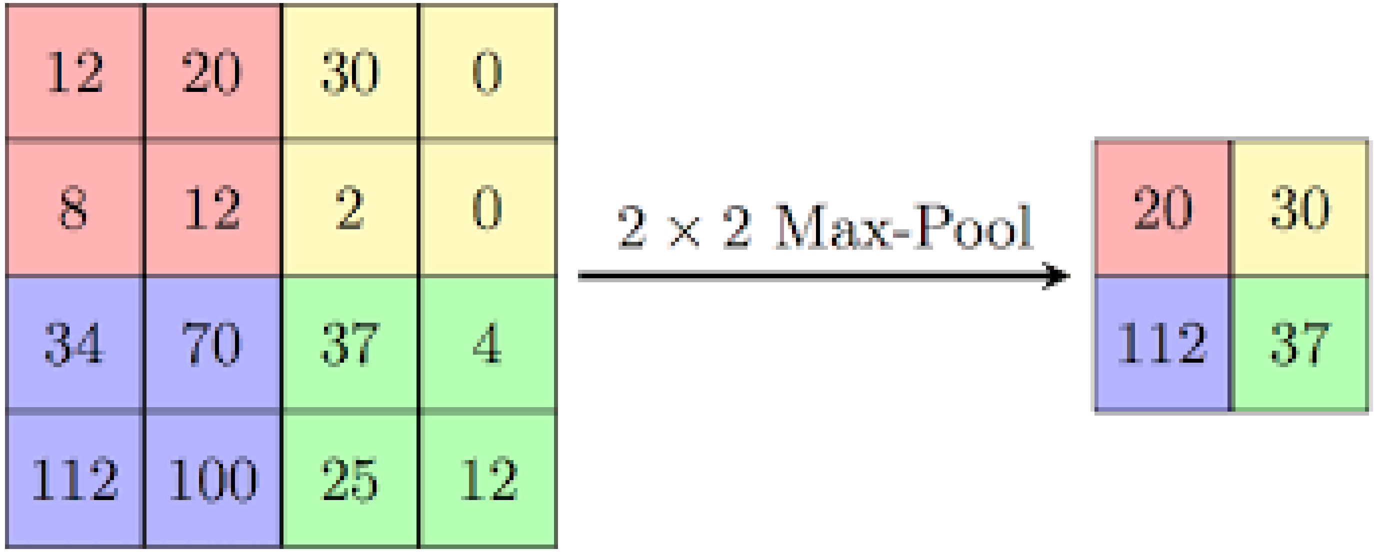


https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

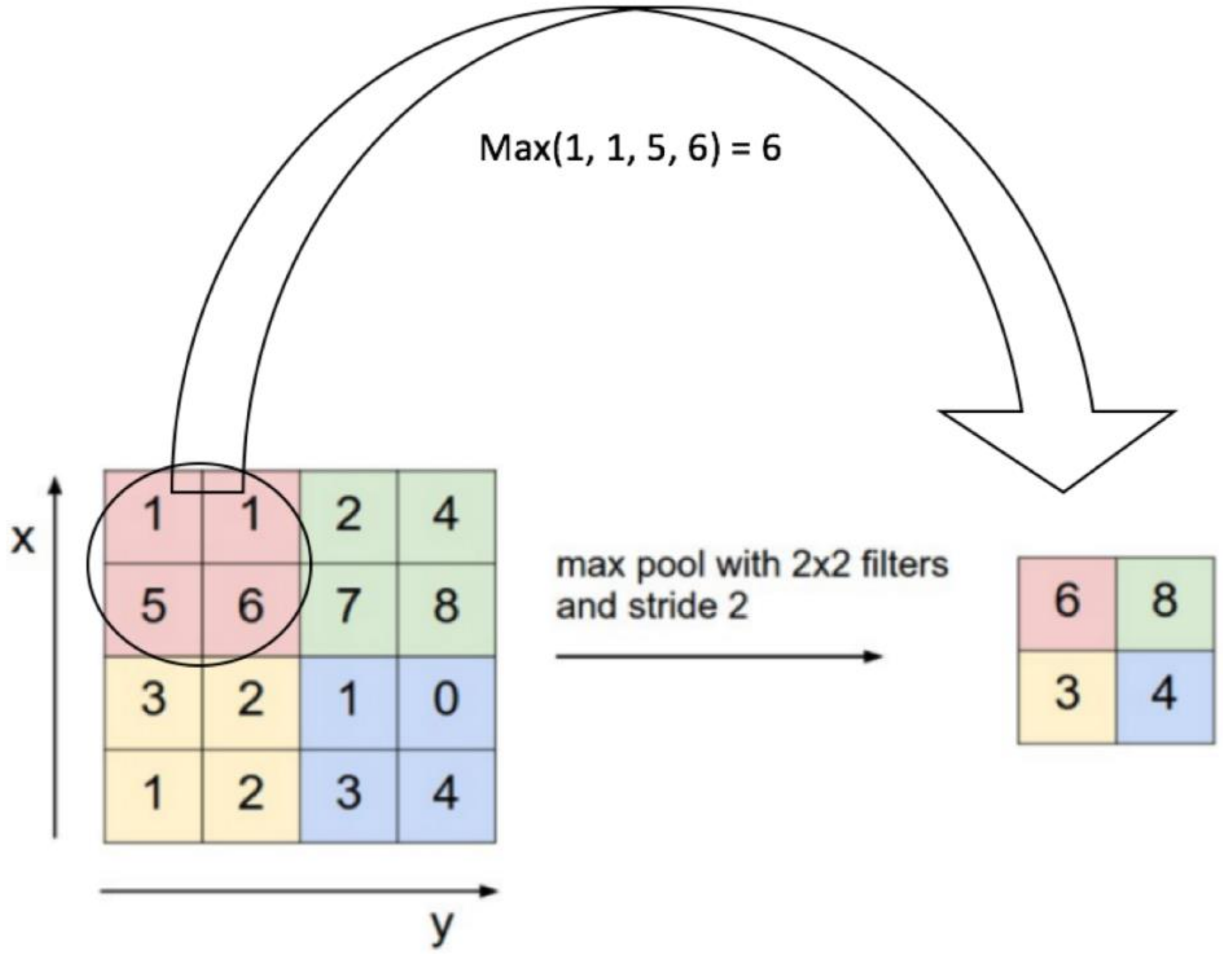
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Max Pooling

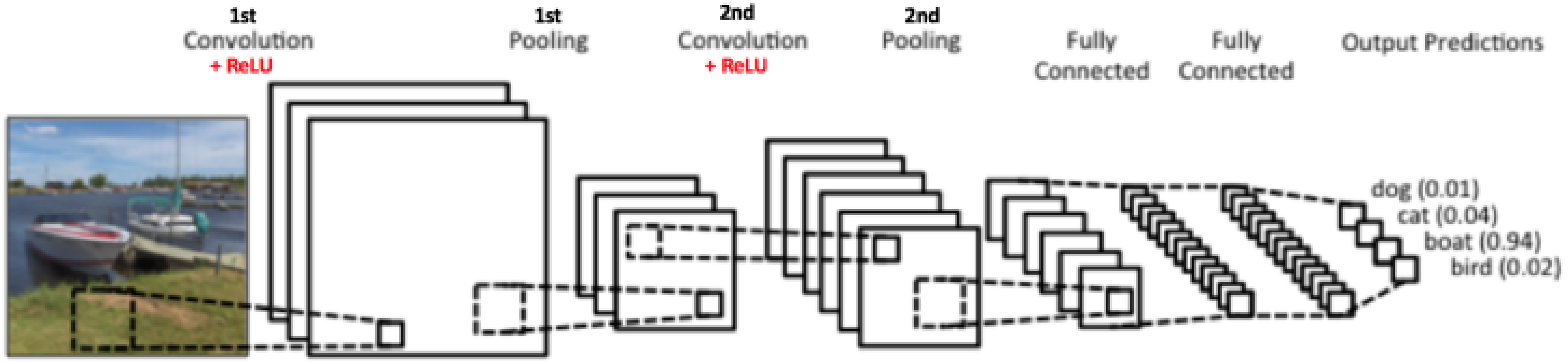
- Further reduce dimensionality
- Reduce parameters → reduce training time
- Summarizes features



