Introduction to Machine Learning and Neural Networks

by Nicolas Symeou

Thanks to

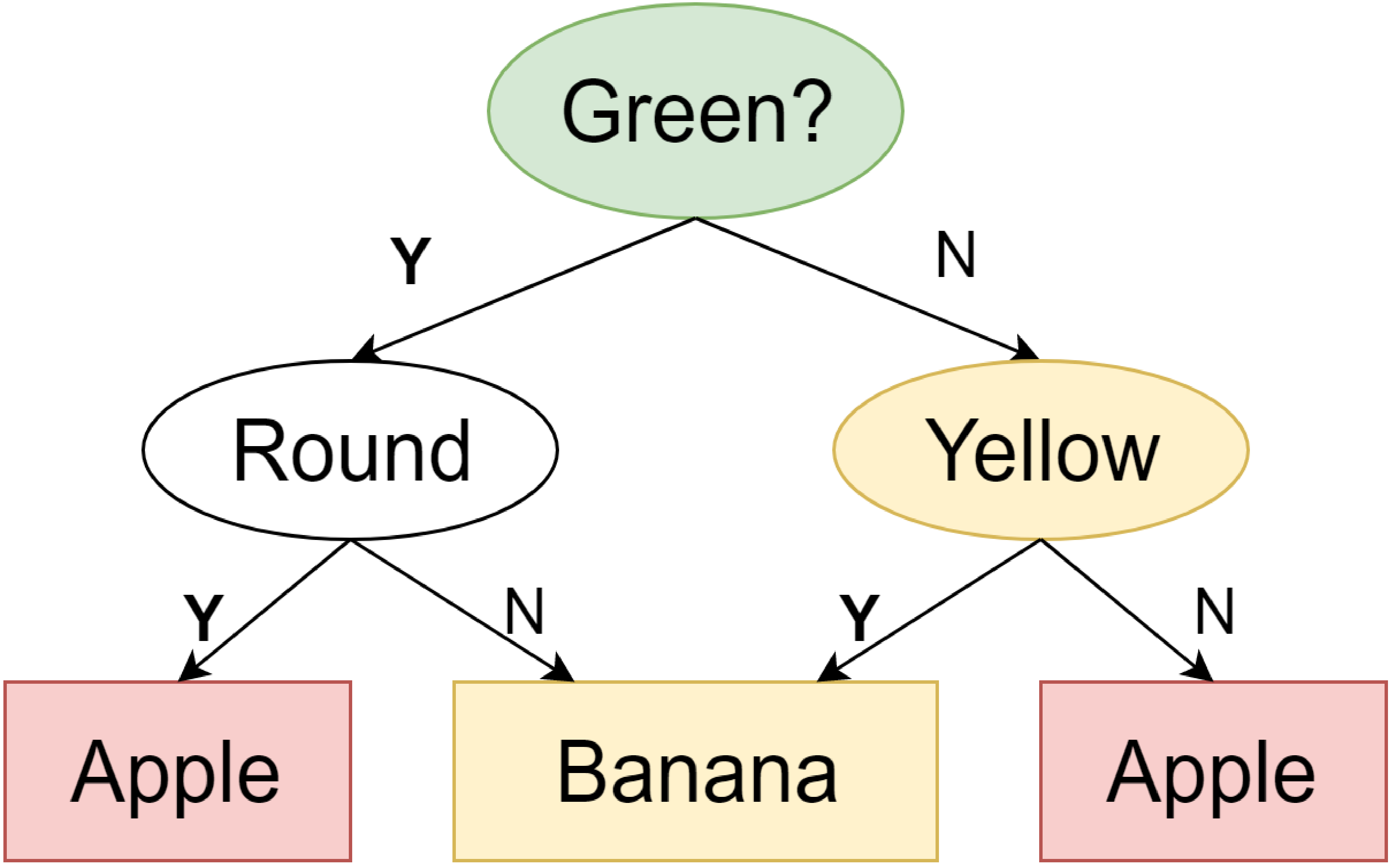
Florin Schwappach

for correction and feedback

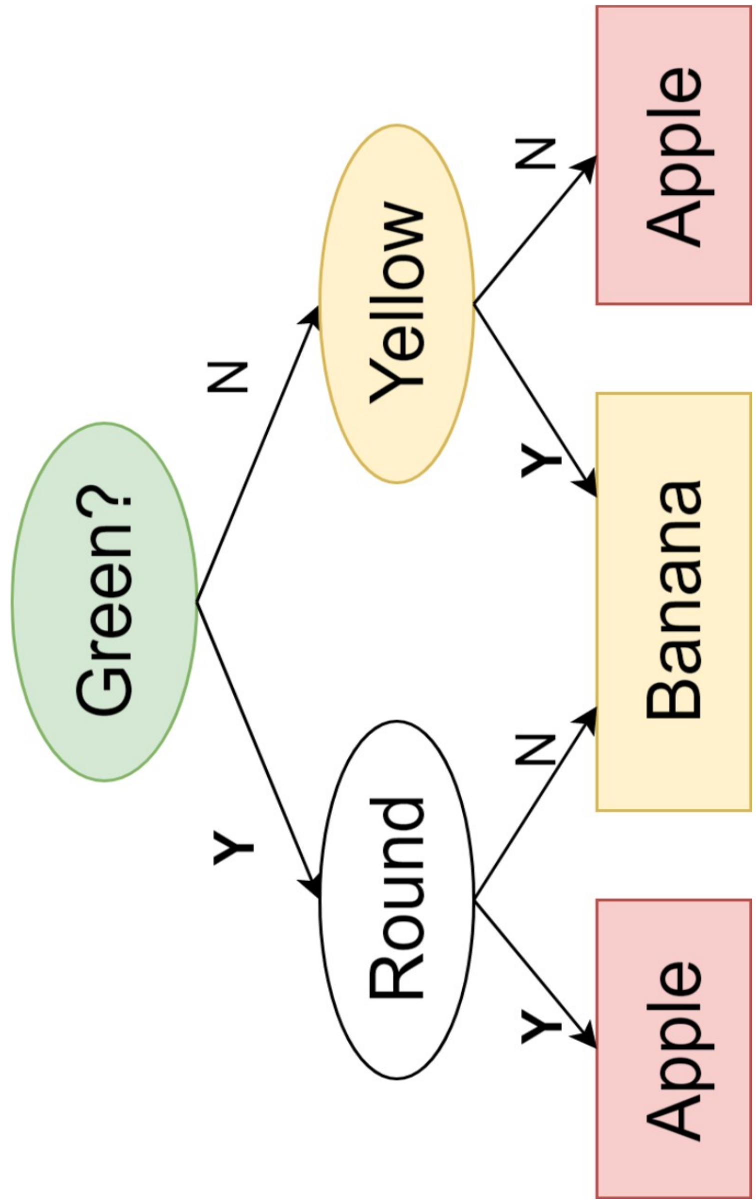
Machine Learning,
An introduction 1:

