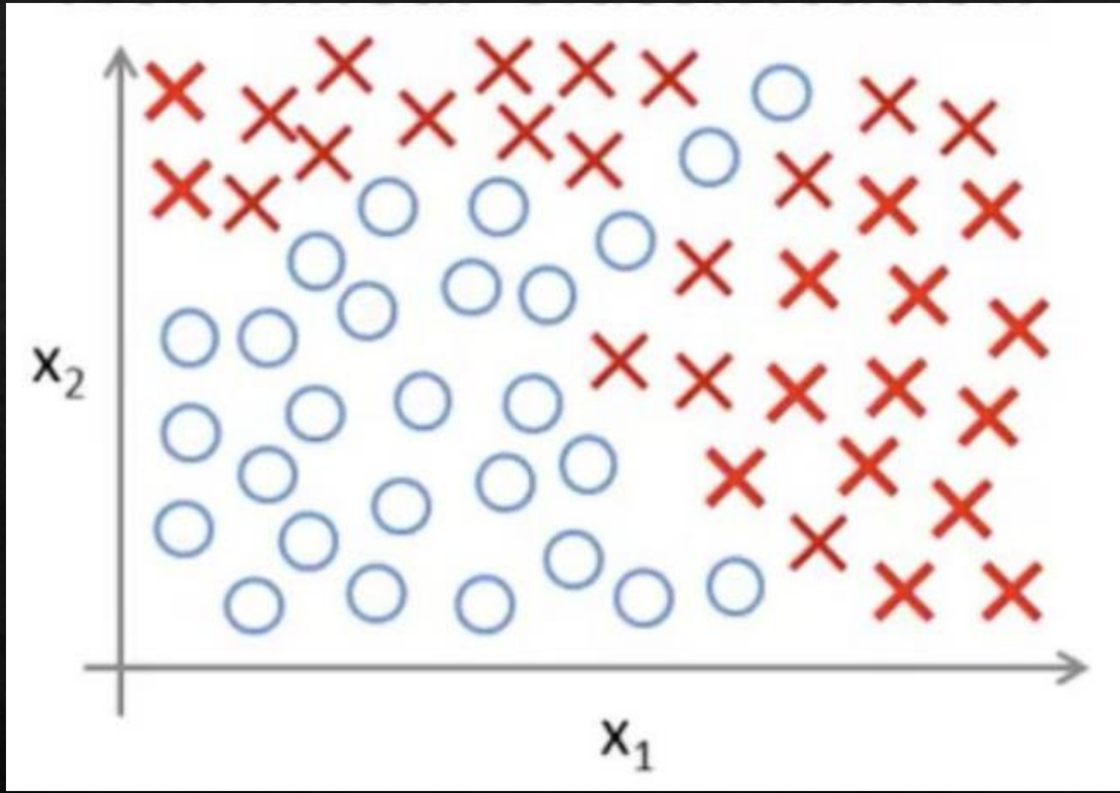
Sztuczne sieci neuronowe

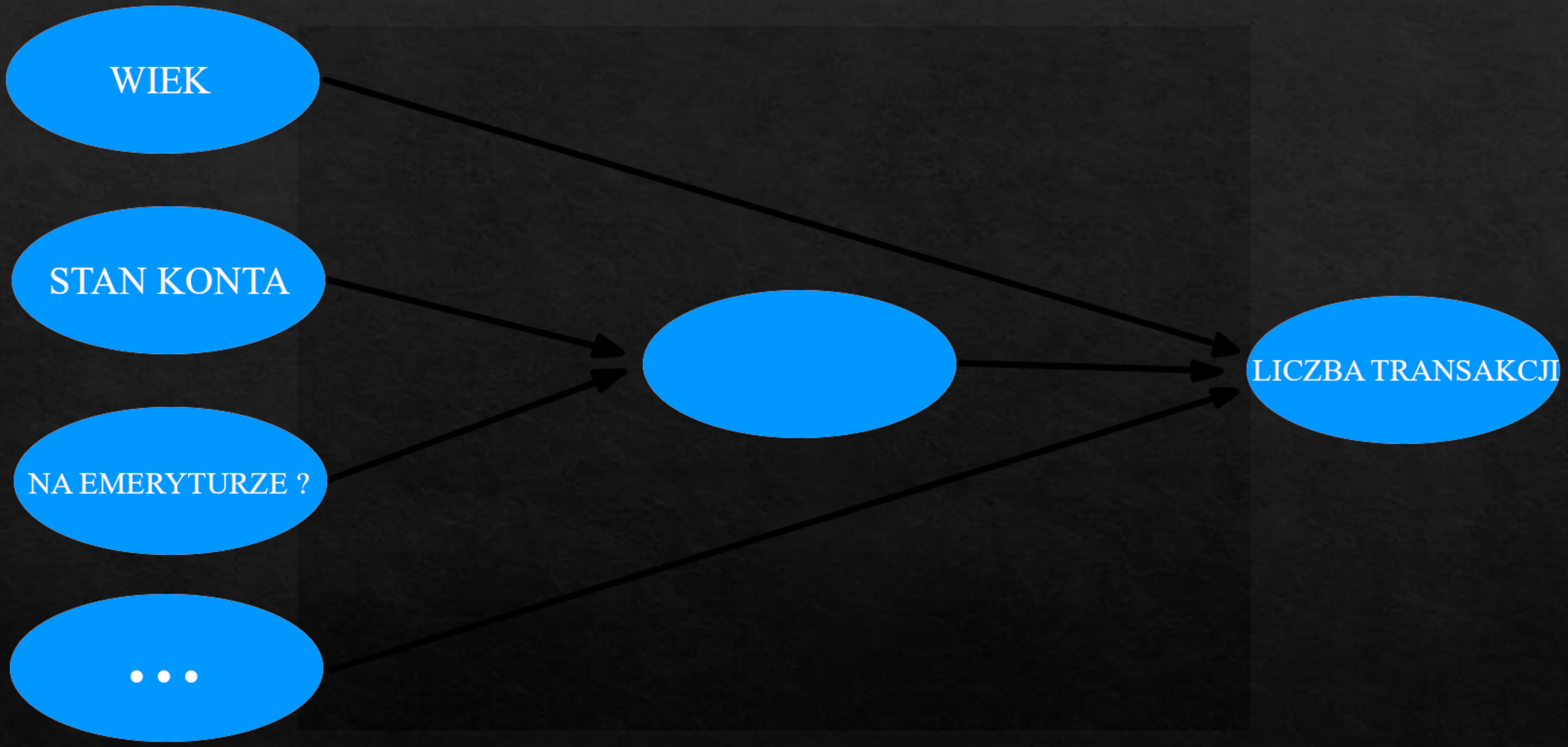
Wprowadzenie

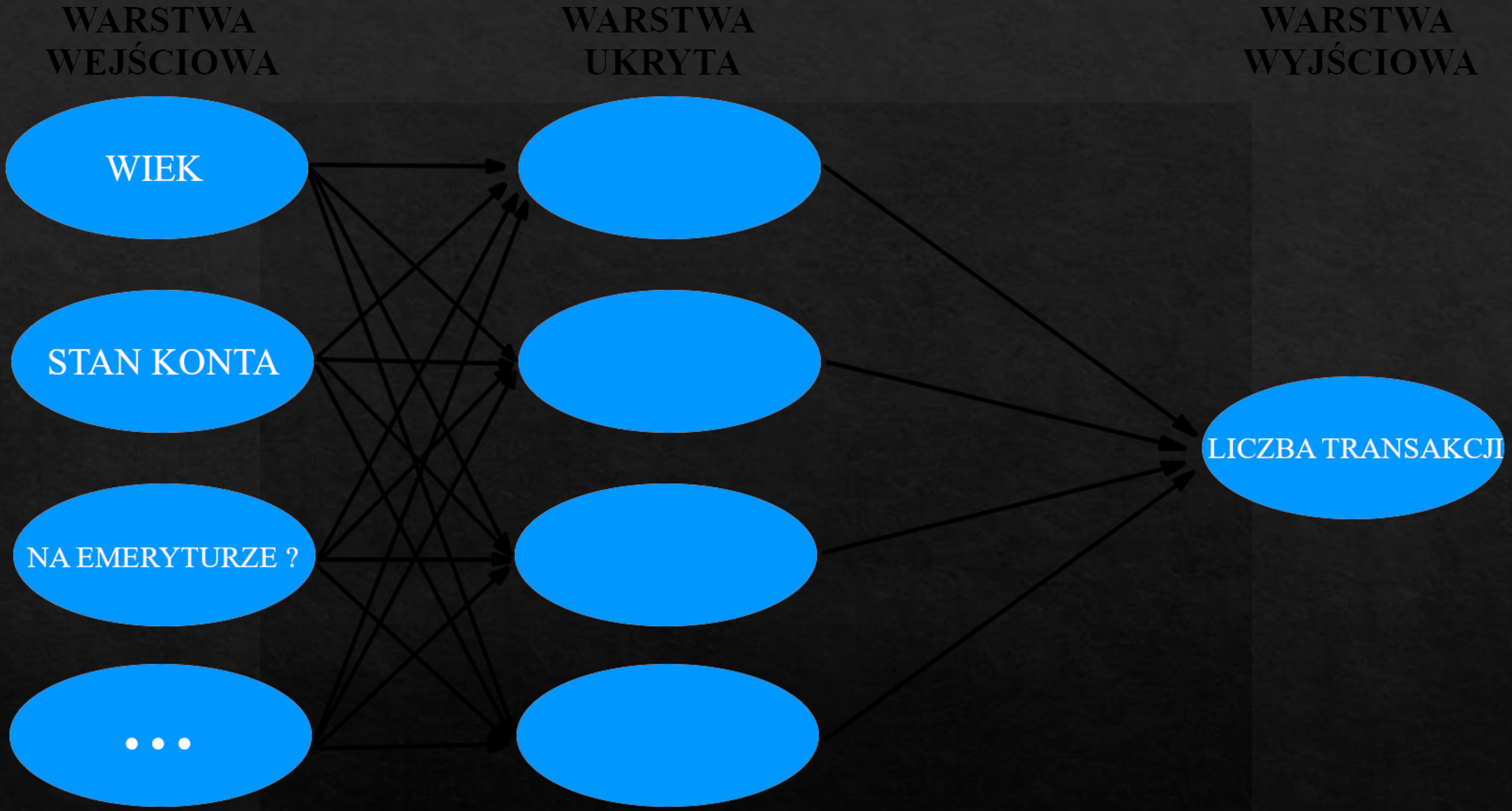
Plan na dzisiaj

- Funkcja kosztu (loss function)
- Regularyzacja
- Backpropagation czyli w jaki sposób sieci się uczą
- Optymalizacje
- Hiperparametry sieci
- Przeuczenie
- Strojenie modelu