Max Pool

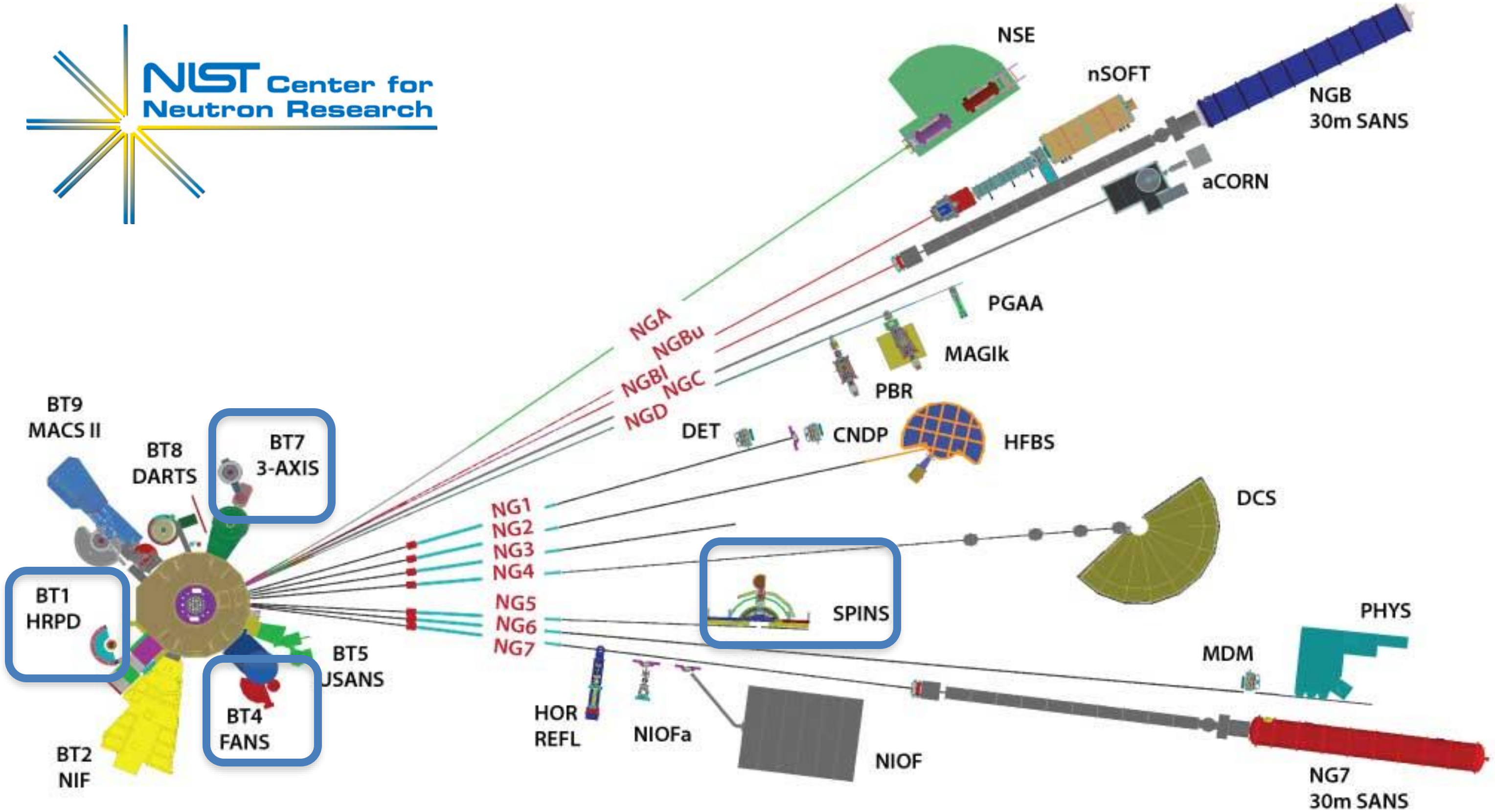


Rectified Feature Map

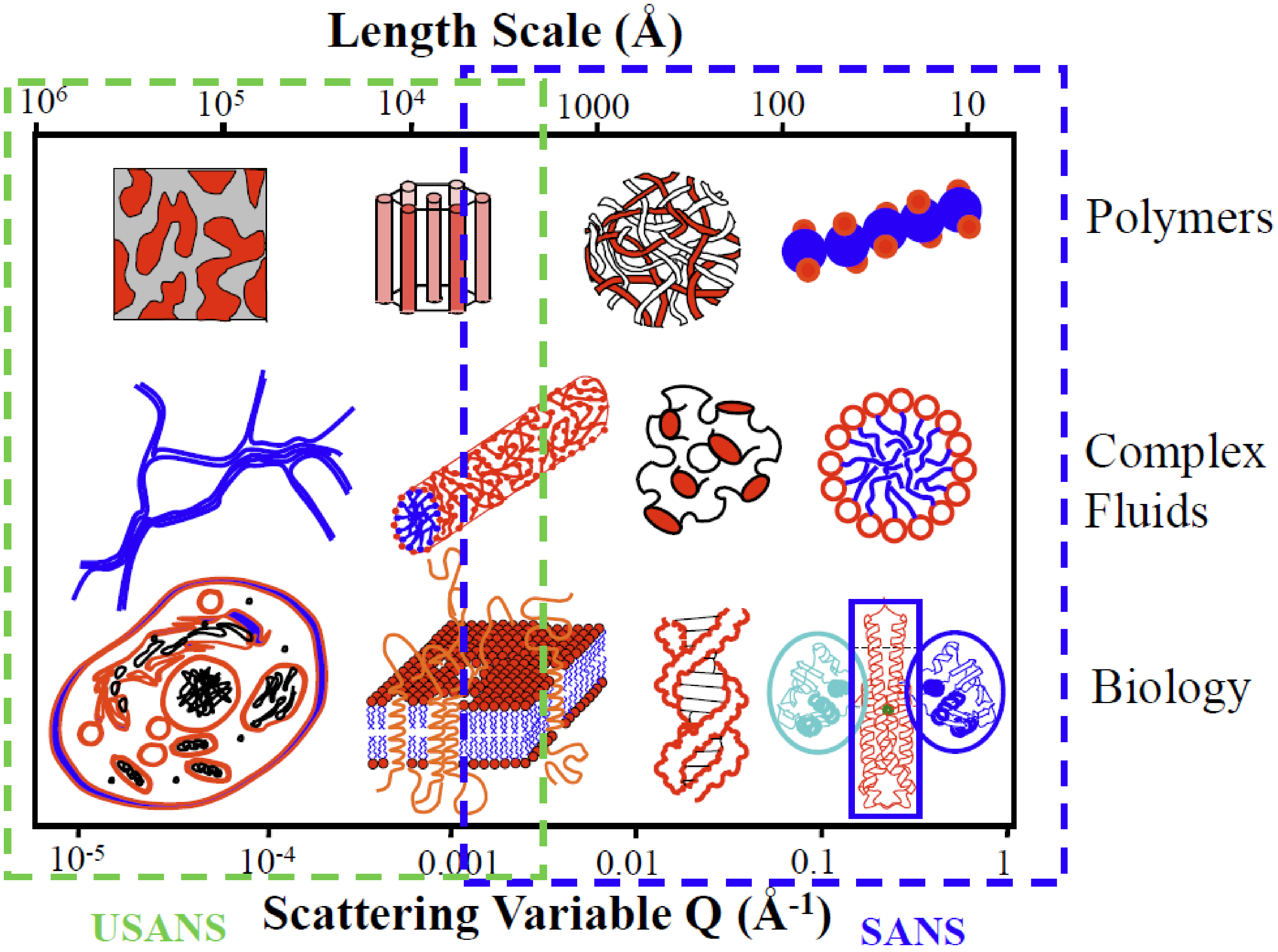

1st
Convolution
+ ReLU
1st
Pooling
2nd
Convolution
+ ReLU
2nd
Pooling
Fully
Connected
Fully
Connected
Output Predictions
dog (0.01)
cat (0.04)
boat (0.94)
bird (0.02)