The BIG question

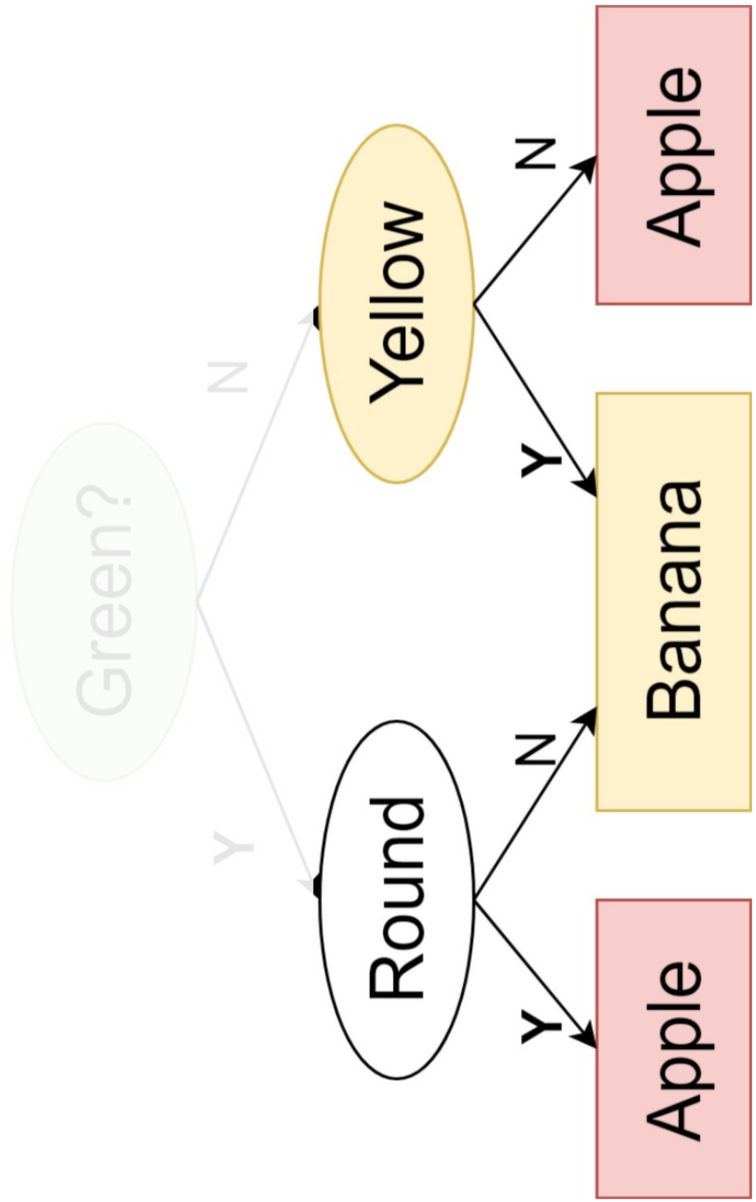
Banana or Apple?



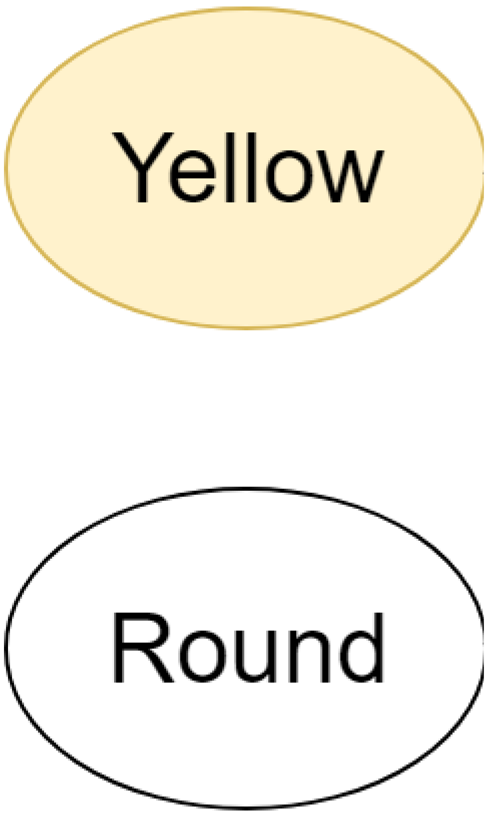
Banana or Apple?