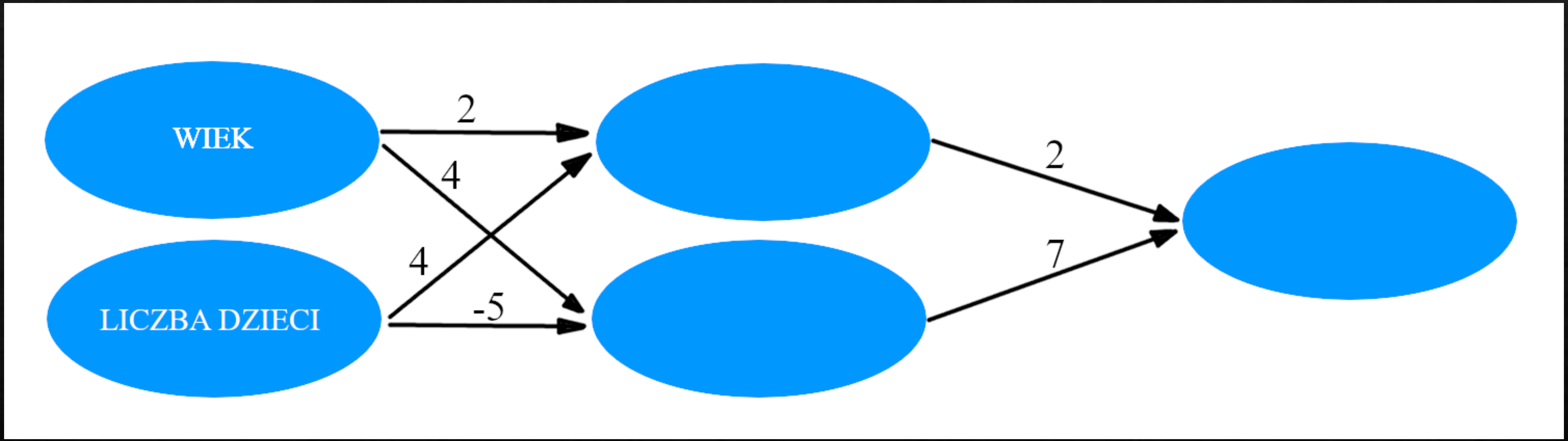

x_2
x_1

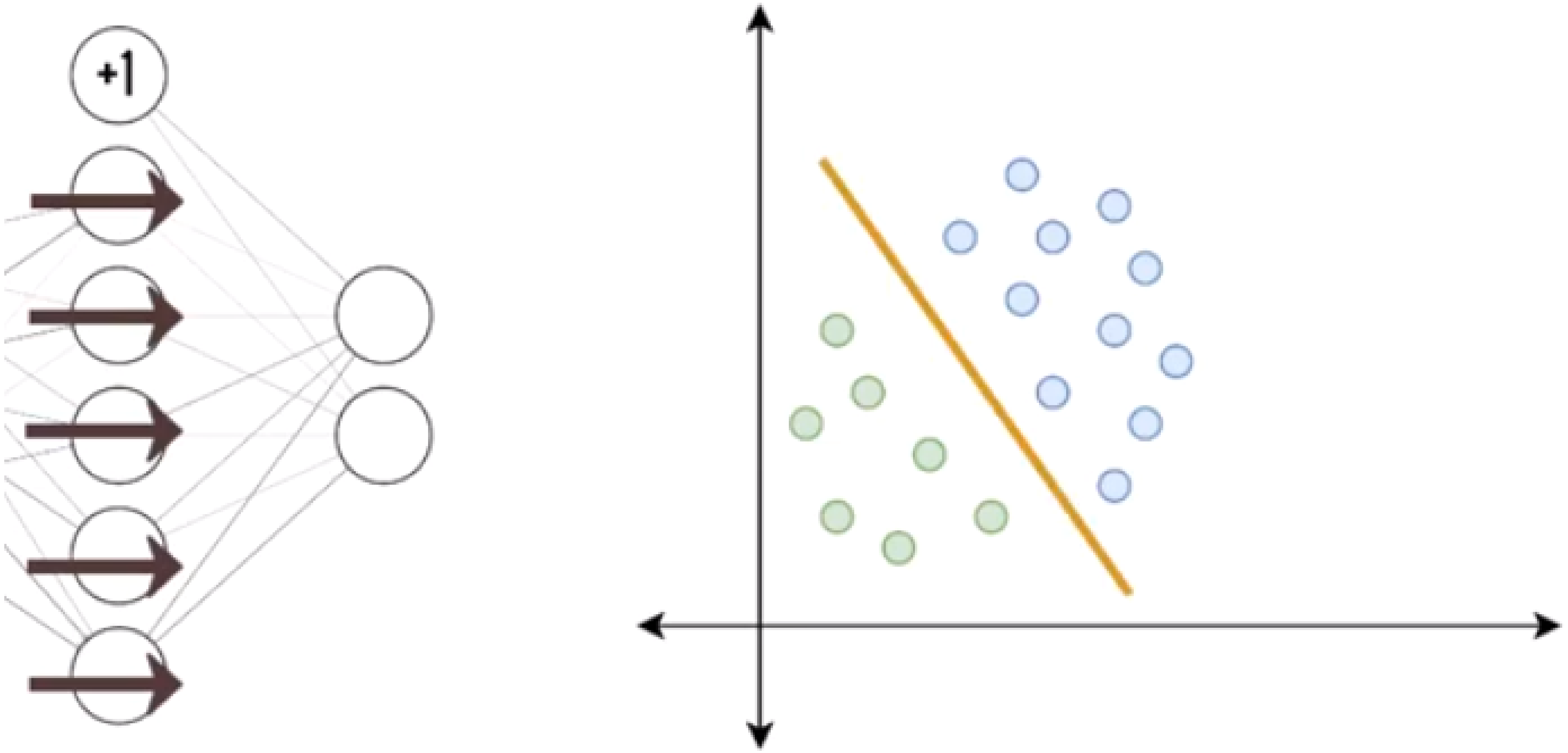

WIEK
STAN KONTA
NA EMERYTURZE ?
...
LICZBA TRANSAKCJI

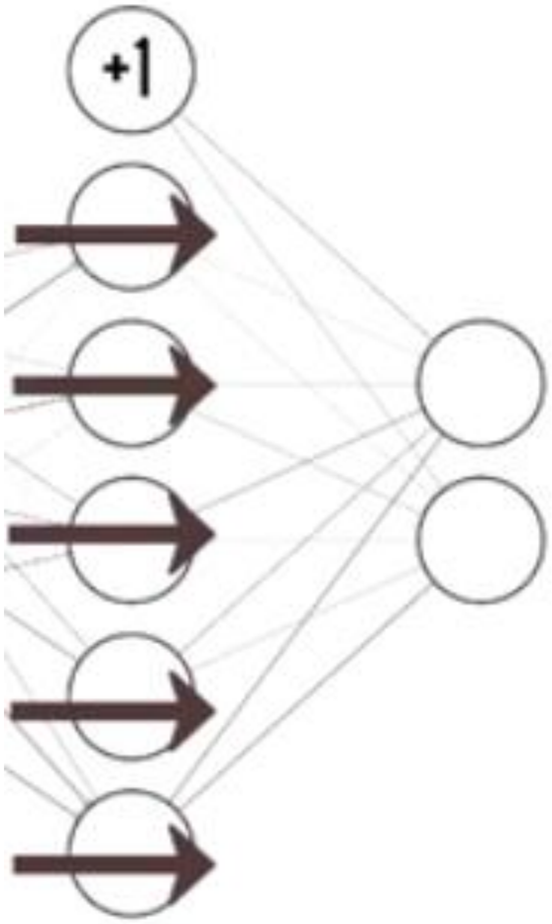

WIEK
STAN KONTA
NA EMERYTURZE ?
...
LICZBA TRANSAKCJI

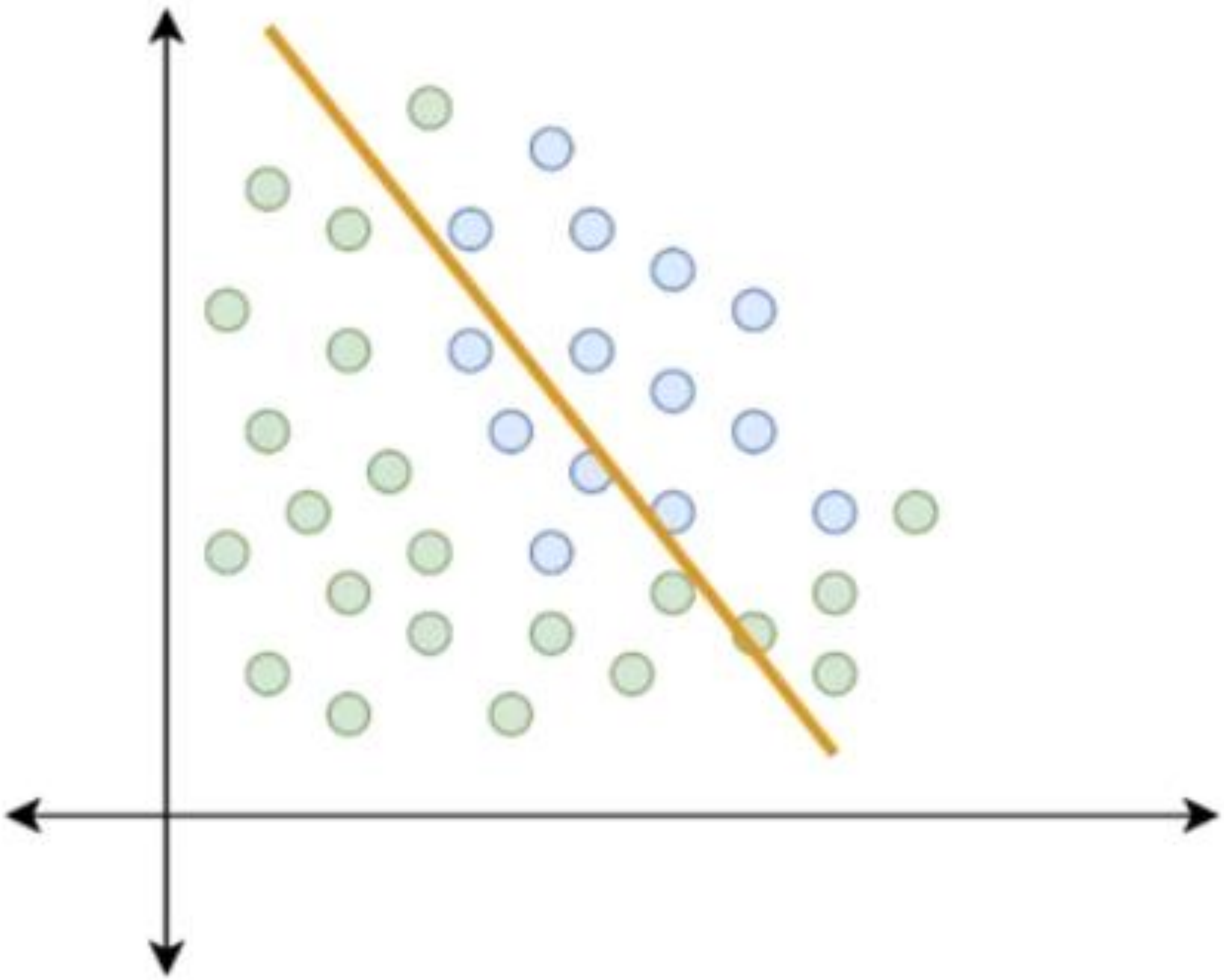

WARSTWA WEJŚCIOWA
WARSTWA UKRYTA
WARSTWA WYJŚCIOWA
WIEK
STAN KONTA
NA EMERYTURZE ?
• • •
LICZBA TRANSAKCJI

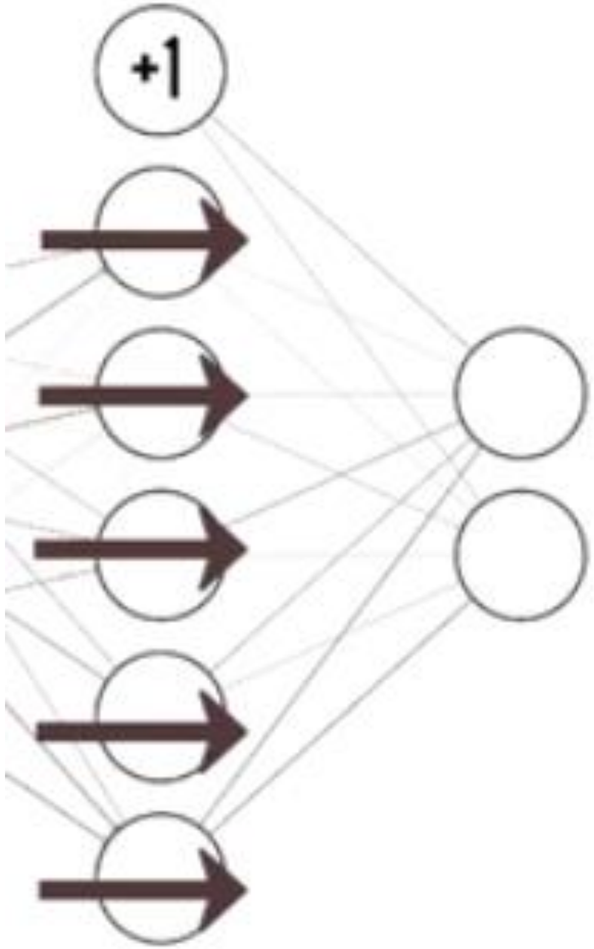
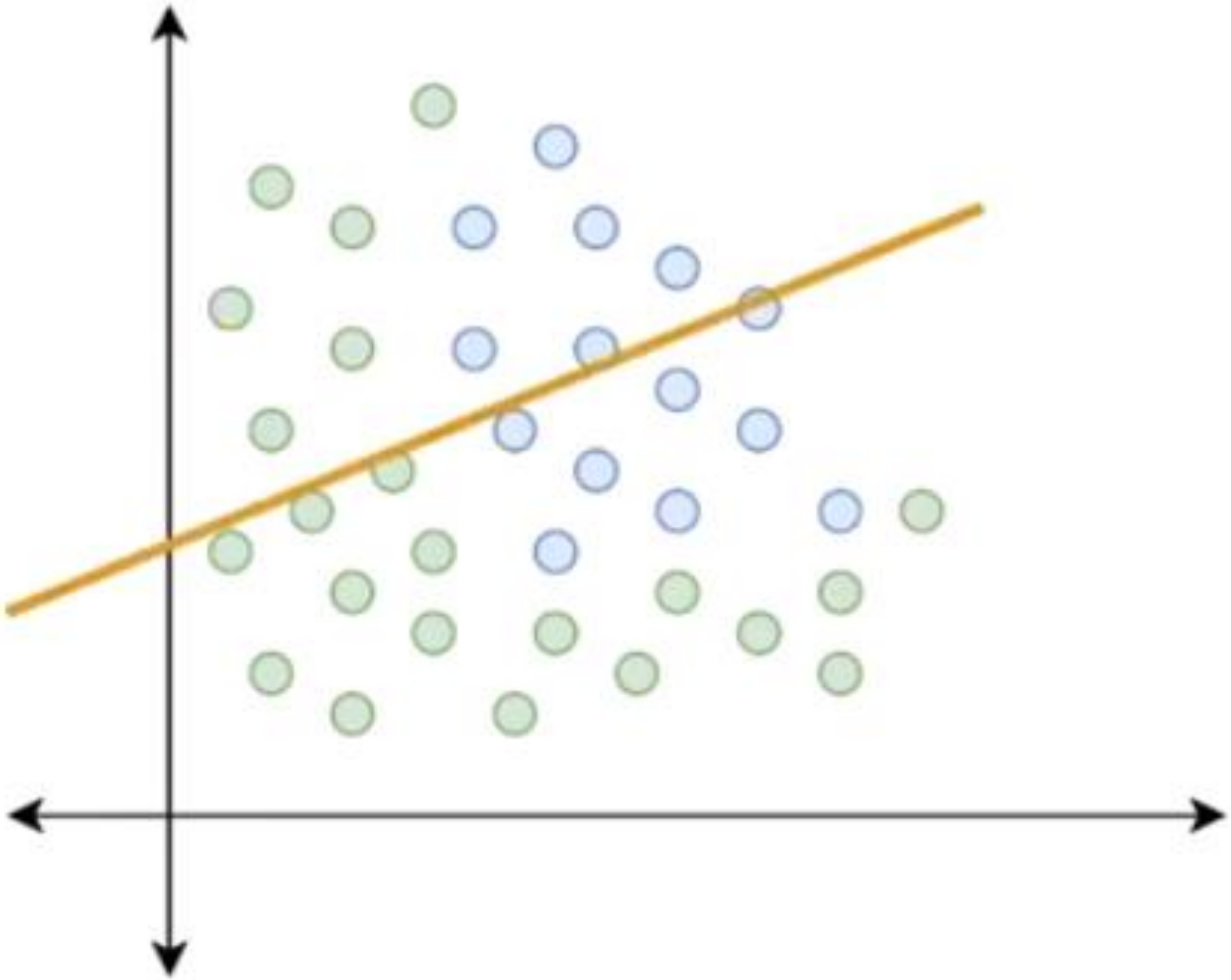
Forward propagation