NIST Center for Neutron Research

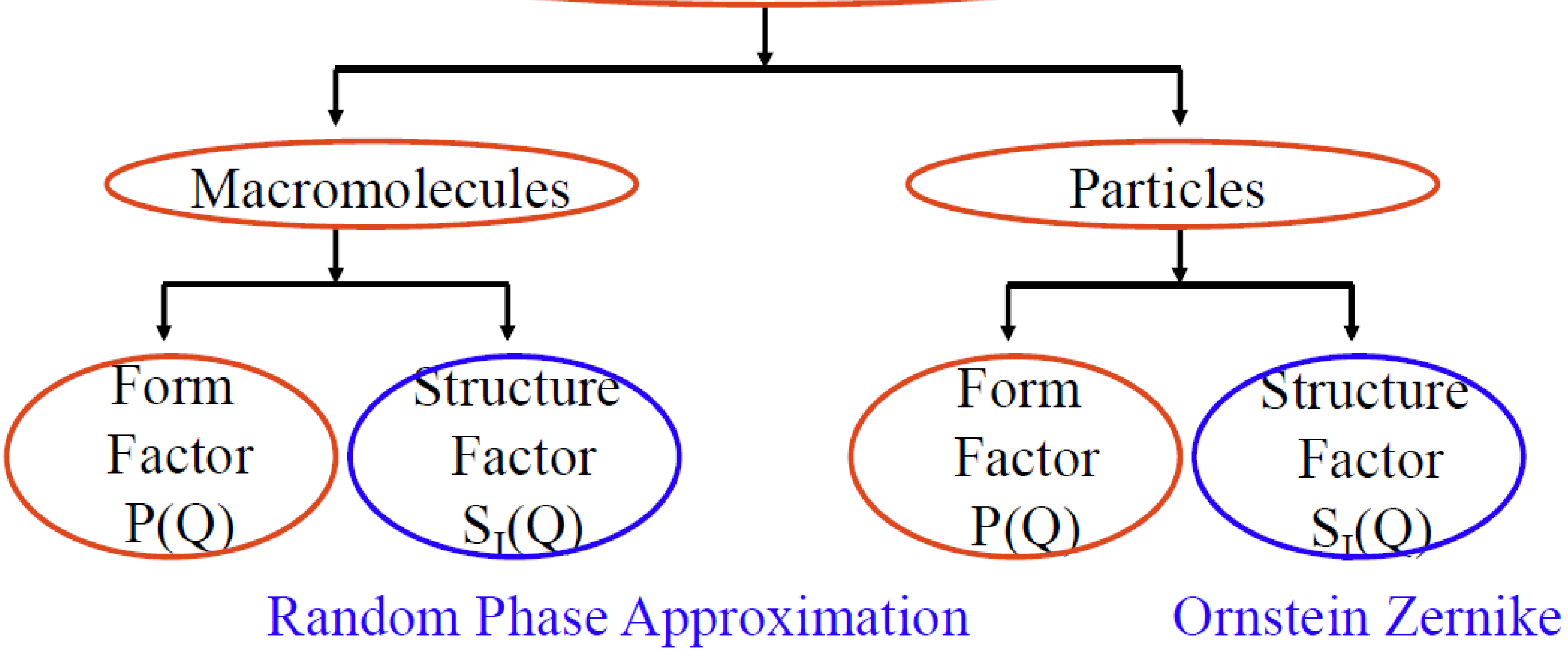
NIST Center for Neutron Research



Nanoscale Structures



SANS Models

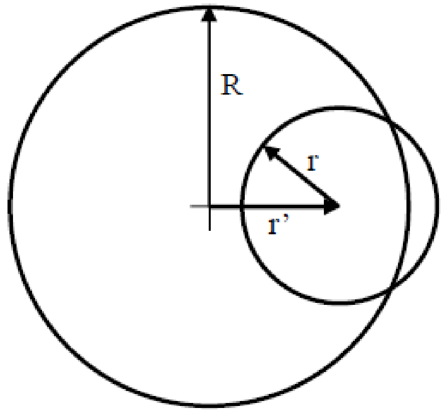


$$\underbrace{\frac{d\Sigma(Q)}{d\Omega}}_{\text{cross section}} = \underbrace{\phi_A}_{\text{volume fraction}} \underbrace{(\rho_A - \rho_B)^2}_{\text{contrast factor}} \underbrace{V_A}_{\text{particle volume}} \underbrace{P(Q)}_{\text{form factor}} \underbrace{S_I(Q)}_{\text{structure factor}}$$

Fourier Transform

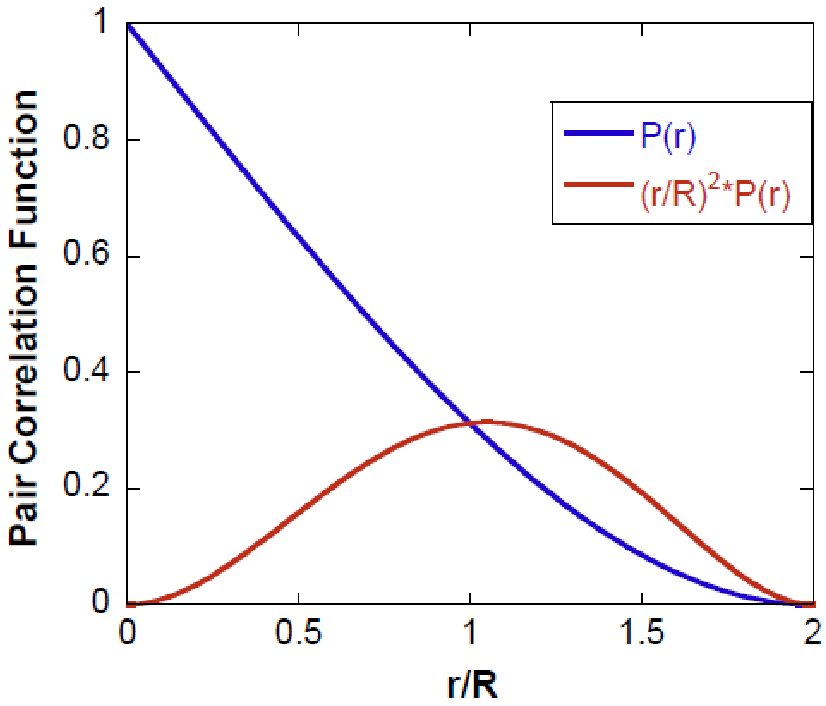
Density-density correlation function:

$$P(Q)=\frac{<n(-Q)n(Q)>}{n^2}=\int d\vec{r}\int d\vec{r}'\frac{<n(r)n(r')>}{n^2}\exp[i\vec{Q}.(\vec{r}'-\vec{r})]$$



Fourier transform:

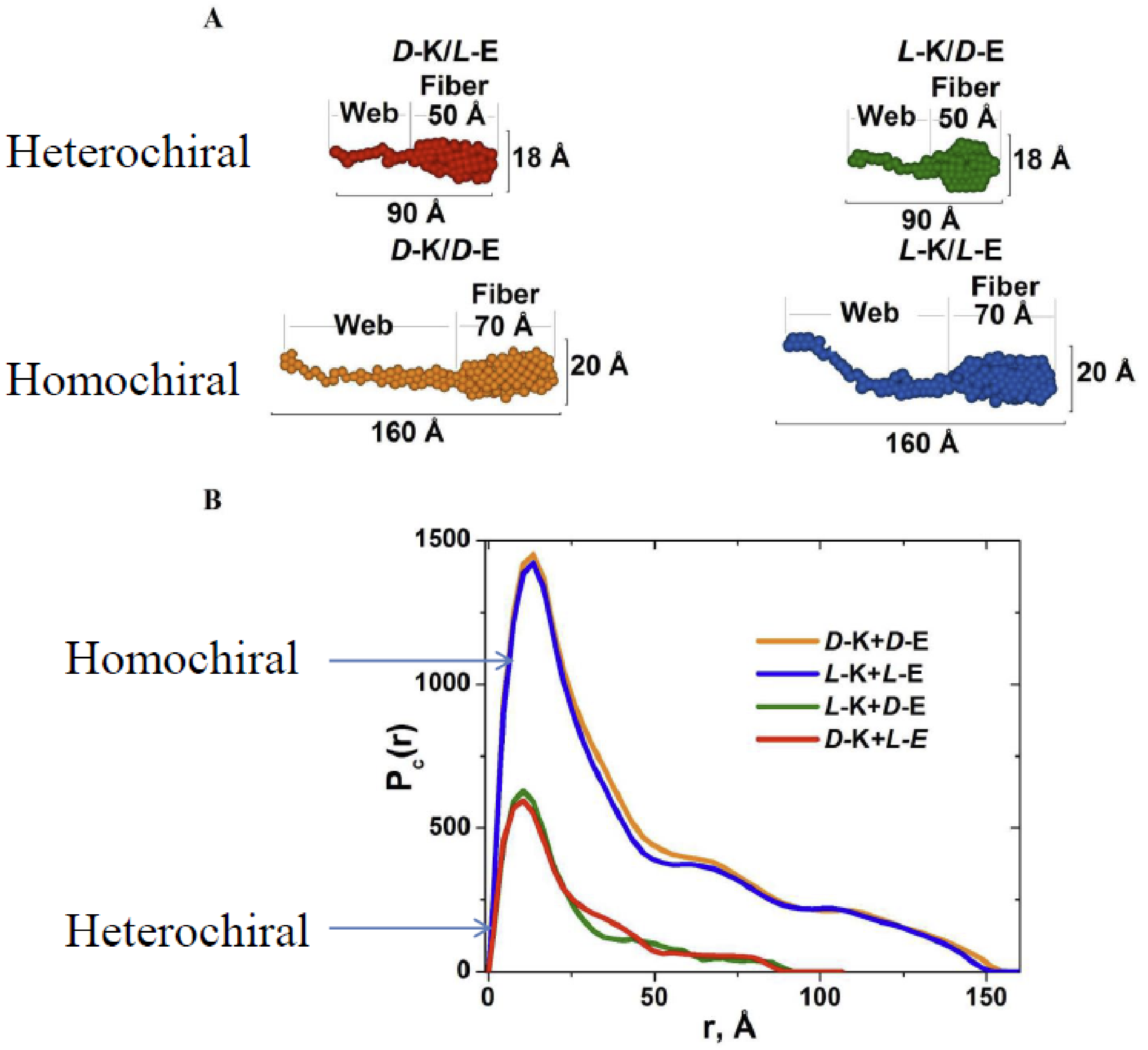
$$P(Q)=\int d^3r\exp[-i\vec{Q}.\vec{r}]\,P(\vec{r})=\frac{1}{V_P}\int_0^\infty dr\,4\pi r^2\frac{\sin(Qr)}{Qr}P(r)$$