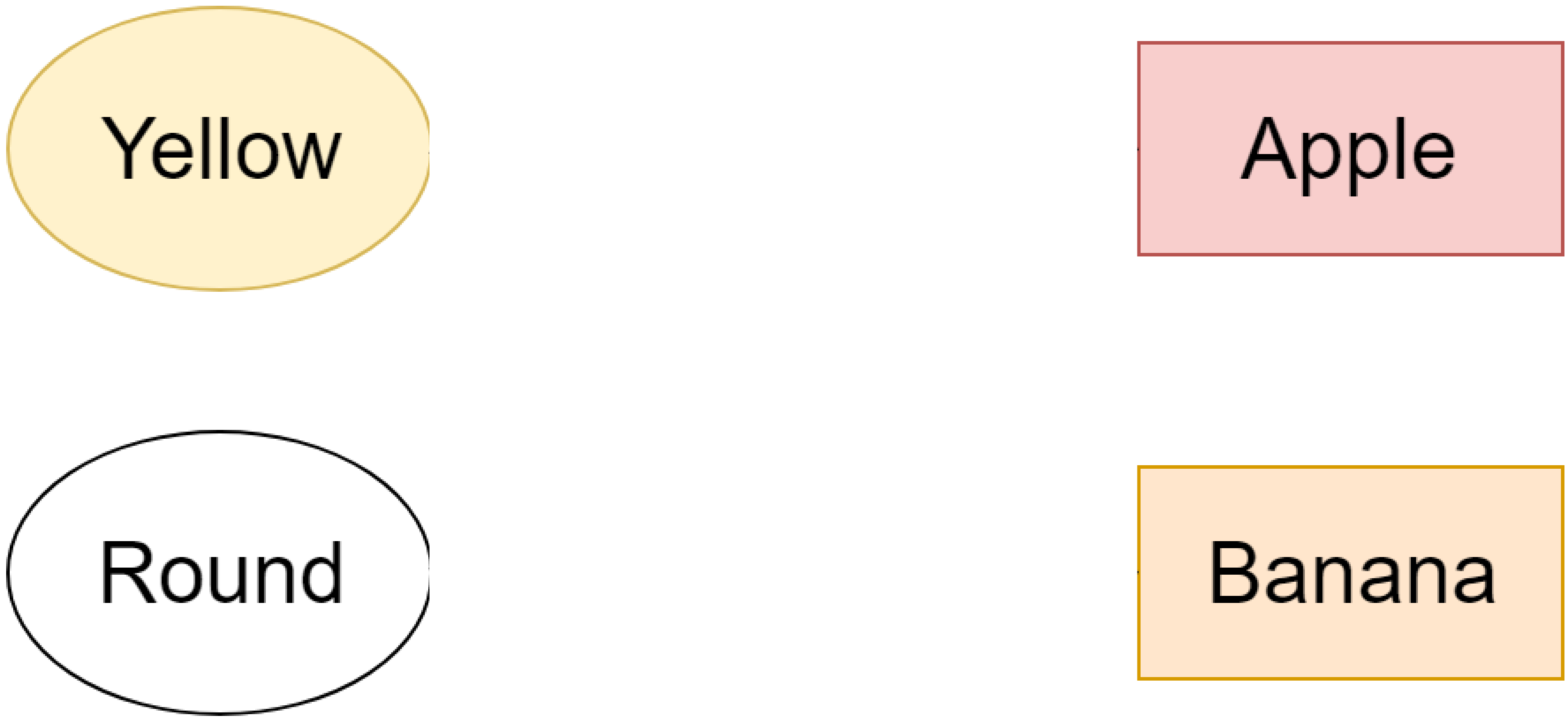
Banana or Apple?