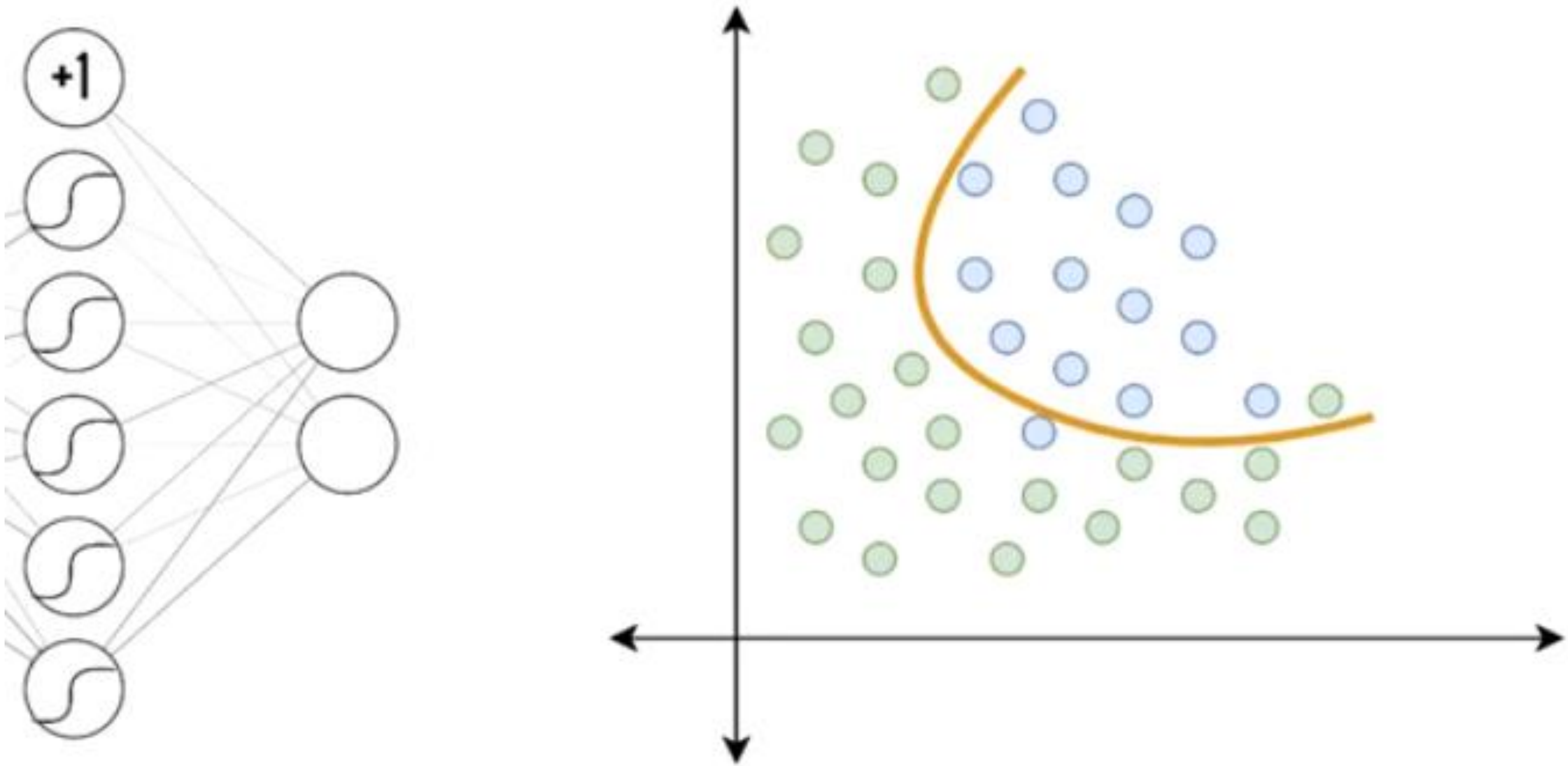


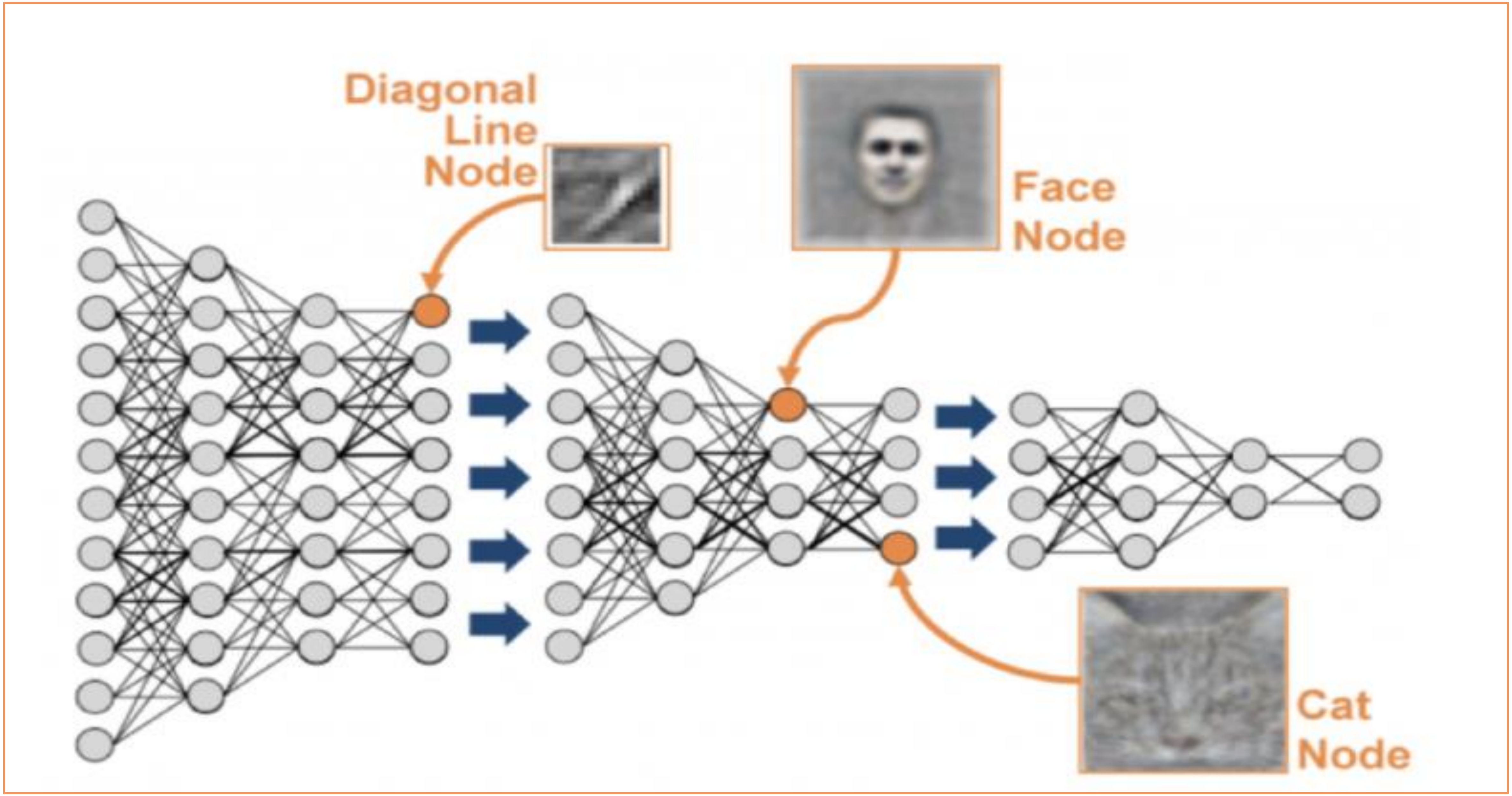

+1


+1

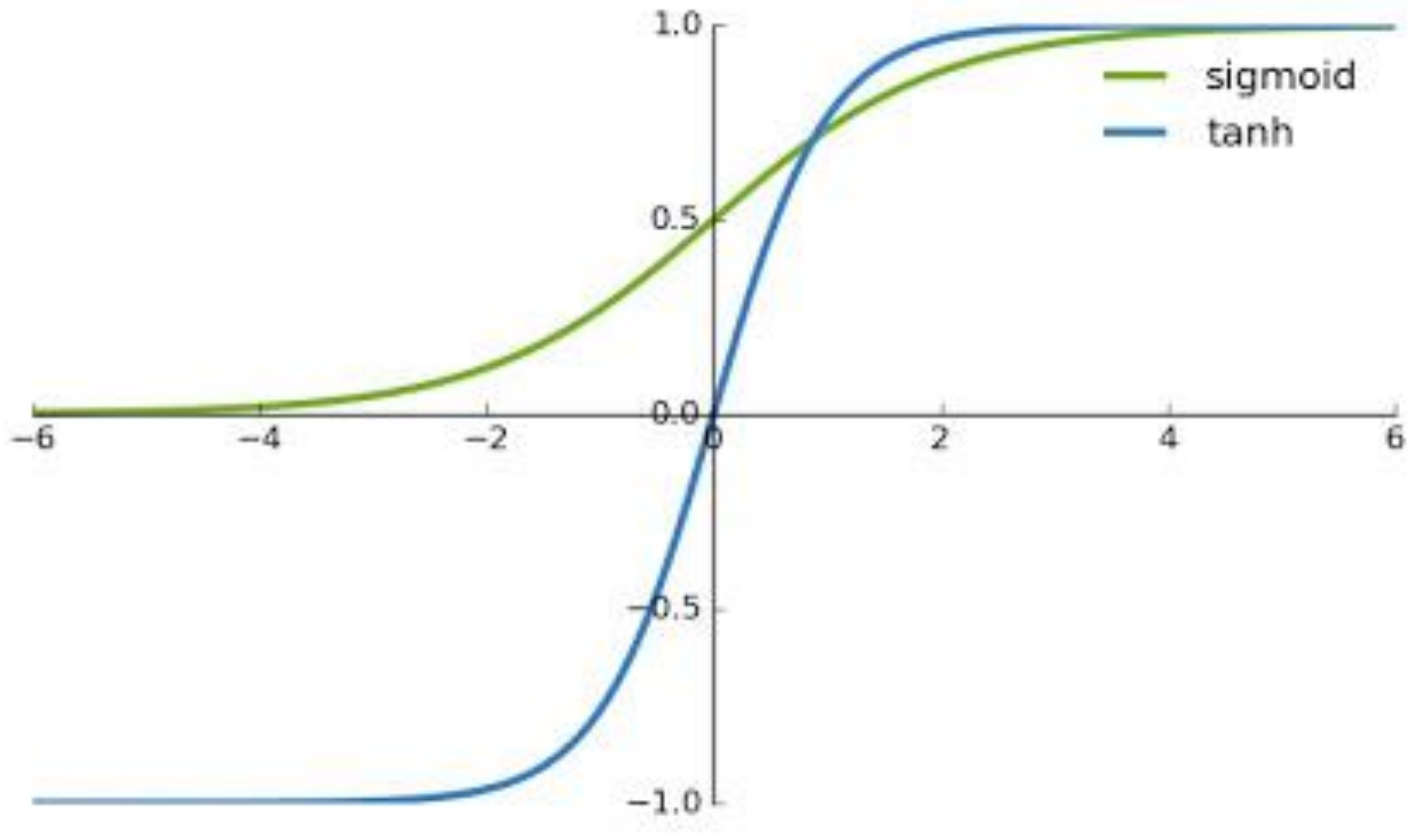


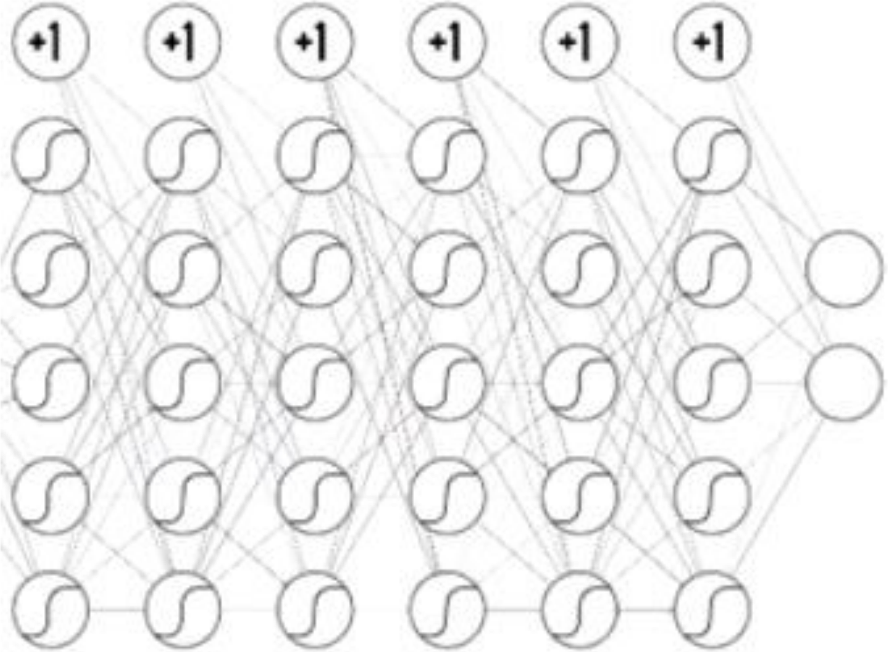

+1


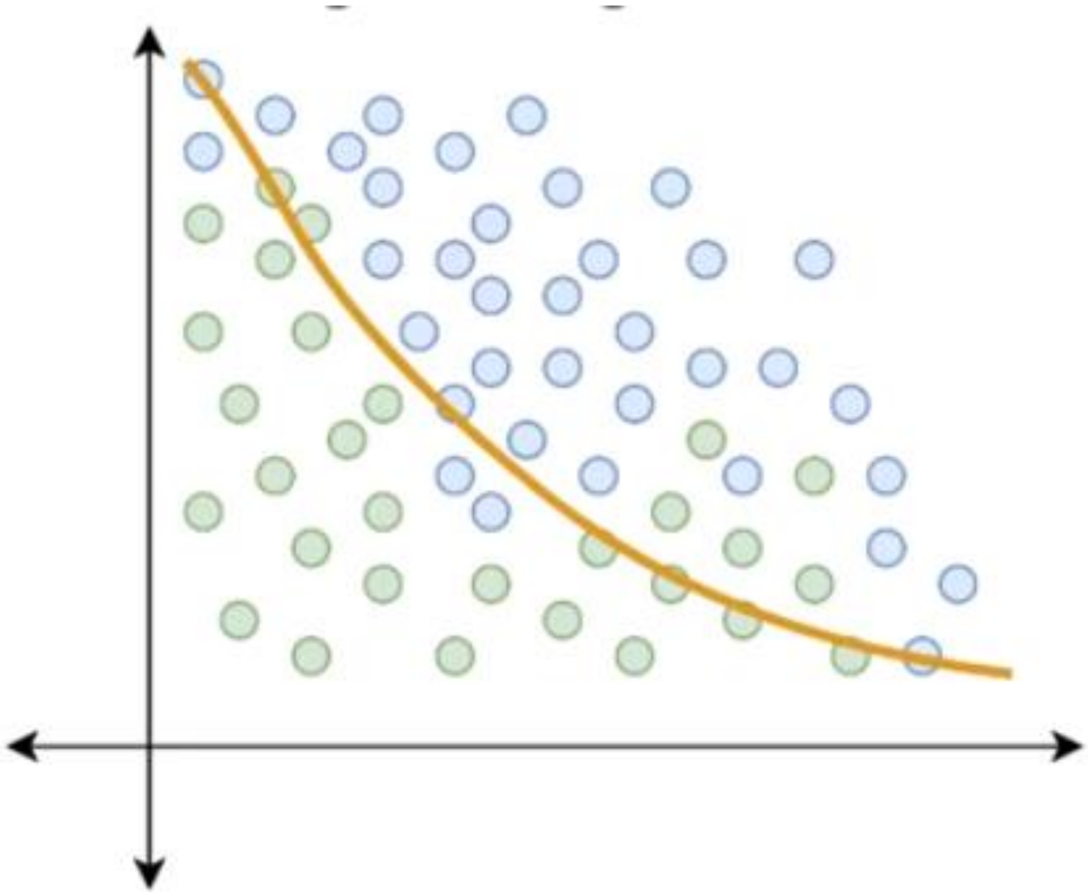

+1

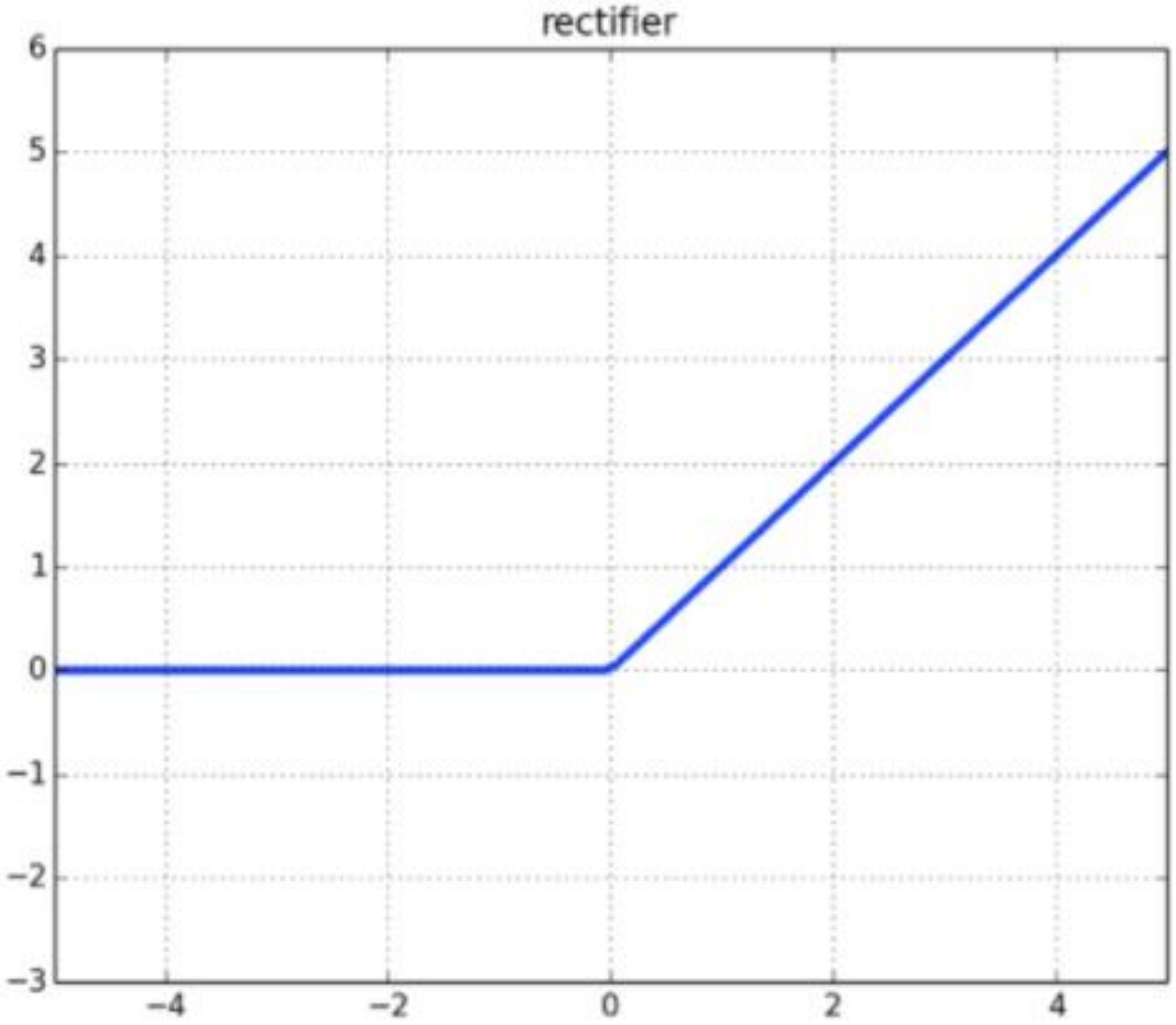


Diagonal
Line
Node
Face
Node
Cat
Node

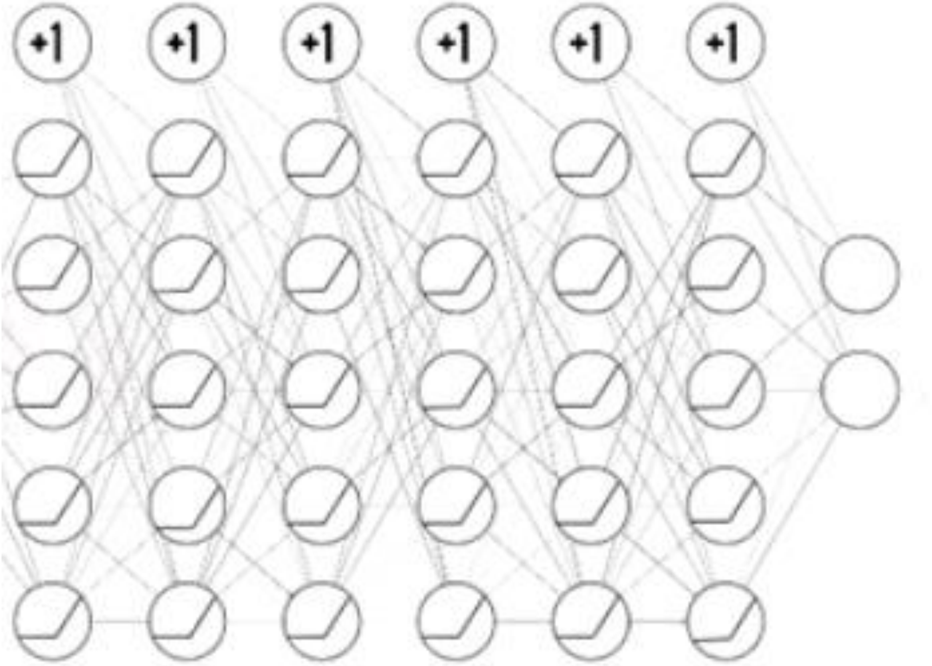

1.0
0.5
0.0
−0.5
−1.0
−6
−4
−2
0
2
4
6
sigmoid
tanh

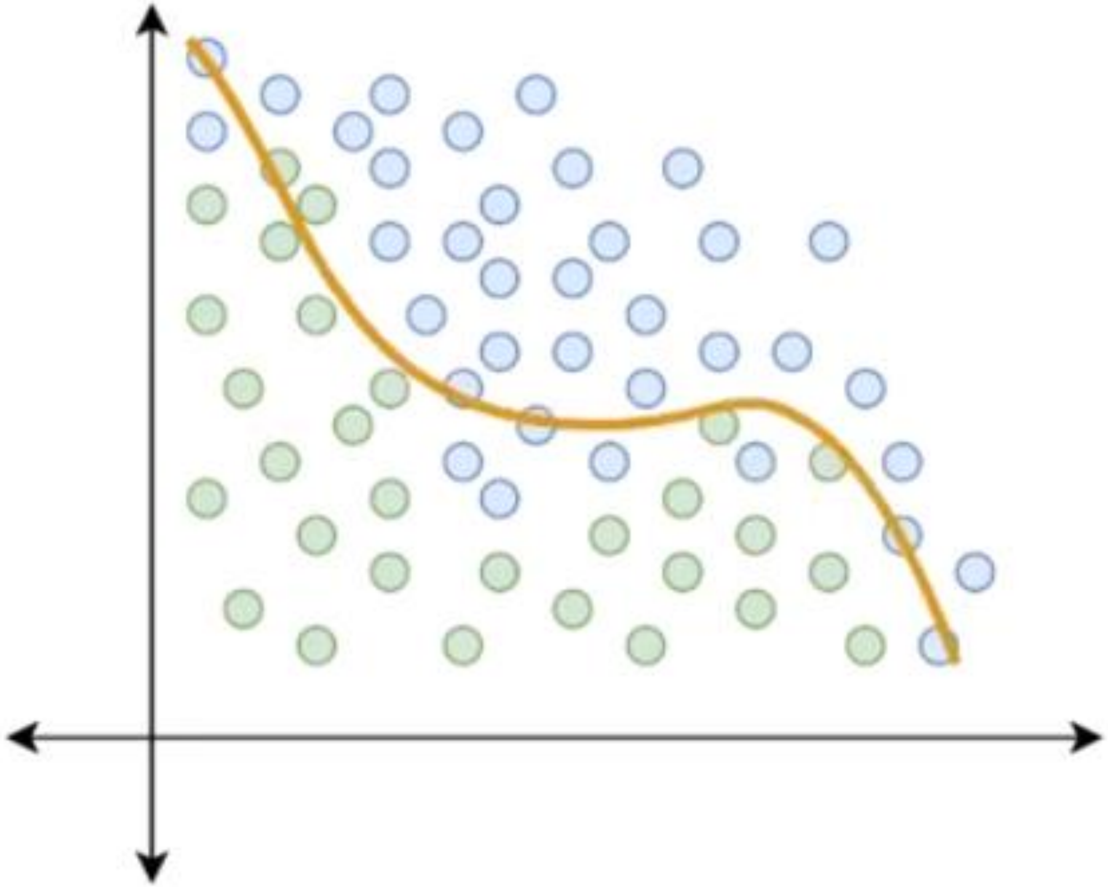

+1
+1
+1
+1
+1
+1