Radial pair correlation function:

$$P(r)=1-\frac{3}{4}\left(\frac{r}{R}\right)+\frac{1}{16}\left(\frac{r}{R}\right)^3$$

Shape Reconstruction and Inverse Fourier Transform

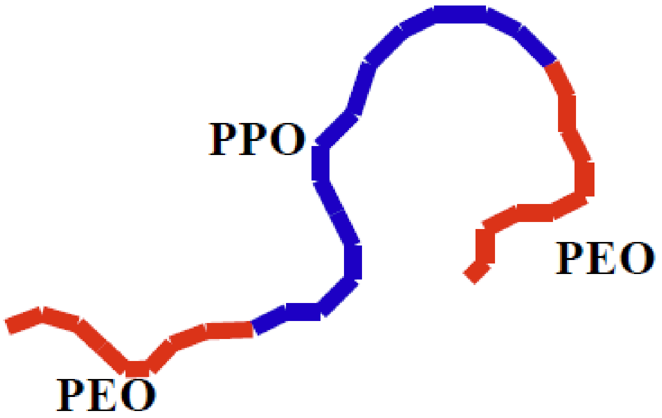


Shape Reconstruction
Fiber cross sections

Inverse Fourier
Transform

Pluronics

Dissolved Unimer
(low temperature)