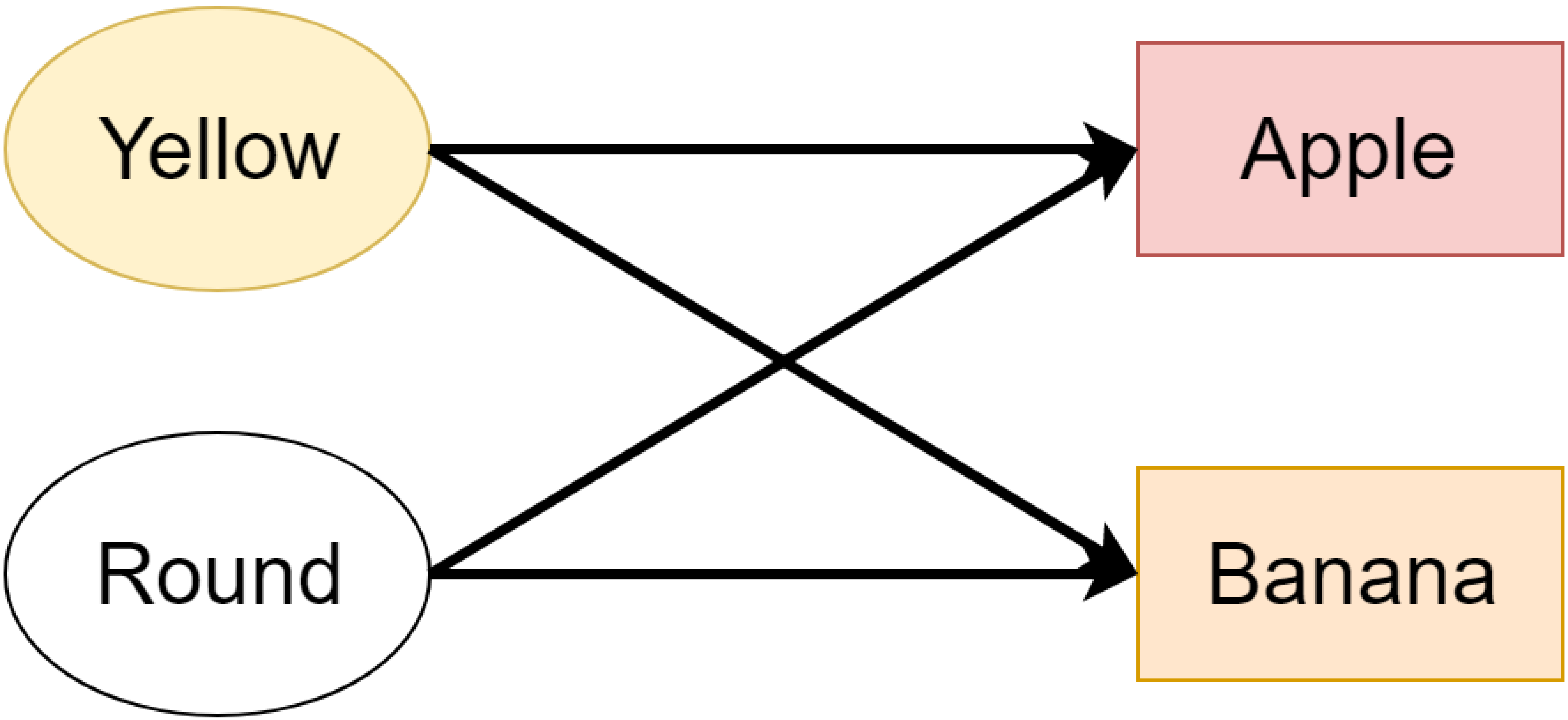
Banana or Apple?