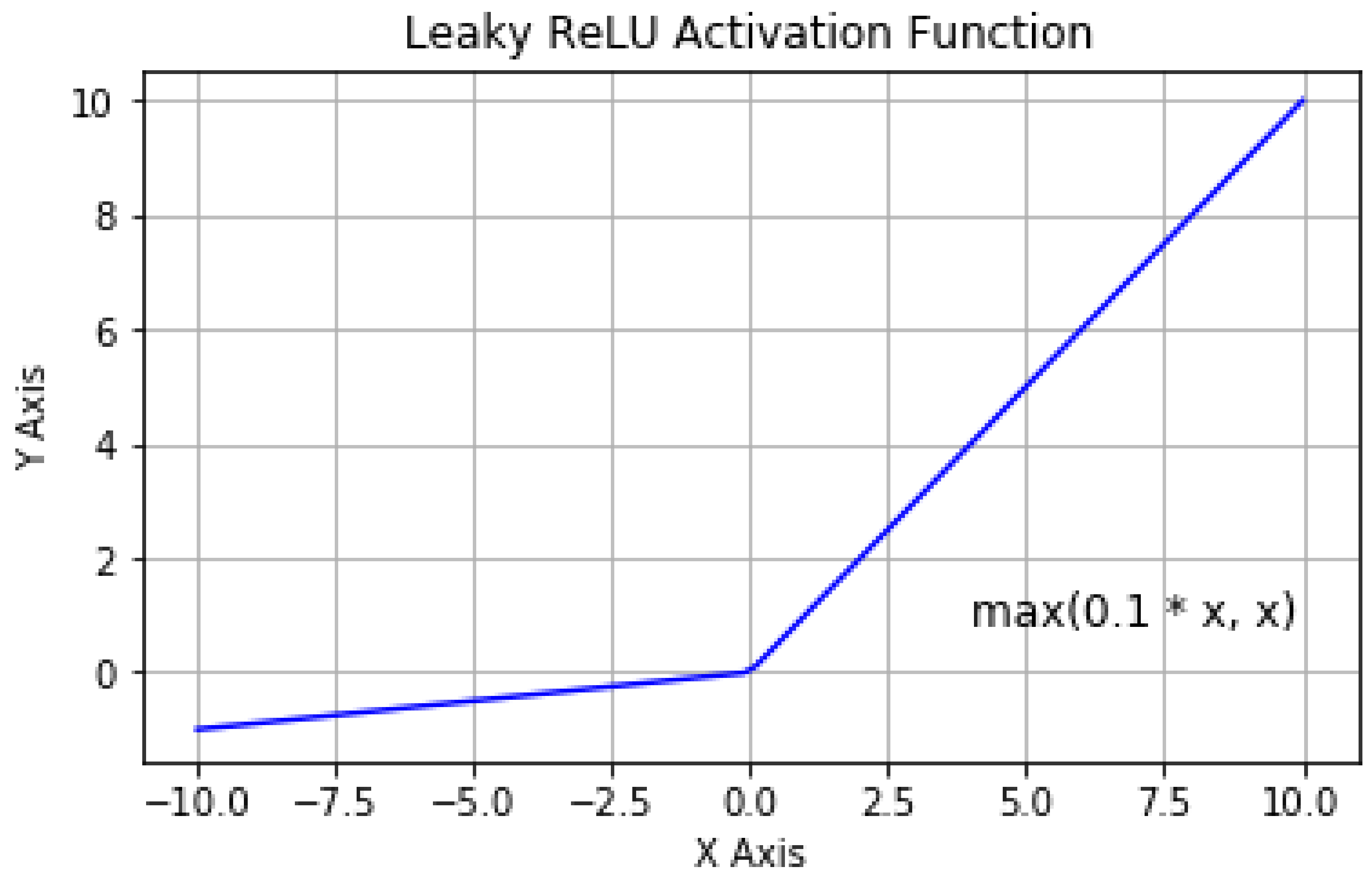


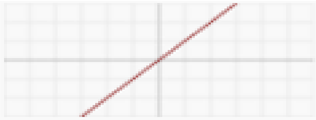
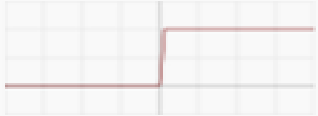
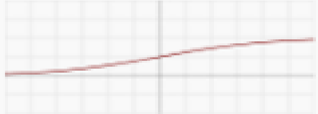
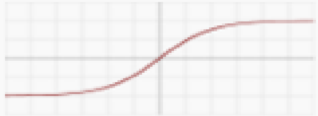
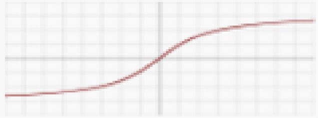
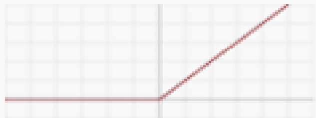
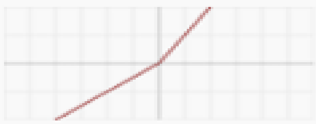
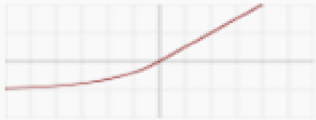
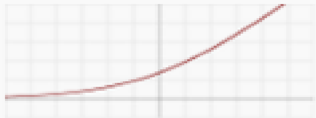


$$RELU(x) = \begin{cases} 0 \; if \;\; x < 0 \\ x \; if \;\; x >= 0 \end{cases}$$