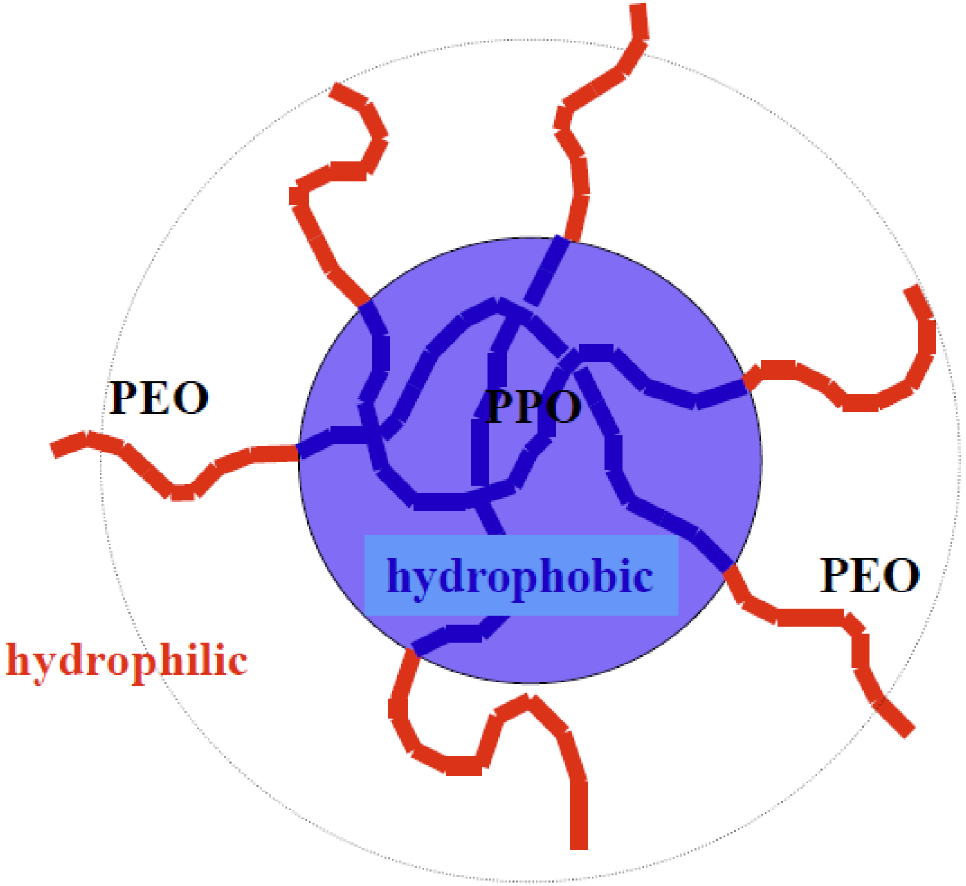


EO: $-CH_2CH_2-O-$

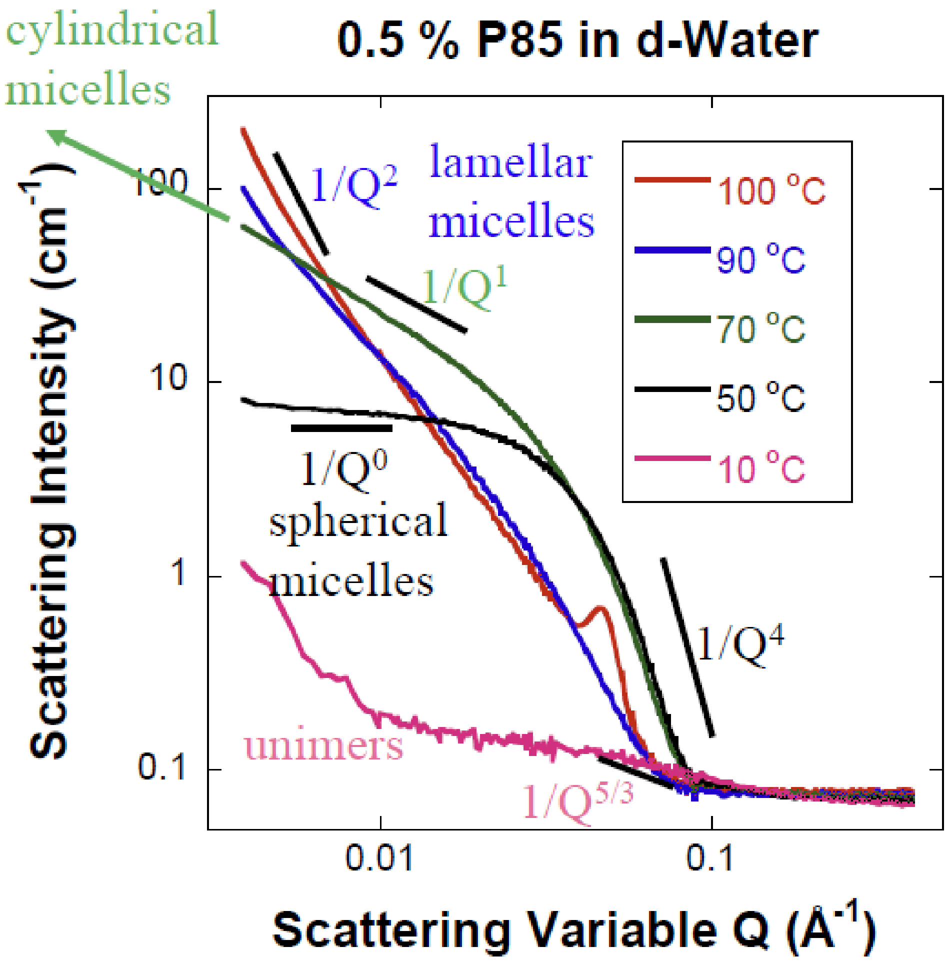
PO: $-CH(CH_3)CH_2-O-$

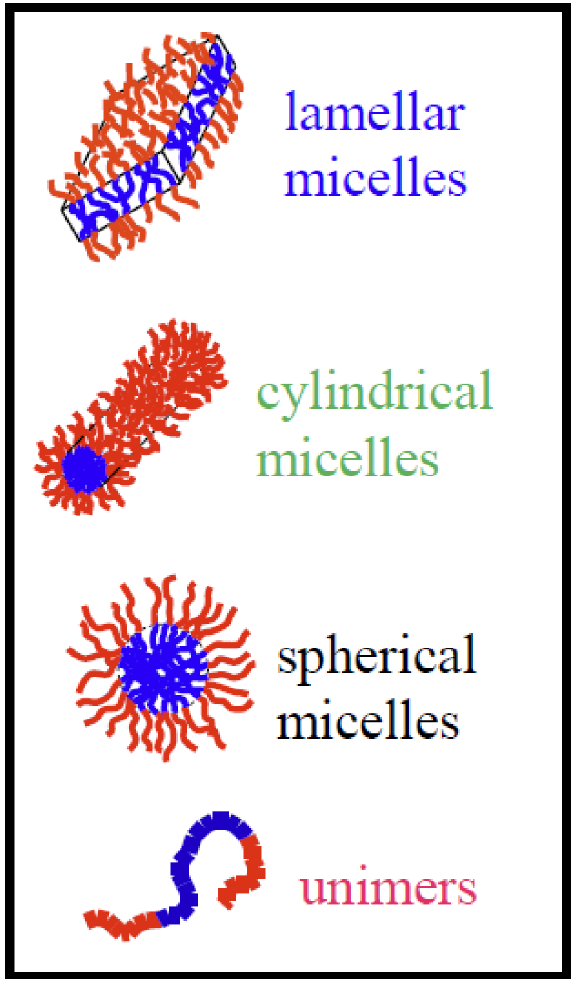
P85: $EO_{26}PO_{40}EO_{26}$

Formed Micelle
(high temperature)