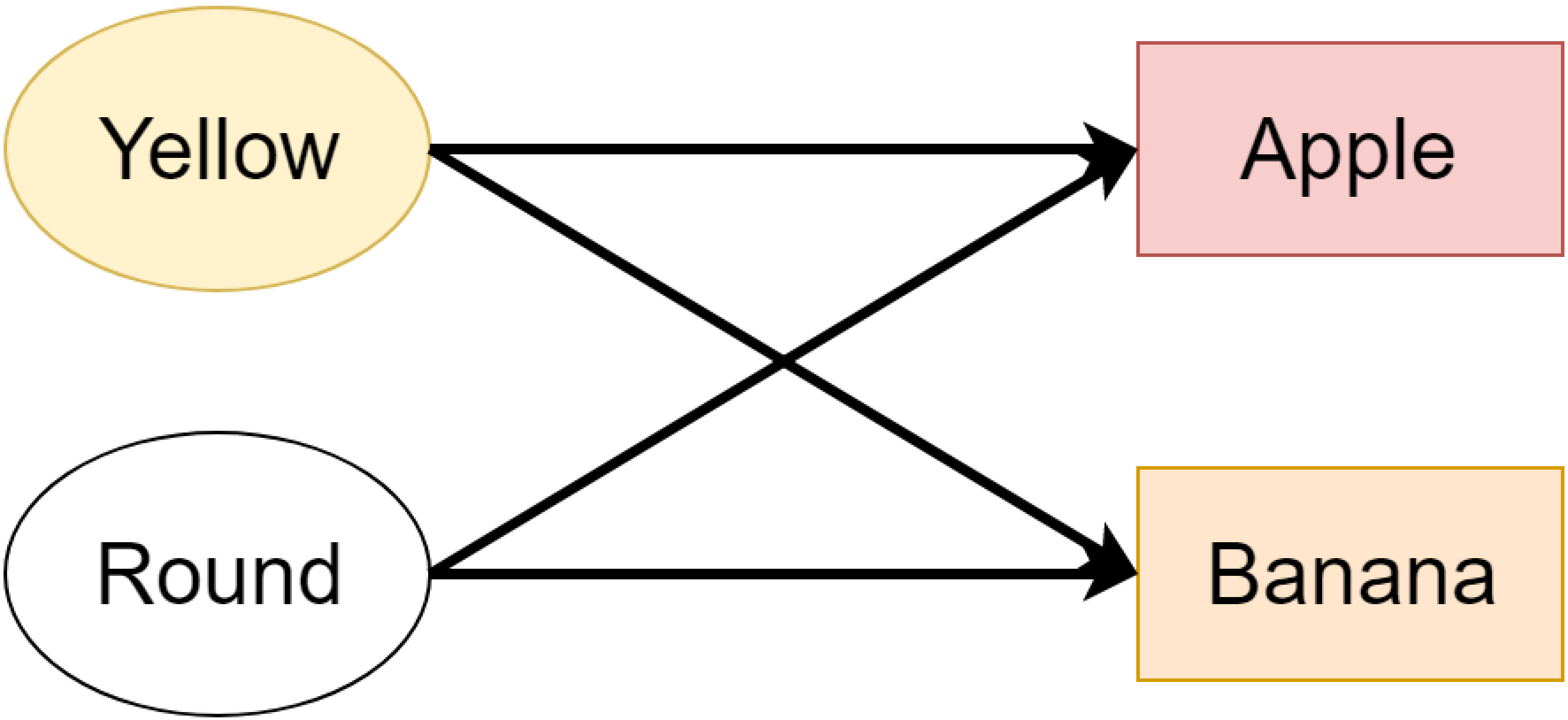
Banana or Apple?