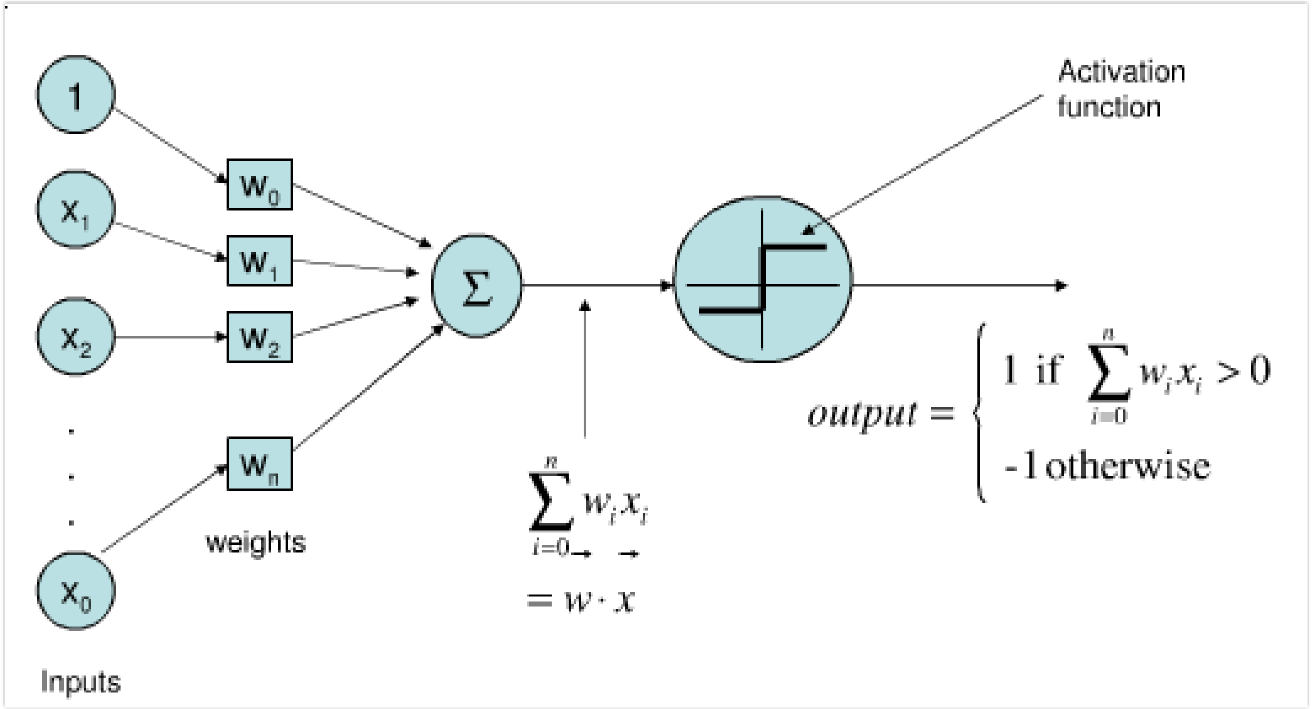

+1
+1
+1
+1
+1
+1




Leaky ReLU Activation Function
10
8
6
4
2
0
Y Axis
max(0.1 * x, x)
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
X Axis

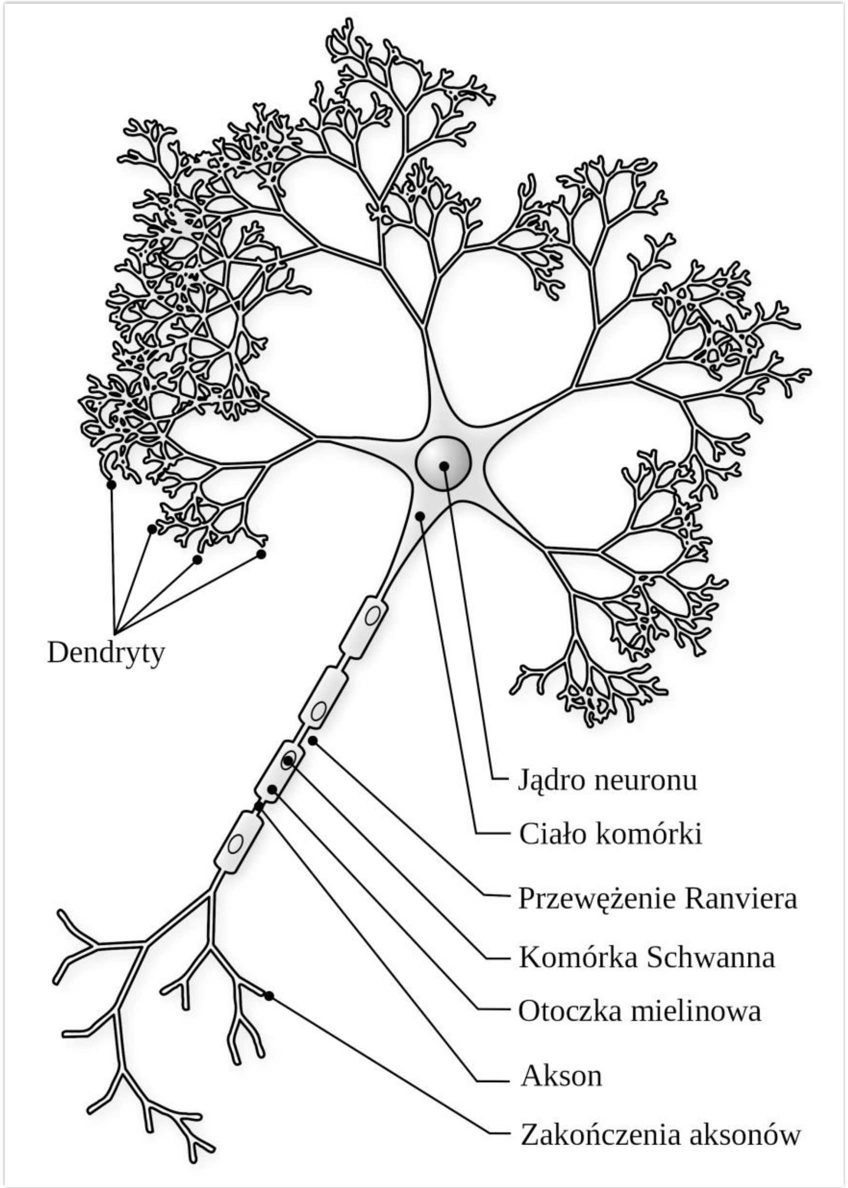
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Perceptron



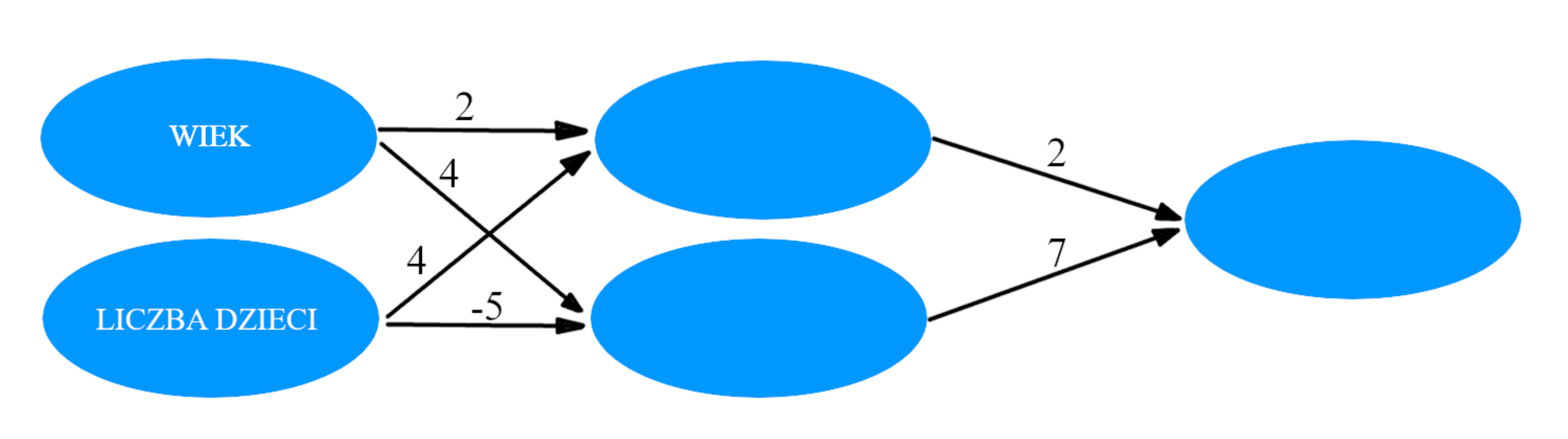
1
x_1
x_2
x_0
w_0
w_1
w_2
w_n
weights
Inputs
Σ
Activation function
$\sum_{i=0}^{n} w_i x_i = \vec{w} \cdot \vec{x}$
$output = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$

Neuron



Dendryty
Jądro neuronu
Ciało komórki
Przewężenie Ranviera
Komórka Schwanna
Otoczka mielinowa
Akson
Zakończenia aksonów