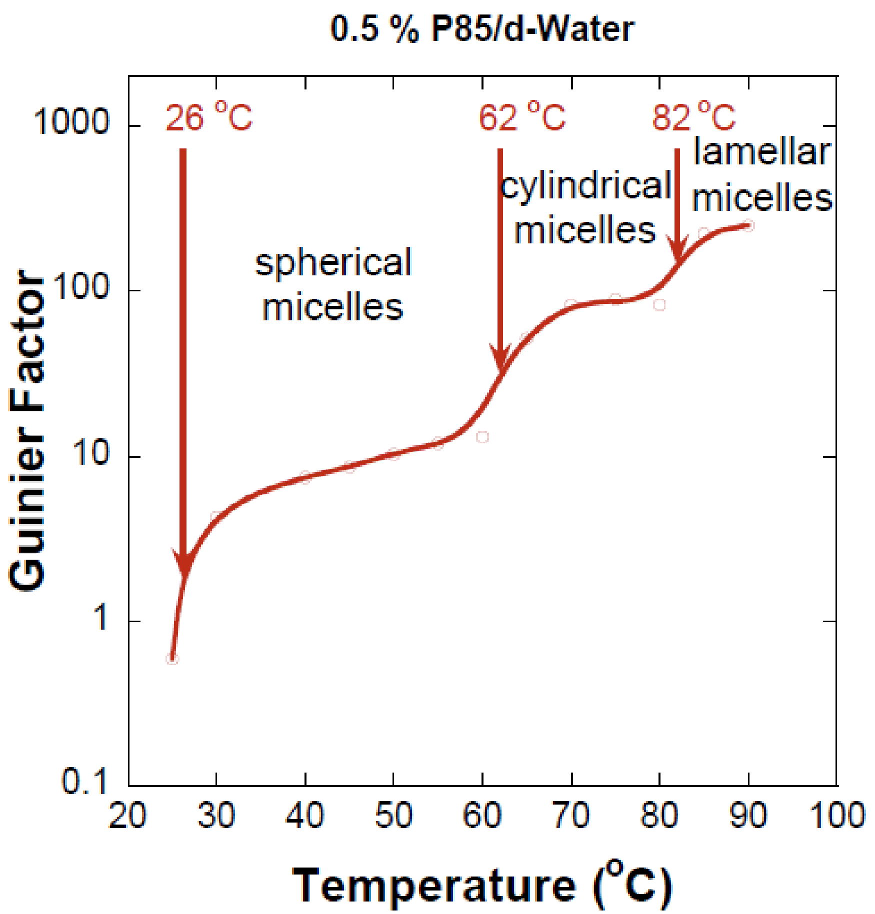


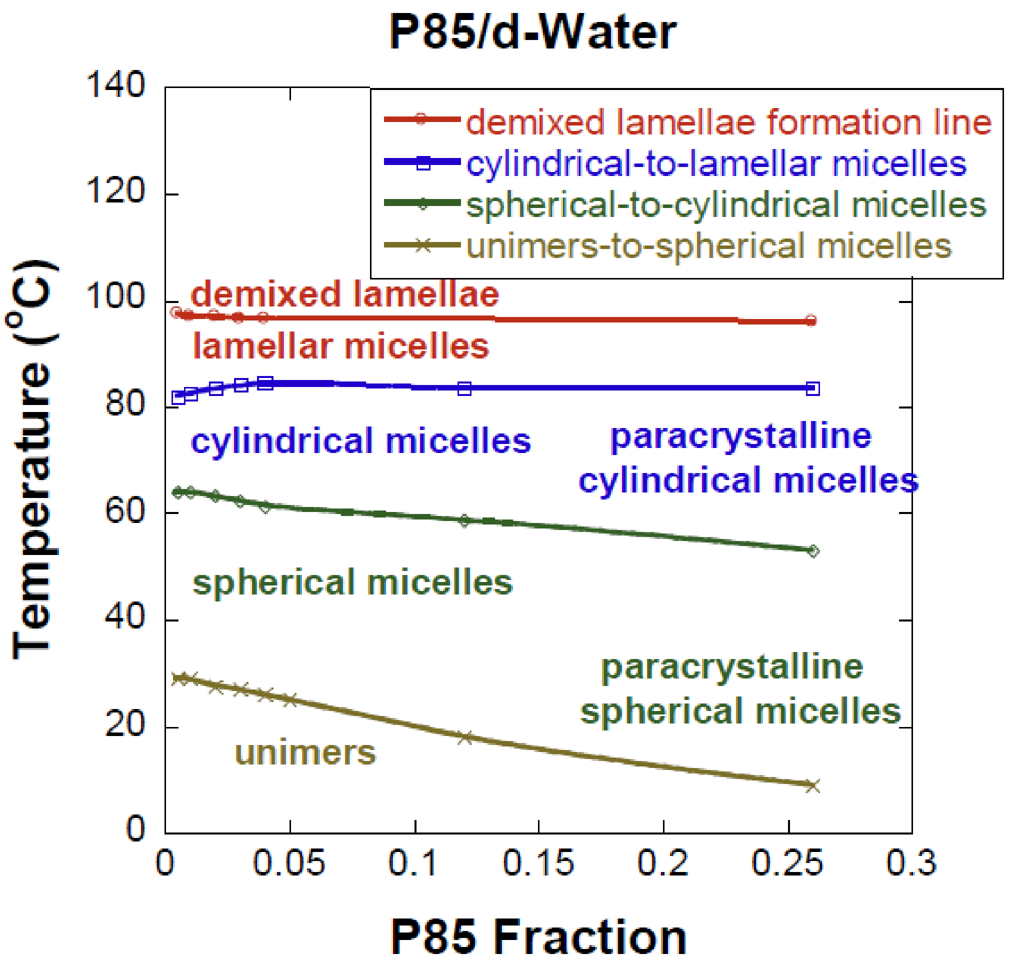
Pluronic Micelles





Phase Diagram





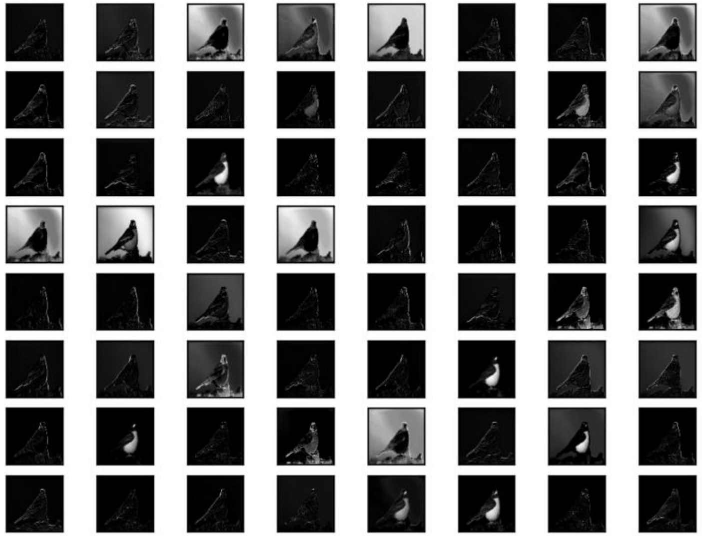
COLAB TIME

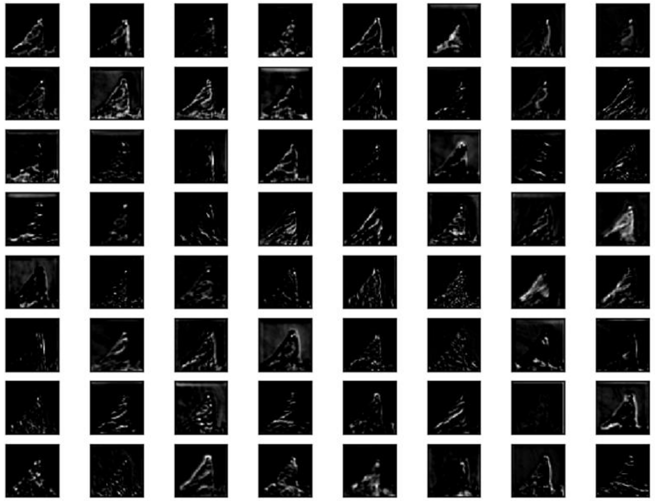
Where to find the Code

https://bit.ly/3R9UdYz

How Did We Do?