Banana or Apple?



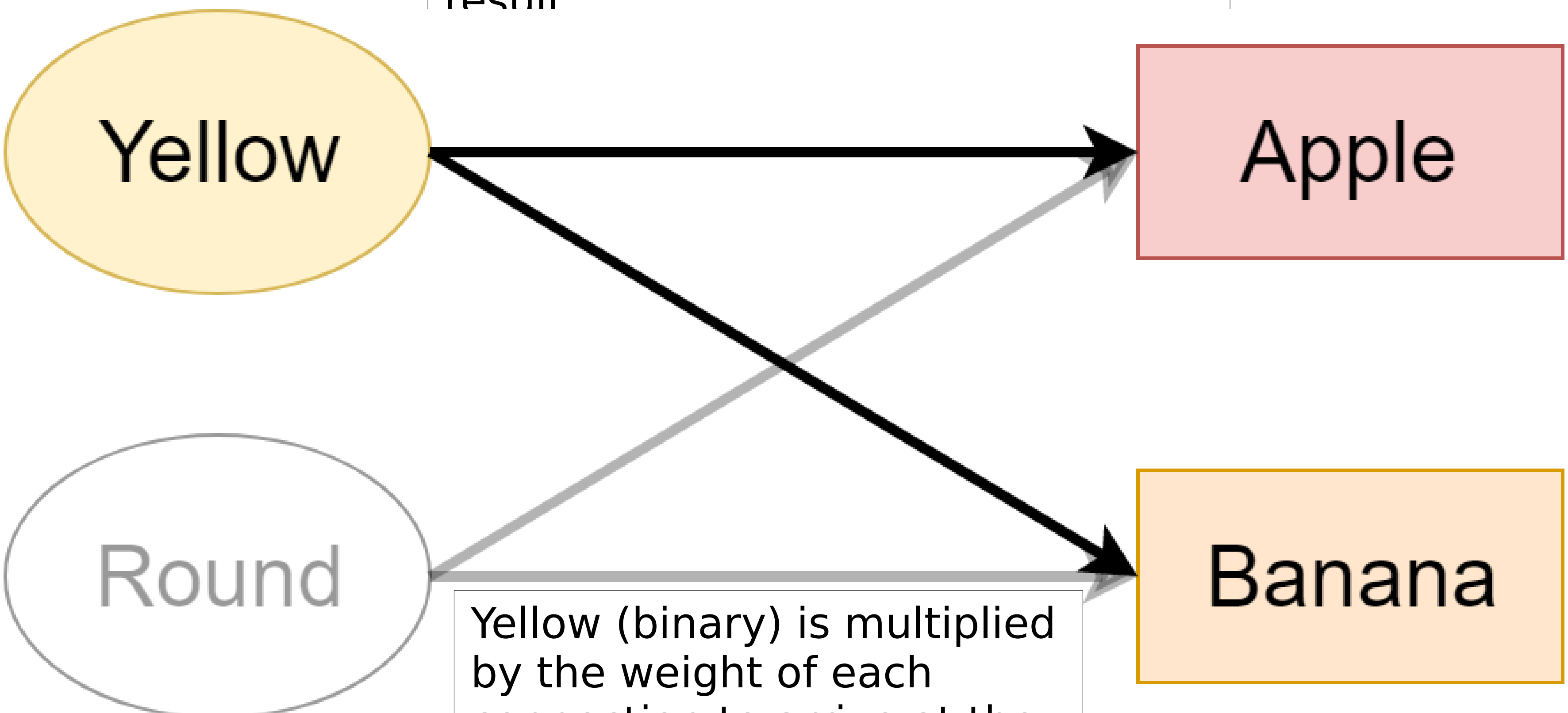
Banana or Apple?