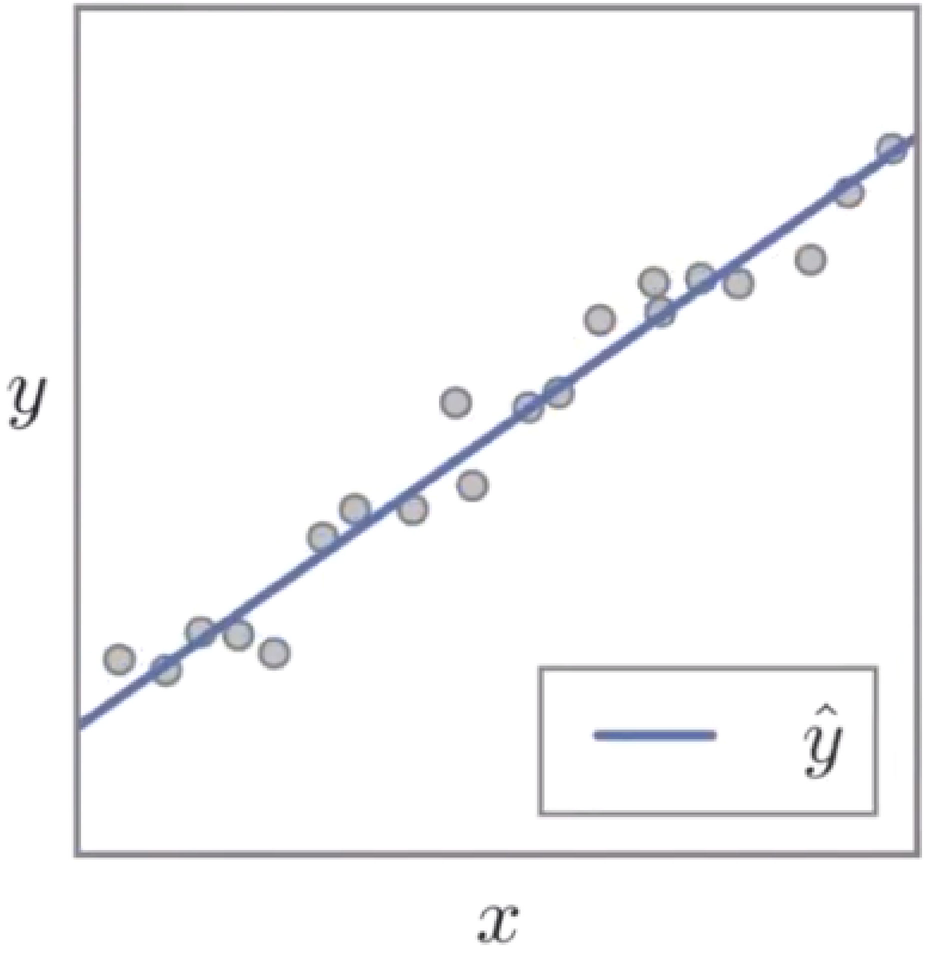
Target = 33

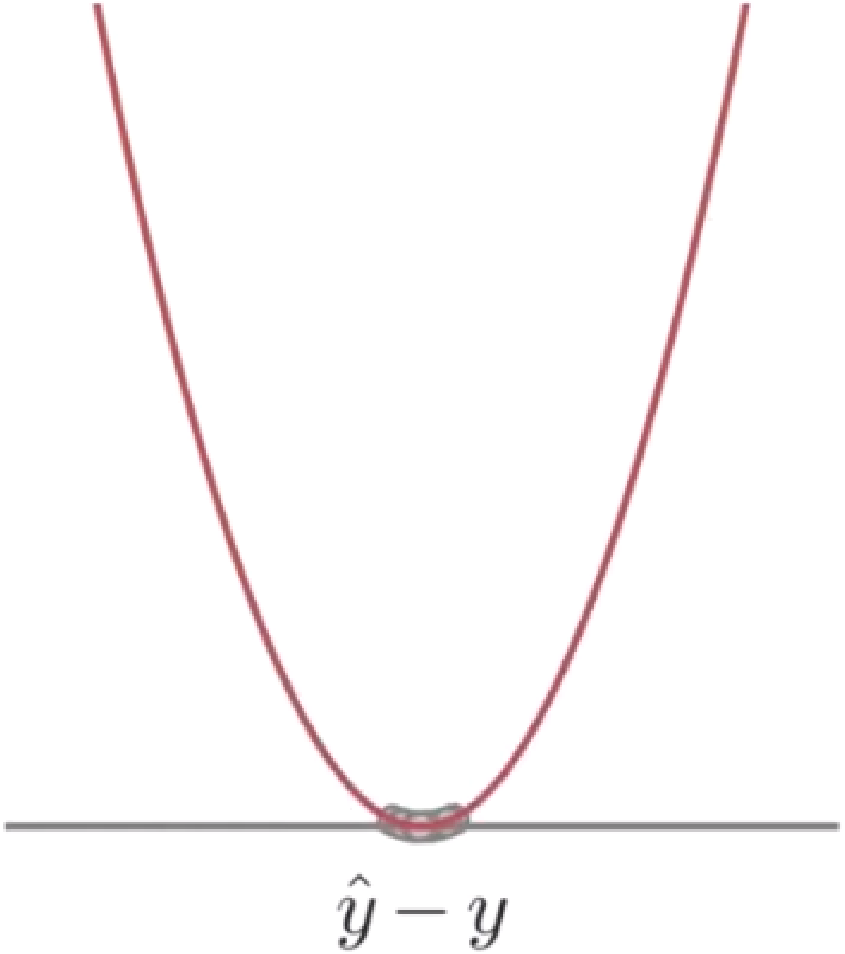


Prediction	Actual	Error
10	20	-10
8	3	5
6	1	5

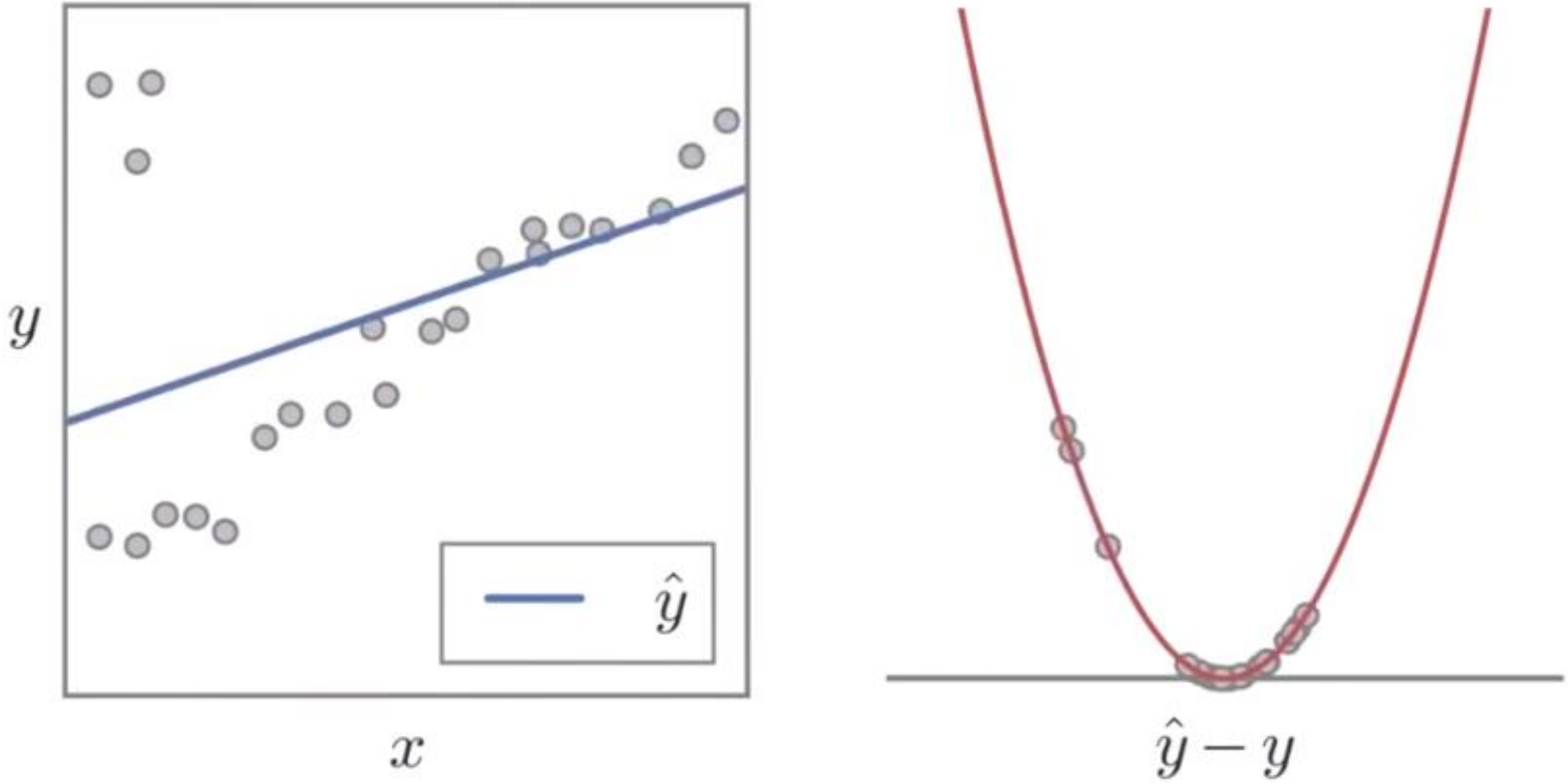
Prediction	Actual	Error	Squared Error
10	20	-10	100
8	3	5	25
6	1	5	25