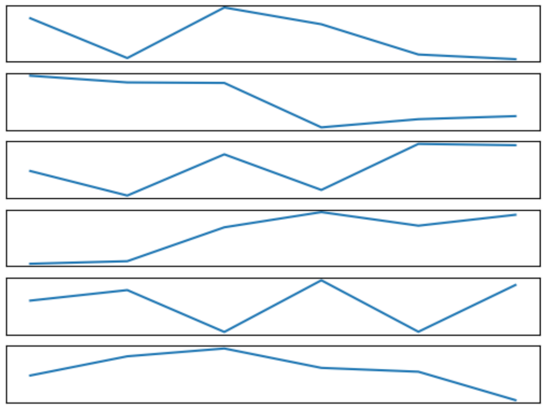
Filters and Feature Maps

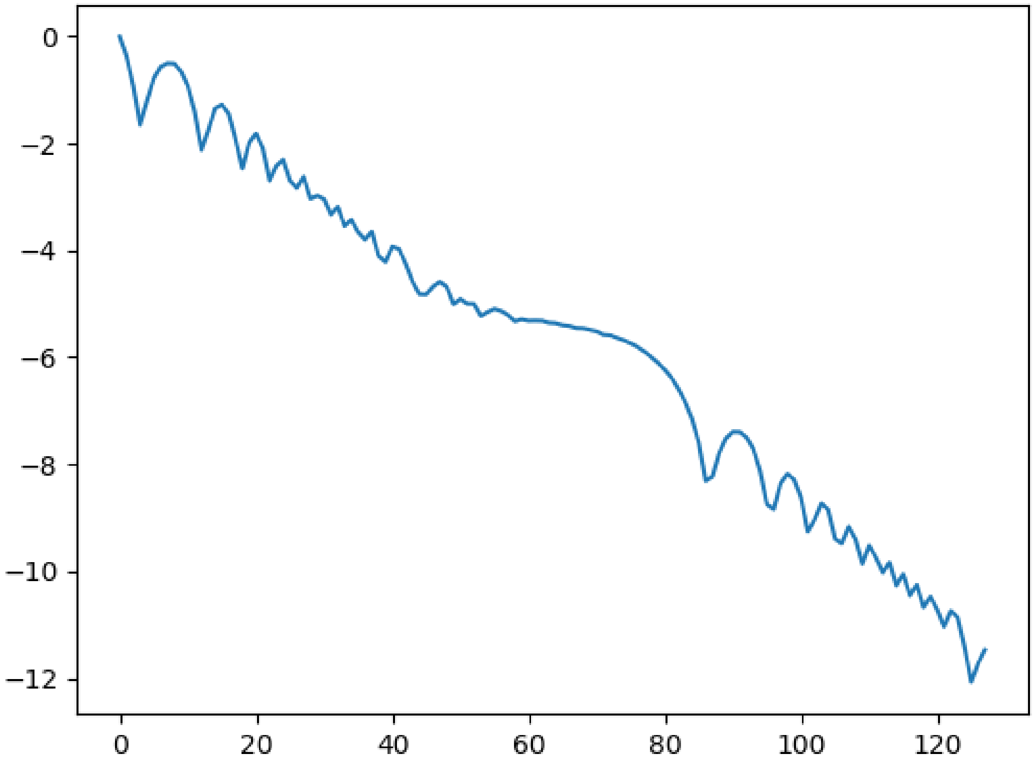


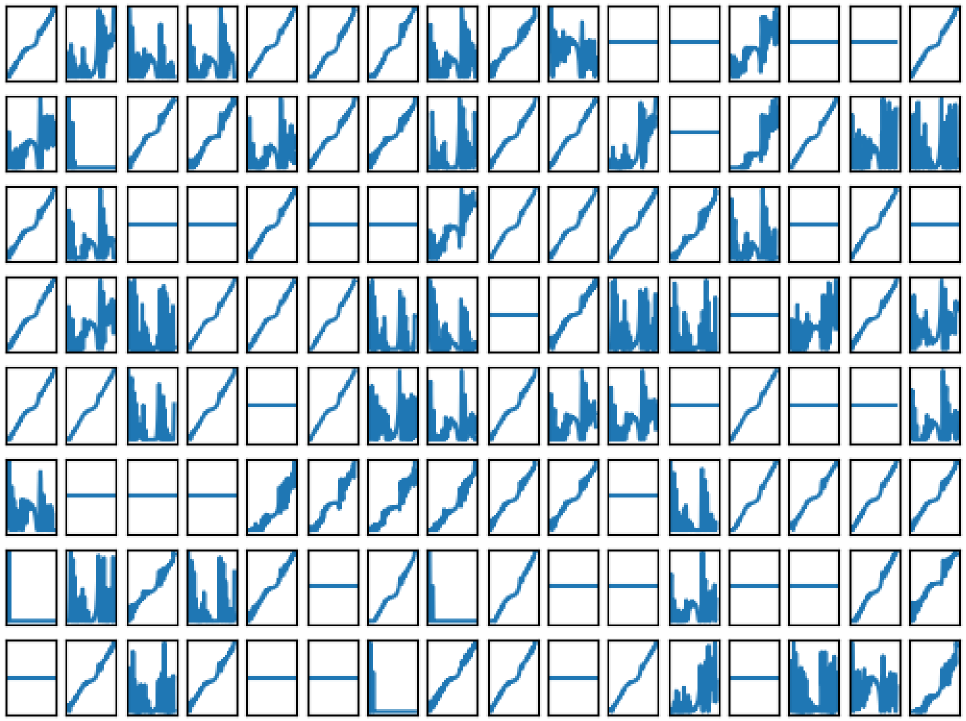


https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

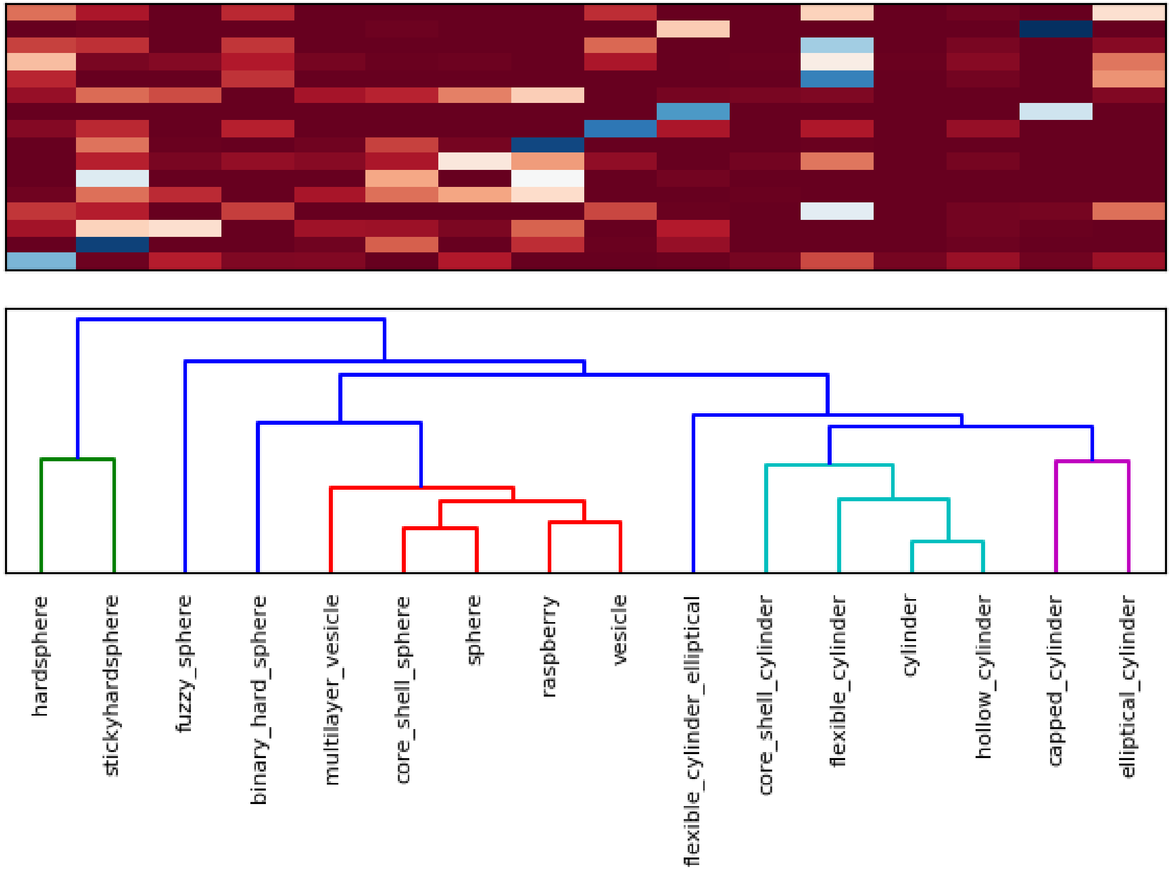
Filters and Feature Maps

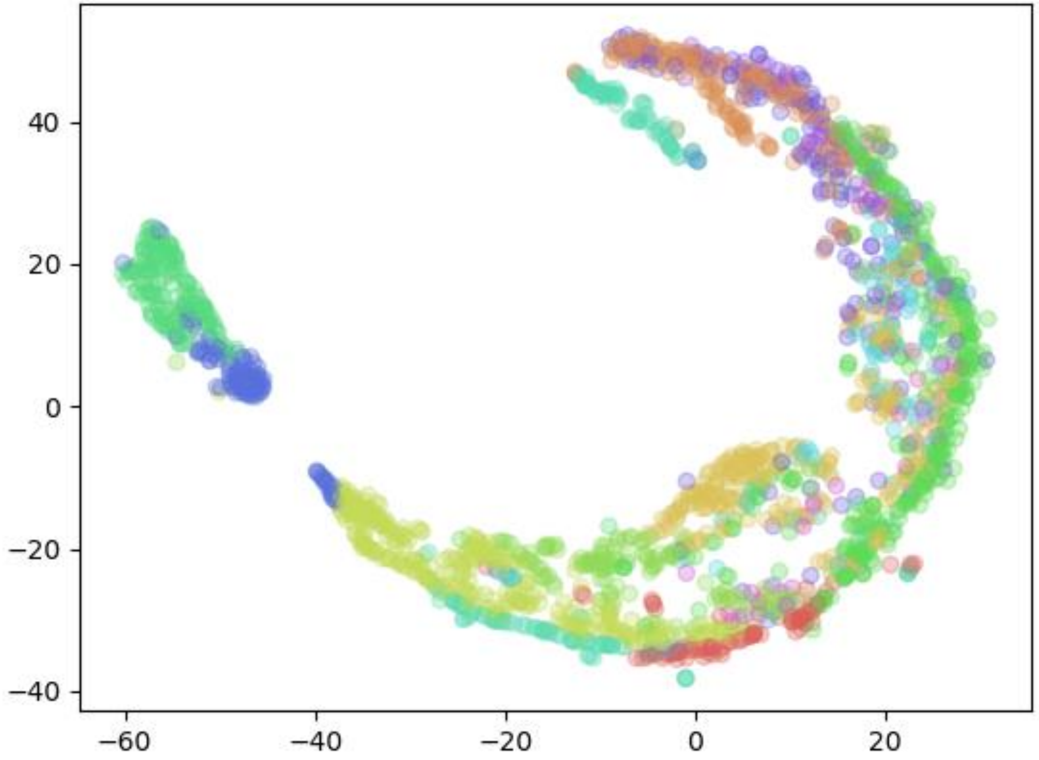




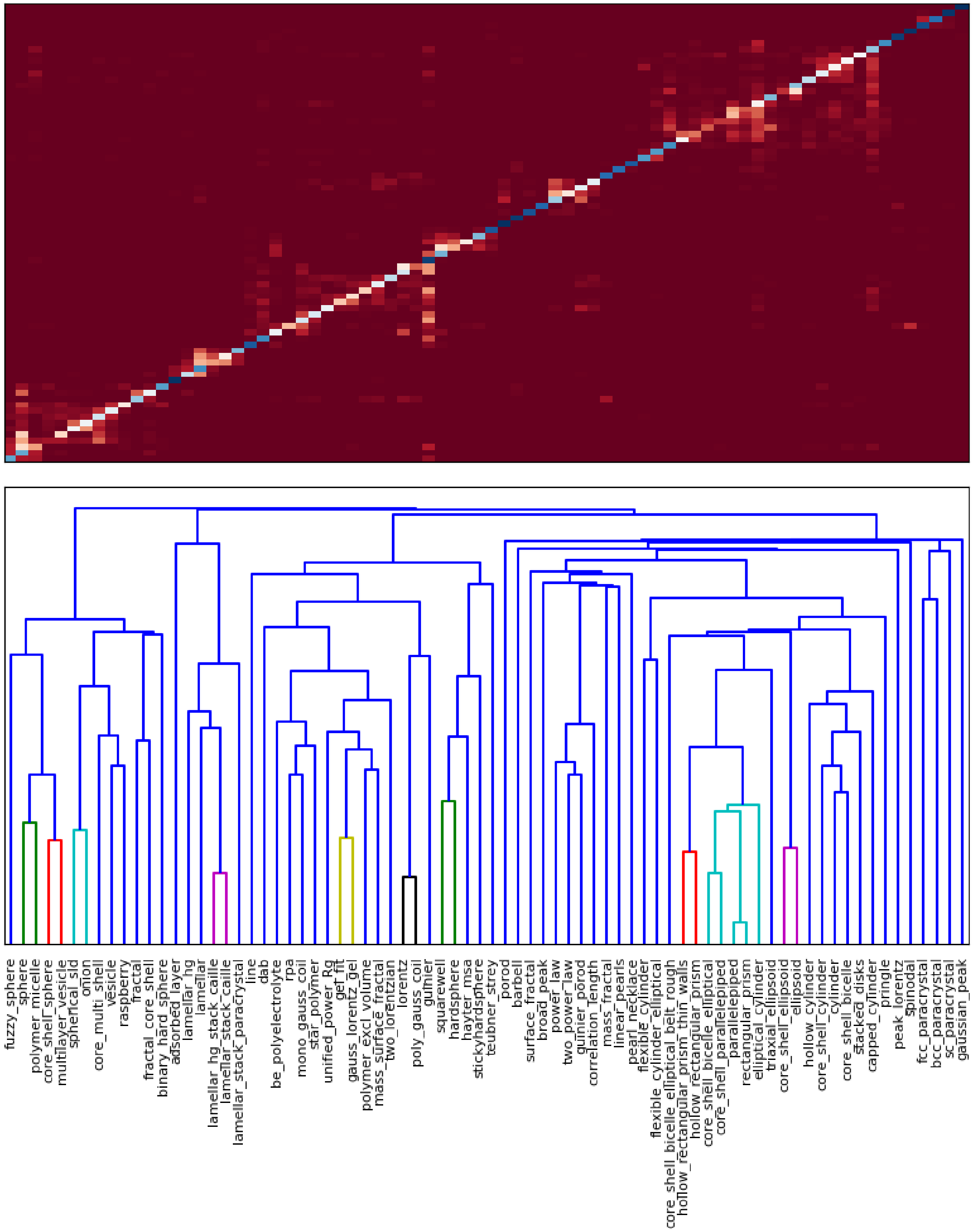


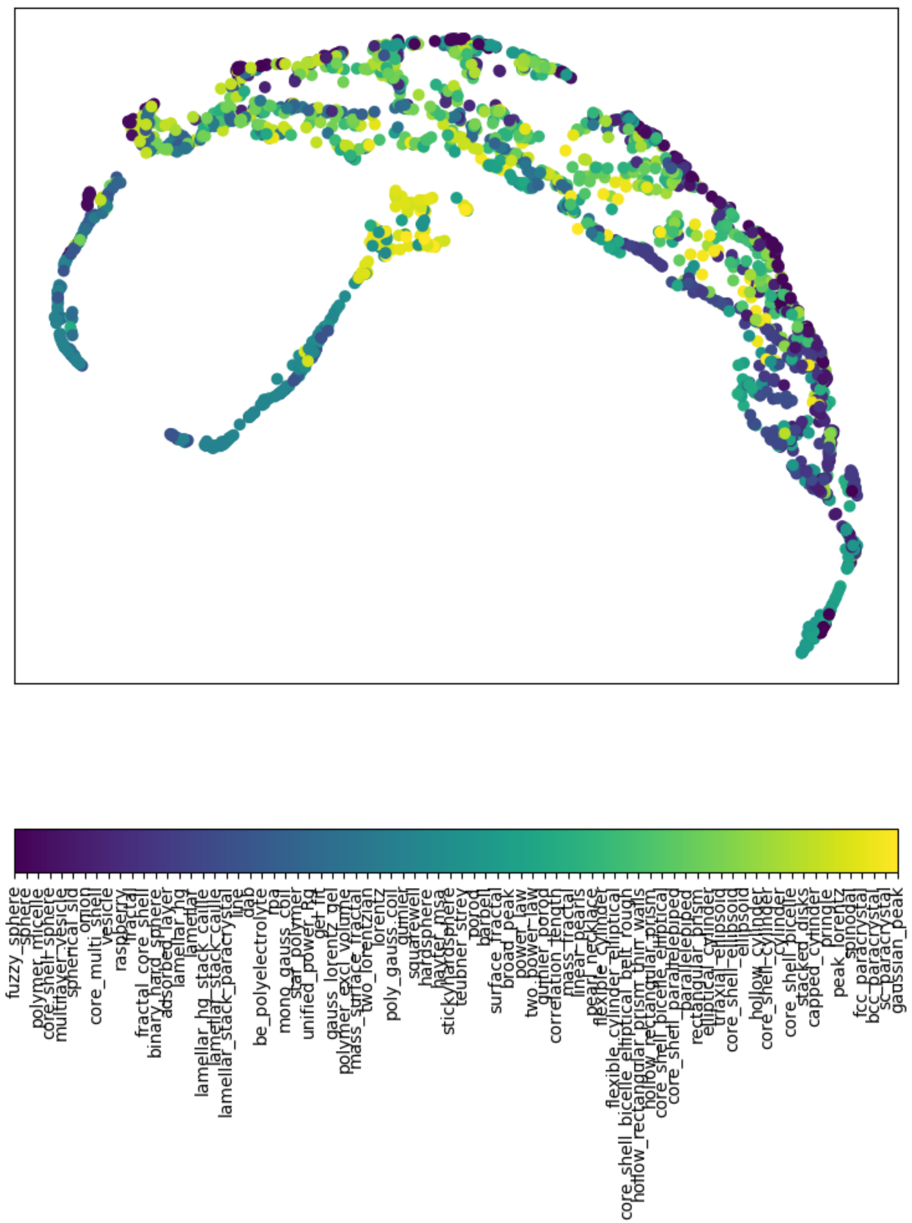
Results


hardsphere
stickyhardsphere
fuzzy_sphere
binary_hard_sphere
multilayer_vesicle
core_shell_sphere
sphere
raspberry
vesicle
flexible_cylinder_elliptical
core_shell_cylinder
flexible_cylinder
cylinder
hollow_cylinder
capped_cylinder
elliptical_cylinder



40
20
0
−20
−40
−60
−40
−20
0
20


fuzzy_sphere
sphere
polymer_micelle
core_shell_sphere
multilayer_vesicle
spherical_sld
onion
core_multi_shell
vesicle
raspberry
fractal
fractal_core_shell
binary_hard_sphere
adsorbed_layer
lamellar_hg
lamellar
lamellar_hg_stack_caille
lamellar_stack_caille
lamellar_stack_paracrystal
line
dab
be_polyelectrolyte
rpa
mono_gauss_coil
star_polymer
unified_power_Rg
gel_fit
gauss_lorentz_gel
polymer_excl_volume
mass_surface_fractal
two_lorentzian
lorentz
poly_gauss_coil
guinier
squarewell
hardsphere
hayter_msa
stickyhardsphere
teubner_strey
porod
barbell
surface_fractal
broad_peak
power_law
two_power_law
guinier_porod
correlation_length
mass_fractal
linear_pearls
pearl_necklace
flexible_cylinder
flexible_cylinder_elliptical
core_shell_bicelle_elliptical_belt_rough
hollow_rectangular_prism_thin_walls
hollow_rectangular_prism