Training the Network!

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result

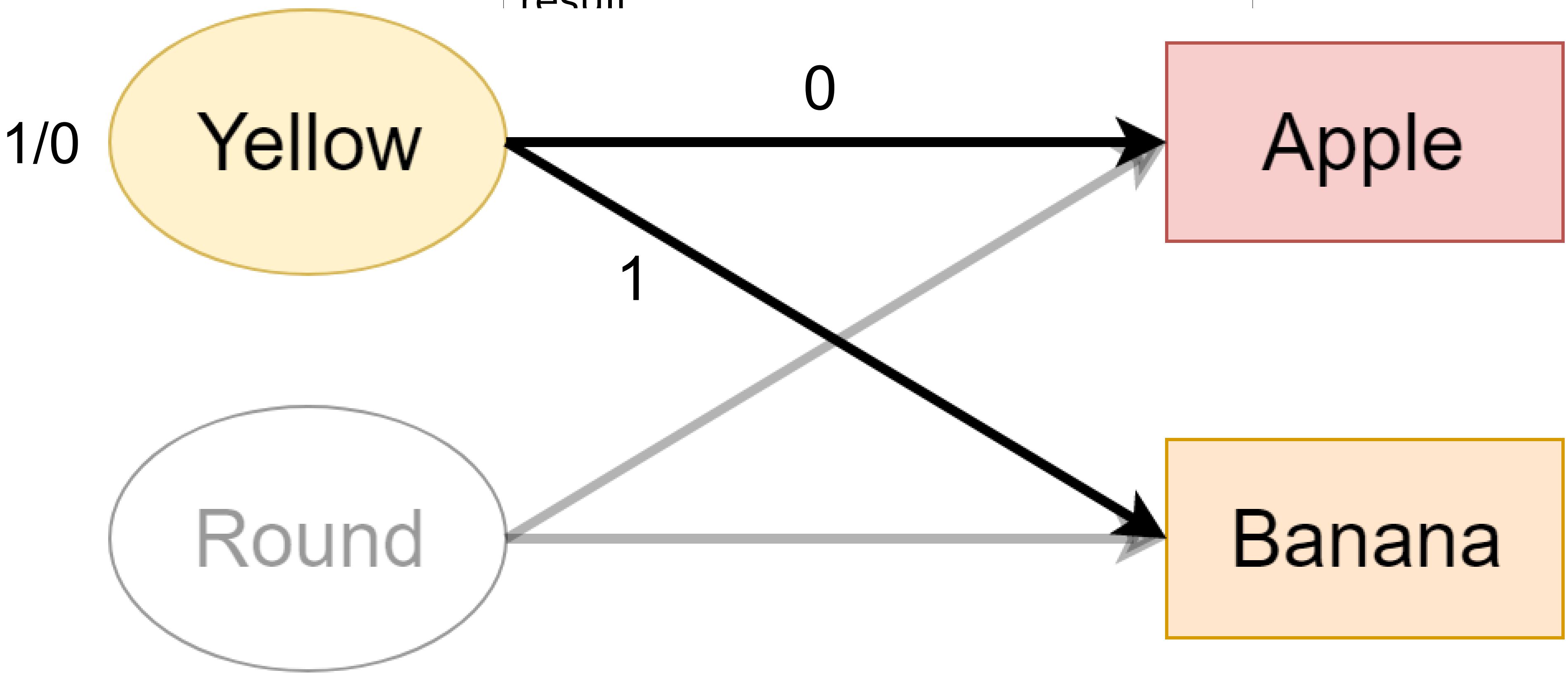


Yellow (binary) is multiplied by the weight of each connection to arrive at the next layer

Training the Network!