1. Squared Loss $= (\hat{y} - y)^2$

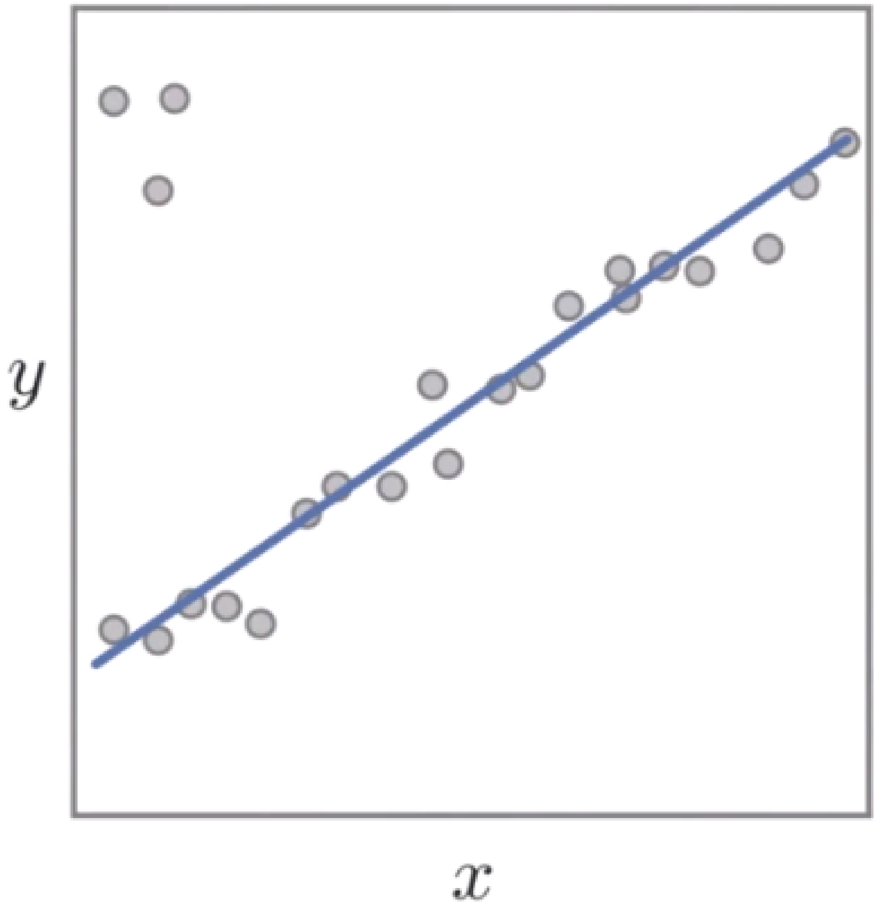


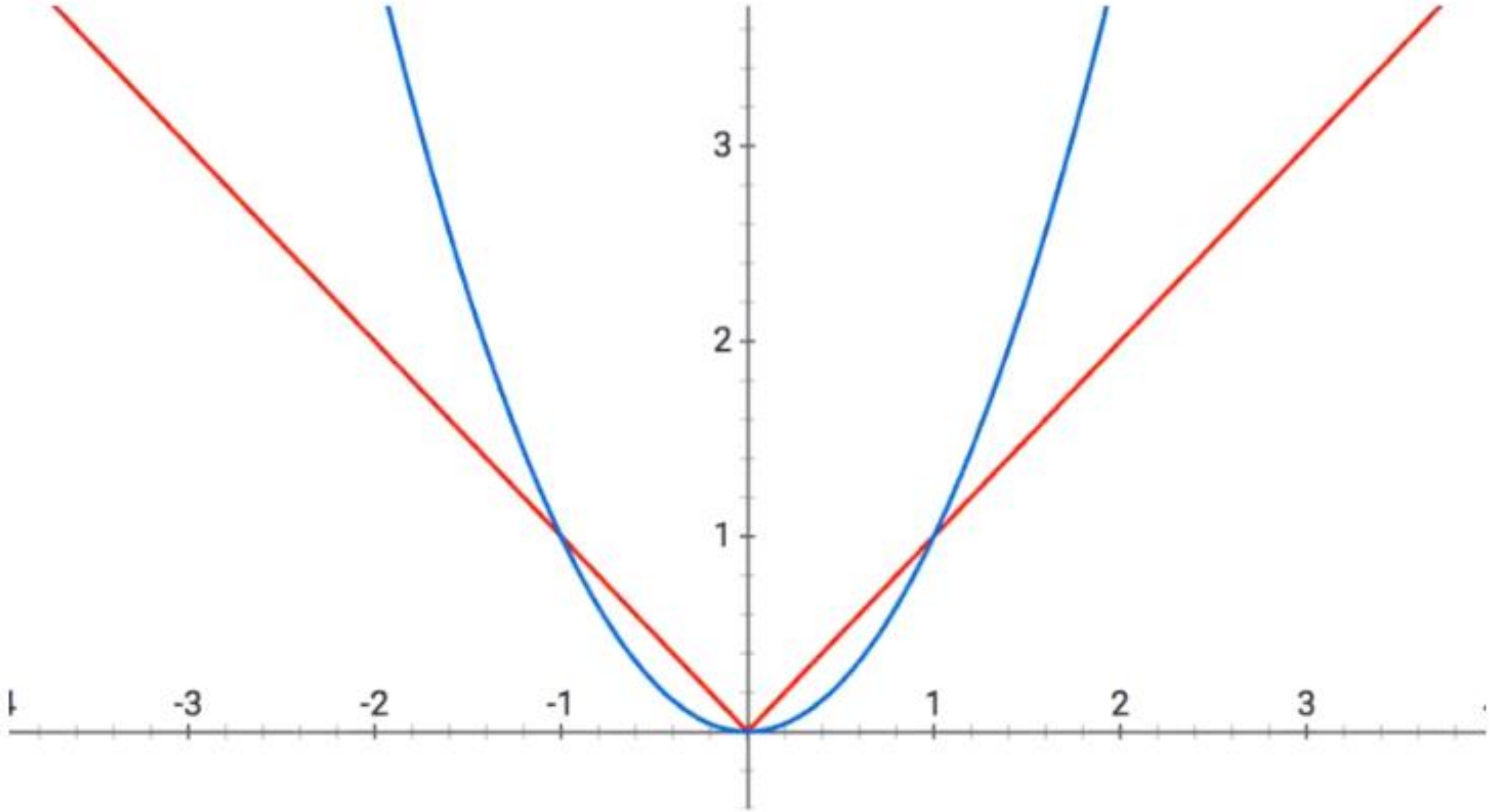


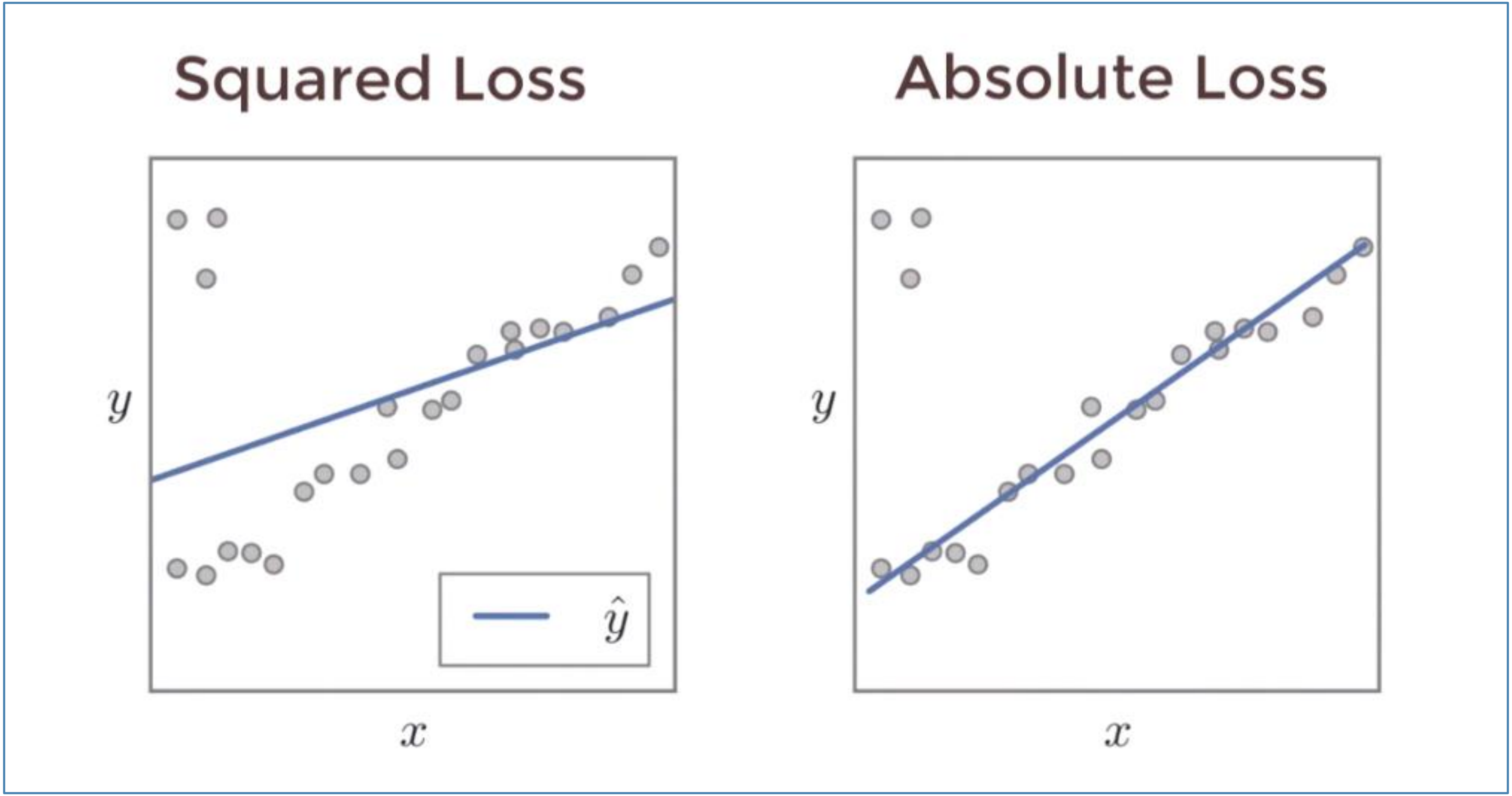
1. Squared Loss $= (\hat{y} - y)^2$



2. Absolute Loss $= |\hat{y} - y|$



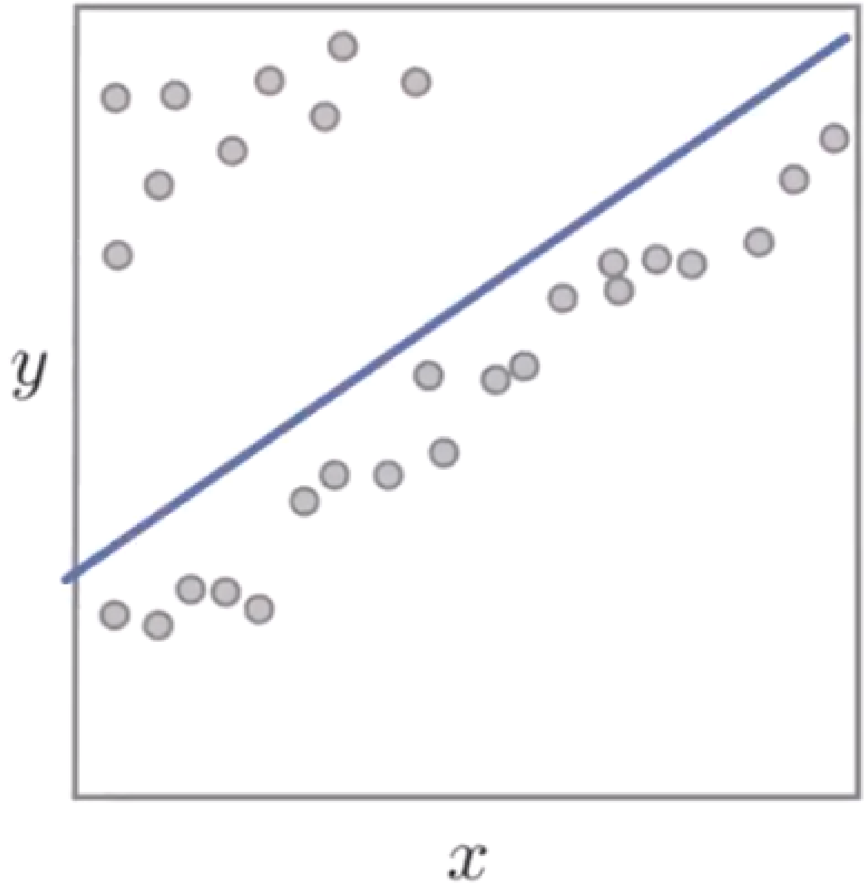

3
2
1
-3
-2
-1
1
2
3

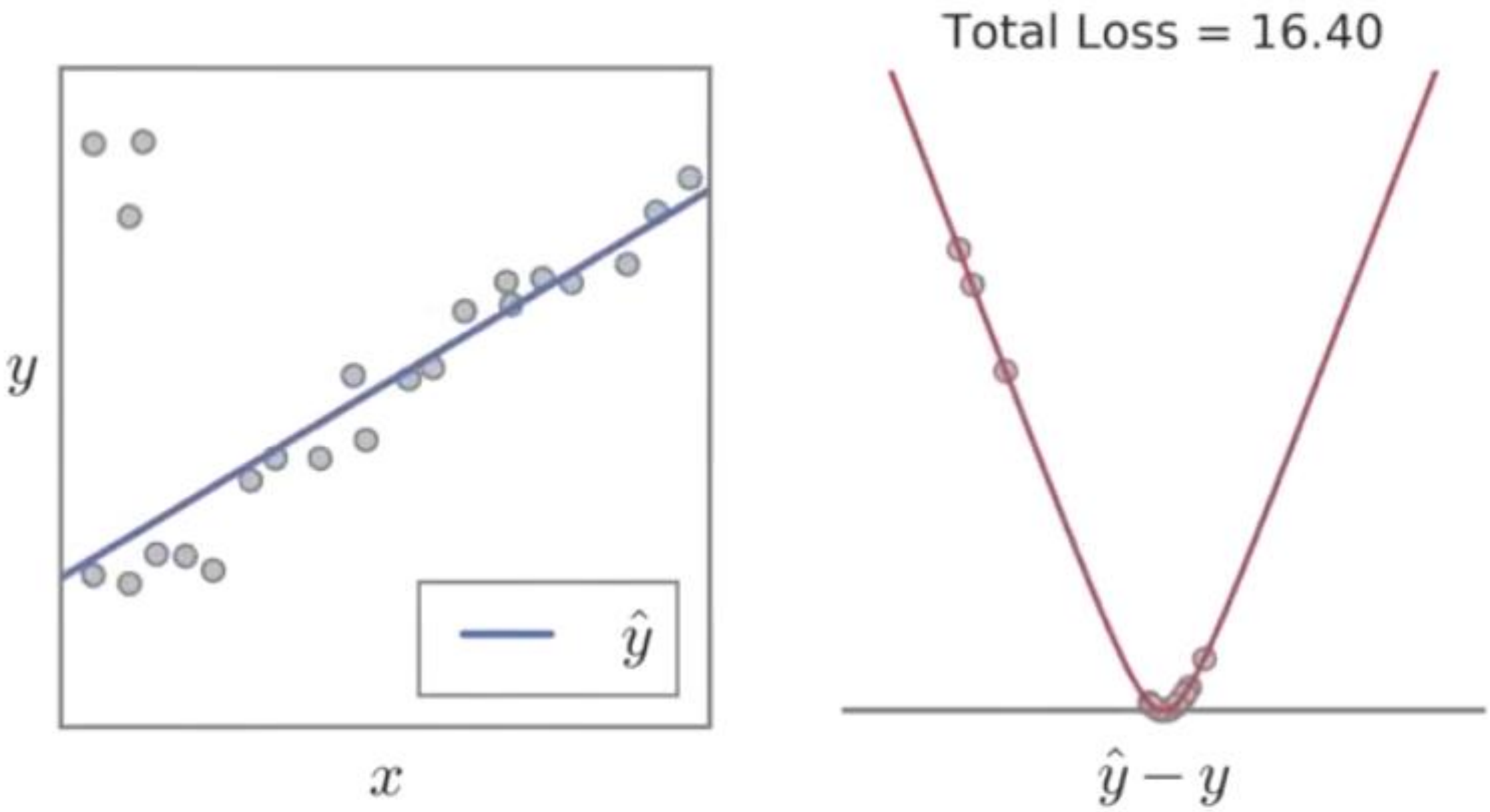


Squared Loss
y
$\hat{y}$
x
Absolute Loss
y
x

3. Pseudo-Huber Loss

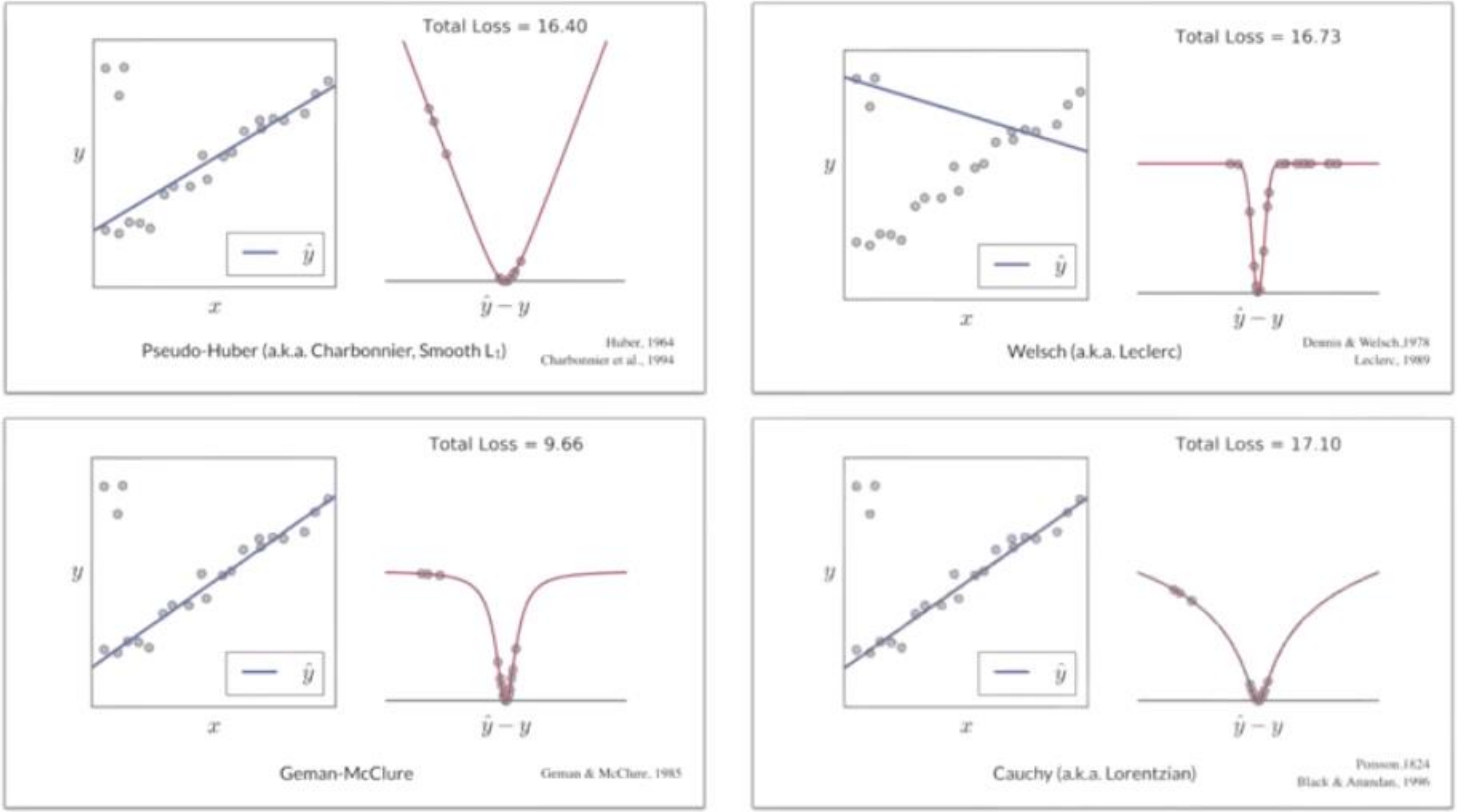
$$= \begin{cases} (y-\hat{y})^2 & ; |y-\hat{y}| \leq \alpha \\ |y-\hat{y}| & ; otherwise \end{cases}$$



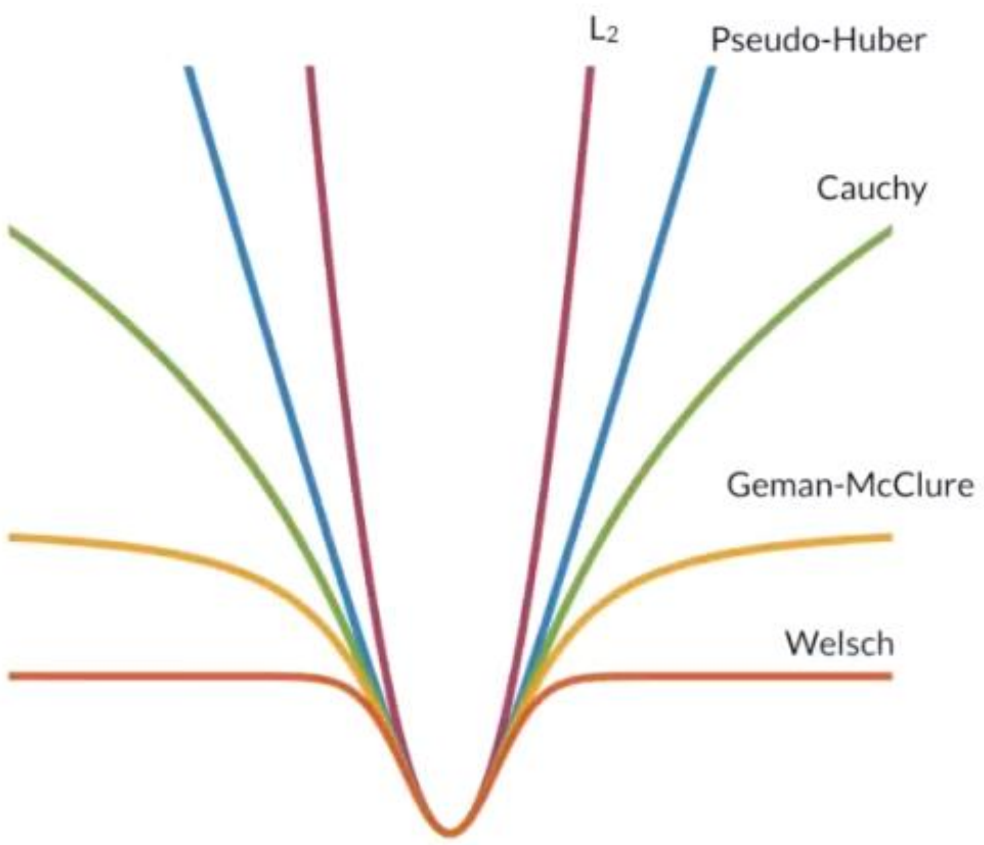


Pseudo-Huber (a.k.a. Charbonnier, Smooth L_1)