core_shell_bicelle_elliptical
core_shell_parallelepiped
parallelepiped
rectangular_prism
elliptical_cylinder
triaxial_ellipsoid
core_shell_ellipsoid
ellipsoid
hollow_cylinder
core_shell_cylinder
cylinder
core_shell_bicelle
stacked_disks
capped_cylinder
pringle
peak_lorentz
spinodal
fcc_paracrystal
bcc_paracrystal
sc_paracrystal
gaussian_peak


fuzzy_sphere
sphere
polymer_micelle
core_shell_sphere
multilayer_vesicle
spherical_sld
onion
core_multi_shell
vesicle
raspberry
fractal
fractal_core_shell
binary_hard_sphere
adsorbed_layer
lamellar_hg
lamellar
lamellar_hg_stack_caille
lamellar_stack_caille
lamellar_stack_paracrystal
line
dab
be_polyelectrolyte
rpa
mono_gauss_coil
star_polymer
unified_power_Rg
gel_fit
gauss_lorentz_gel
polymer_excl_volume
mass_surface_fractal
two_lorentzian
lorentz
poly_gauss_coil
guinier
squarewell
hardsphere
hayter_msa
stickyhardsphere
teubner_strey
porod
barbell
surface_fractal
broad_peak
power_law
two_power_law
guinier_porod
correlation_length
mass_fractal
linear_pearls
pearl_necklace
flexible_cylinder
flexible_cylinder_elliptical
core_shell_bicelle_elliptical_belt_rough
hollow_rectangular_prism_thin_walls
hollow_rectangular_prism
core_shell_bicelle_elliptical
core_shell_parallelepiped
parallelepiped
rectangular_prism
elliptical_cylinder
triaxial_ellipsoid
core_shell_ellipsoid
ellipsoid
hollow_cylinder
core_shell_cylinder
cylinder
core_shell_bicelle
stacked_disks
capped_cylinder
pringle
peak_lorentz
spinodal
fcc_paracrystal
bcc_paracrystal
sc_paracrystal
gaussian_peak

Questions