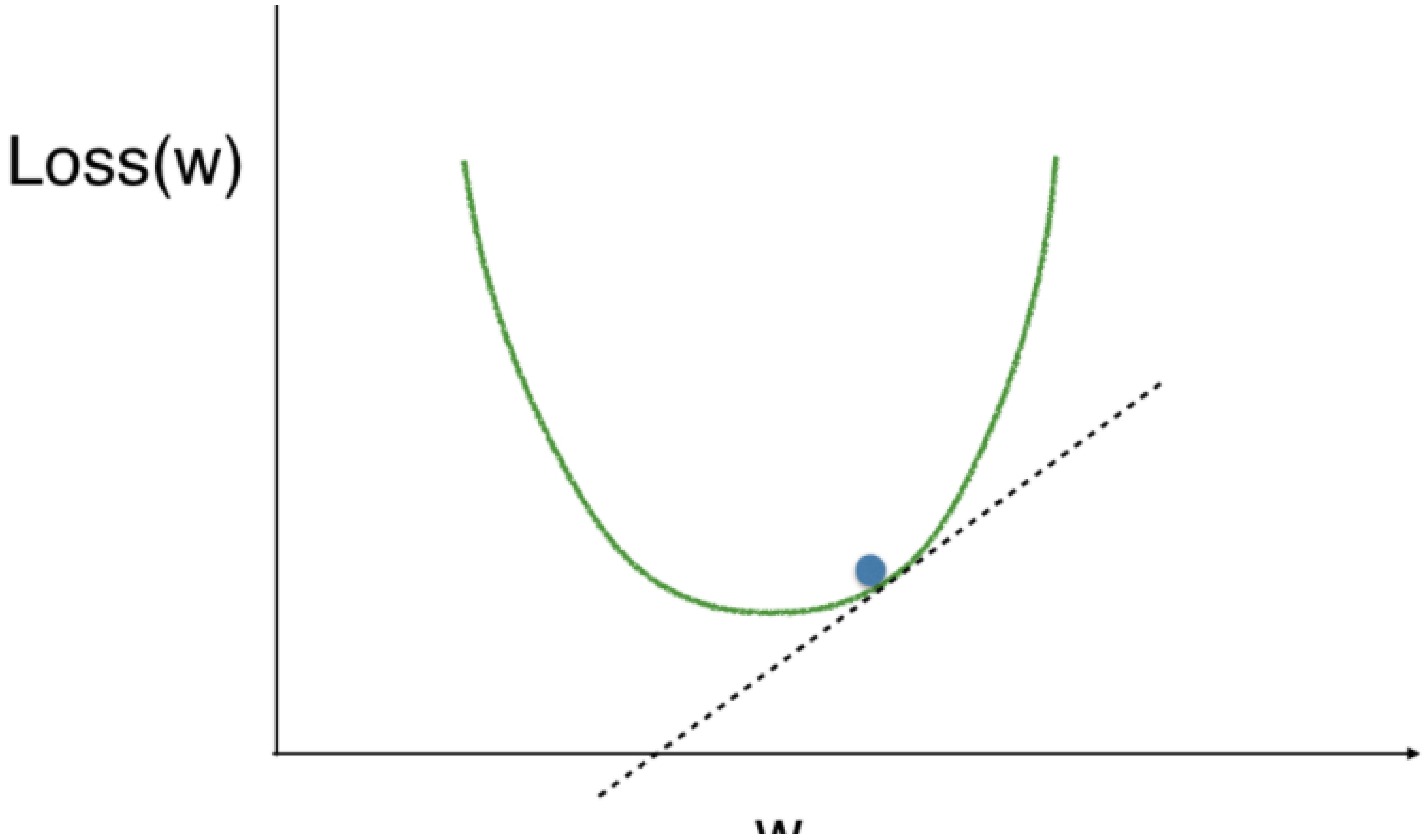
Huber, 1964
Charbonnier et al., 1994

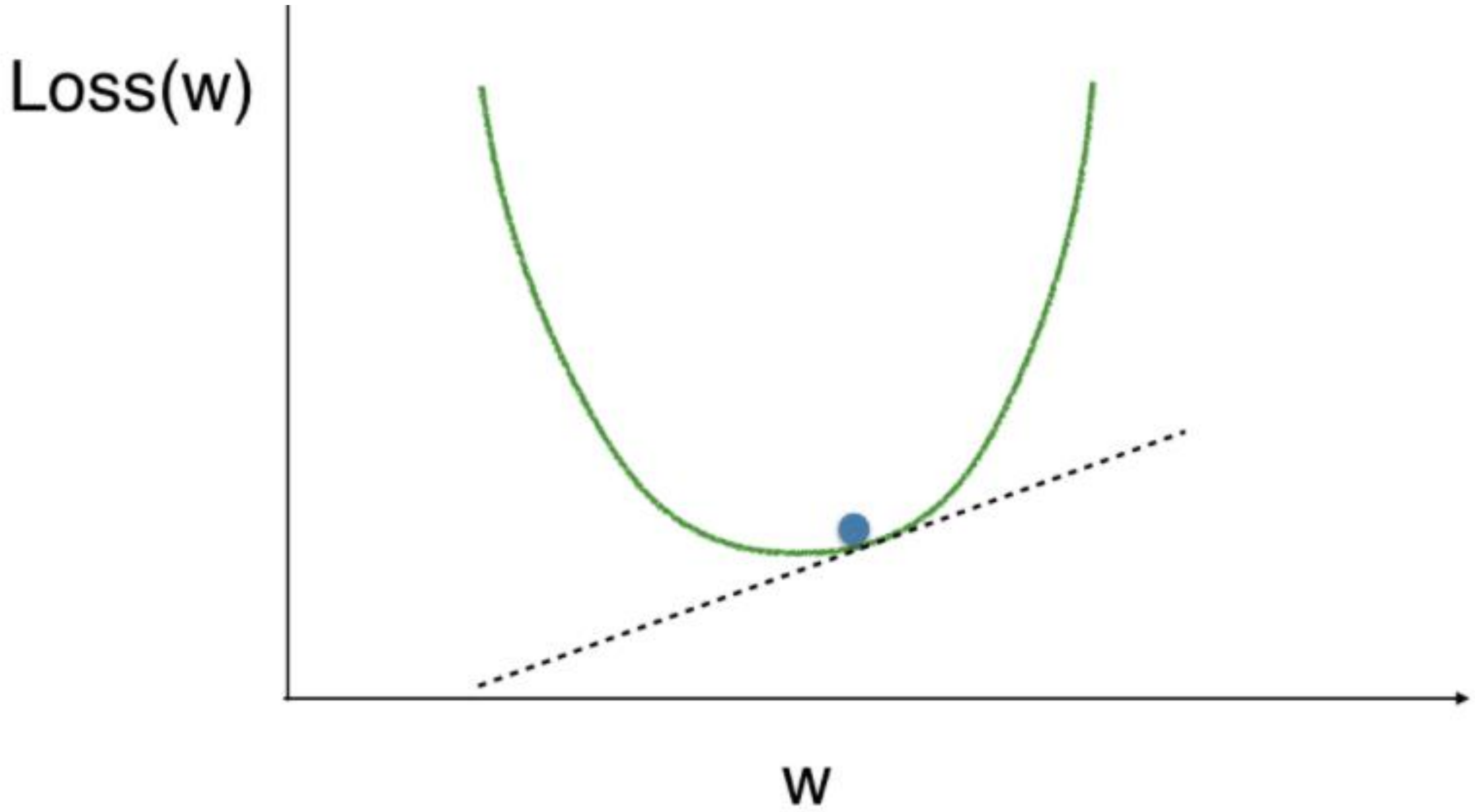


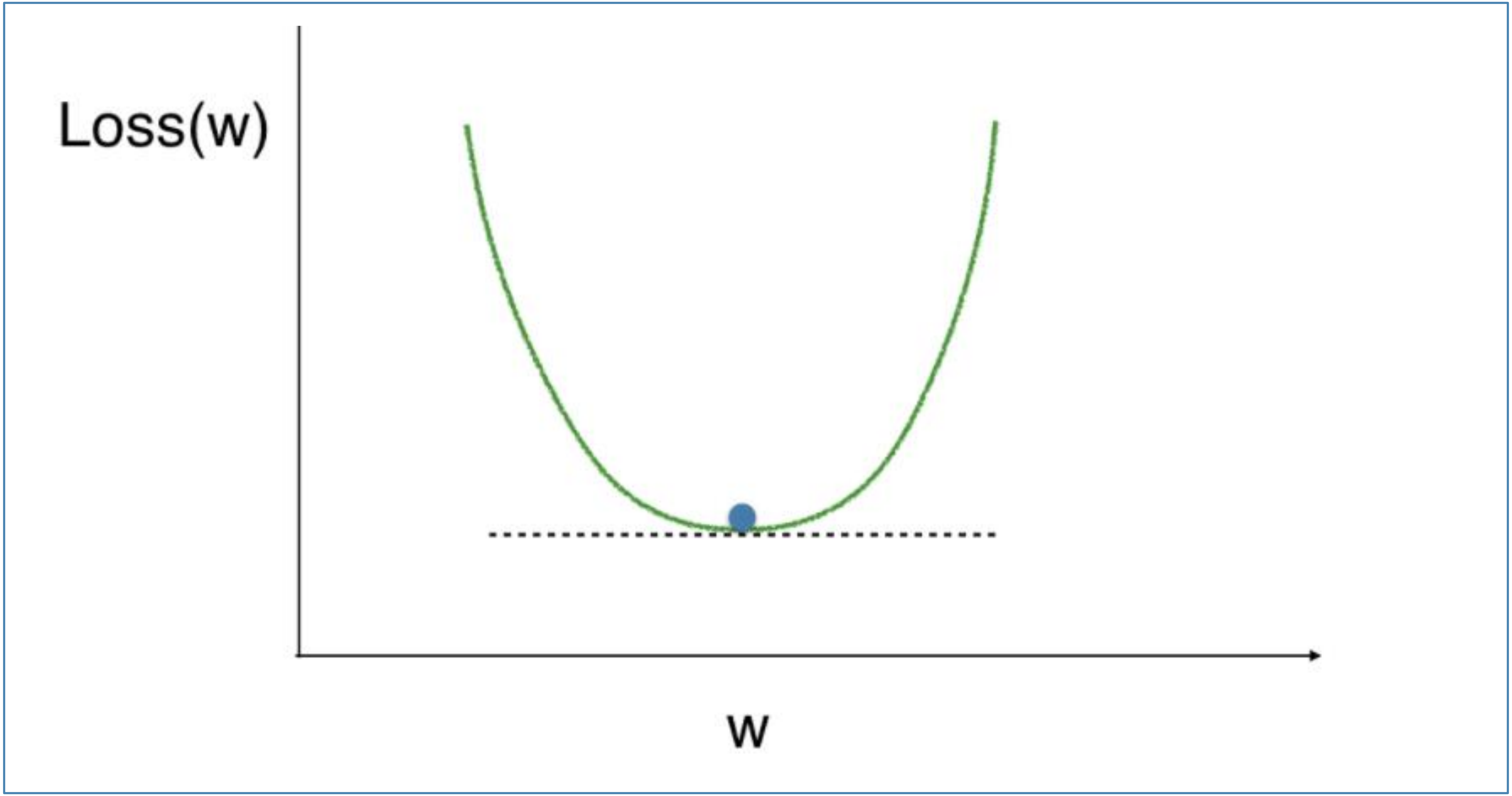
Total Loss = 16.40
Pseudo-Huber (a.k.a. Charbonnier, Smooth L1)
Huber, 1964
Charbonnier et al., 1994
Total Loss = 16.73
Welsch (a.k.a. Leclerc)
Dennis & Welsch, 1978
Leclerc, 1989
Total Loss = 9.66
Geman-McClure
Geman & McClure, 1985
Total Loss = 17.10
Cauchy (a.k.a. Lorentzian)
Poisson, 1824
Black & Anandan, 1996

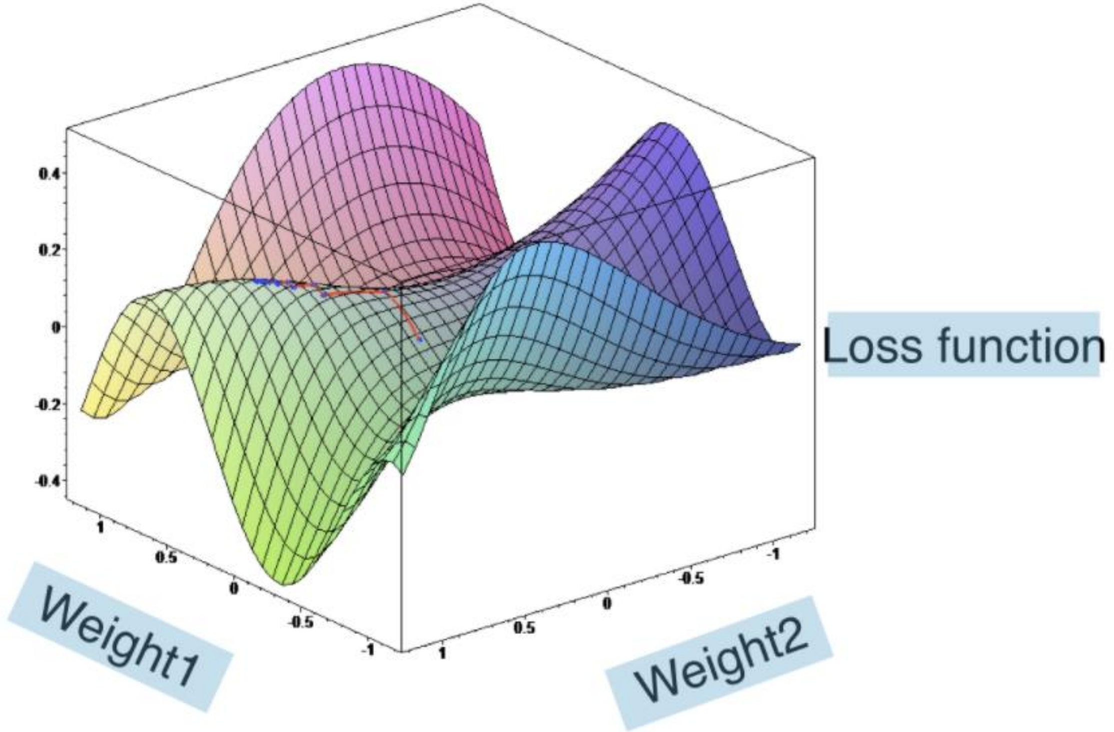


$$\rho(x,\alpha) = \frac{|2-\alpha|}{\alpha}\left(\left(\frac{x^2}{|2-\alpha|}+1\right)^{\frac{\alpha}{2}} - 1\right)$$


Loss(w)
w


Loss(w)
w


Loss(w)
w

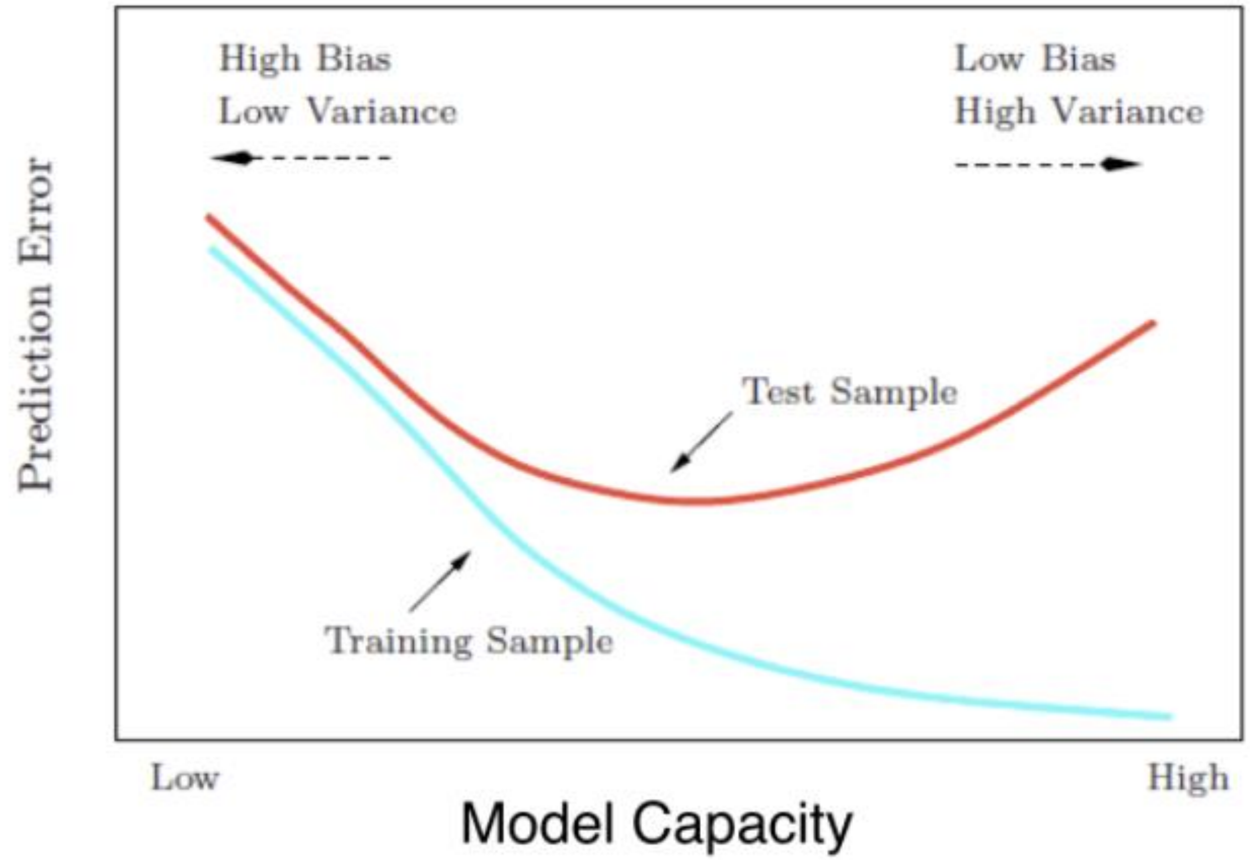


0.4
0.2
0
-0.2
-0.4
1
0.5
0
-0.5
-1
Weight1
1
0.5
0
-0.5
-1
Weight2
Loss function

shot_clock	dribbles	touch_time	shot_dis	close_def_dis	shot_result
10.8	2	1.9	7.7	1.3	1
3.4	0	0.8	28.2	6.1	0
0	3	2.7	10.1	0.9	0
10.3	2	1.9	17.2	3.4	0

shot_result
1
0
0
0

Outcome 0	Outcome 1
0	1
1	0
1	0
1	0


High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Training Sample
Low
High
Model Capacity

Hidden Layers	Nodes Per Layer	Mean Squared Error	Next Step
1	100	5.4	Increase Capacity
1	250	4.8	Increase Capacity
2	250	4.4	Increase Capacity
3	250	4.5	Decrease Capacity
3	200	4.3	Done