Banana or Apple?

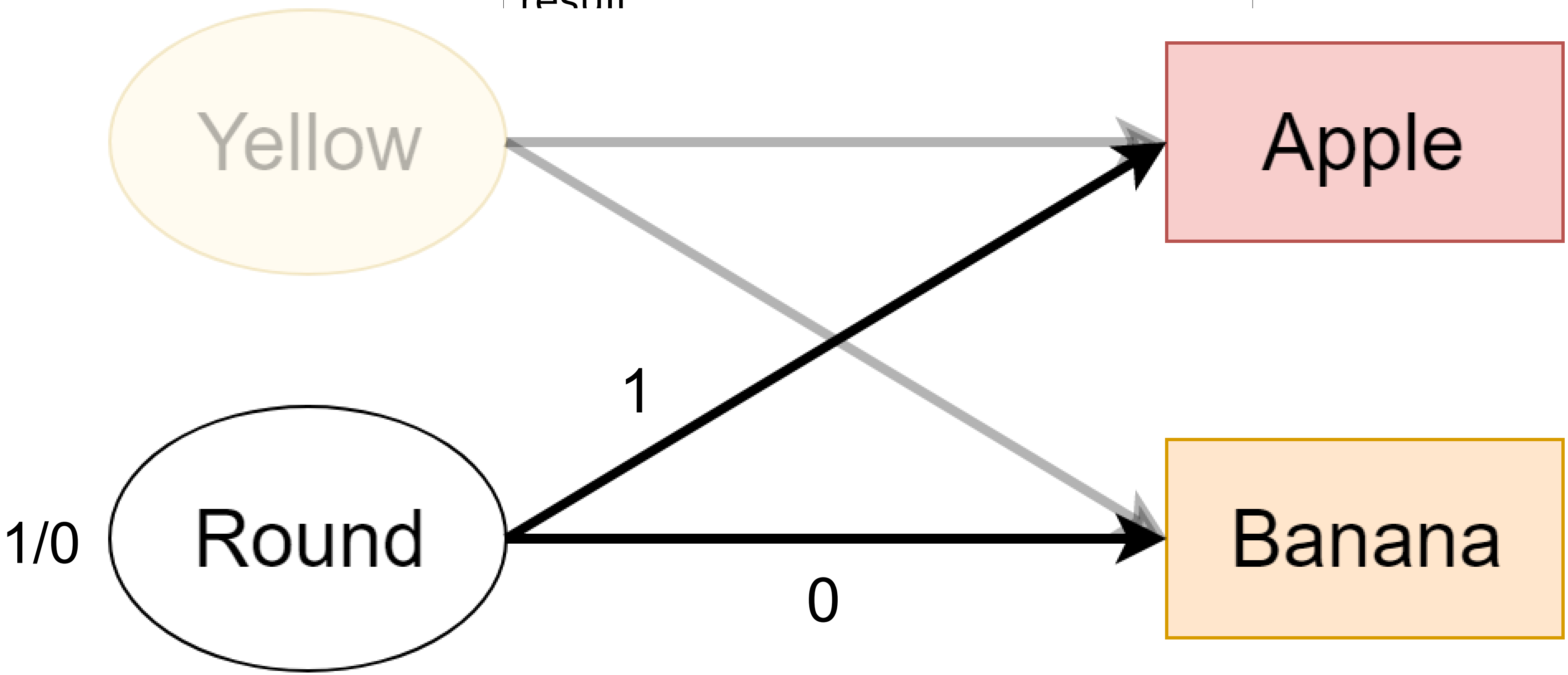
Strength of connection (Weight) shows importance of parameter to result



Training the Network!

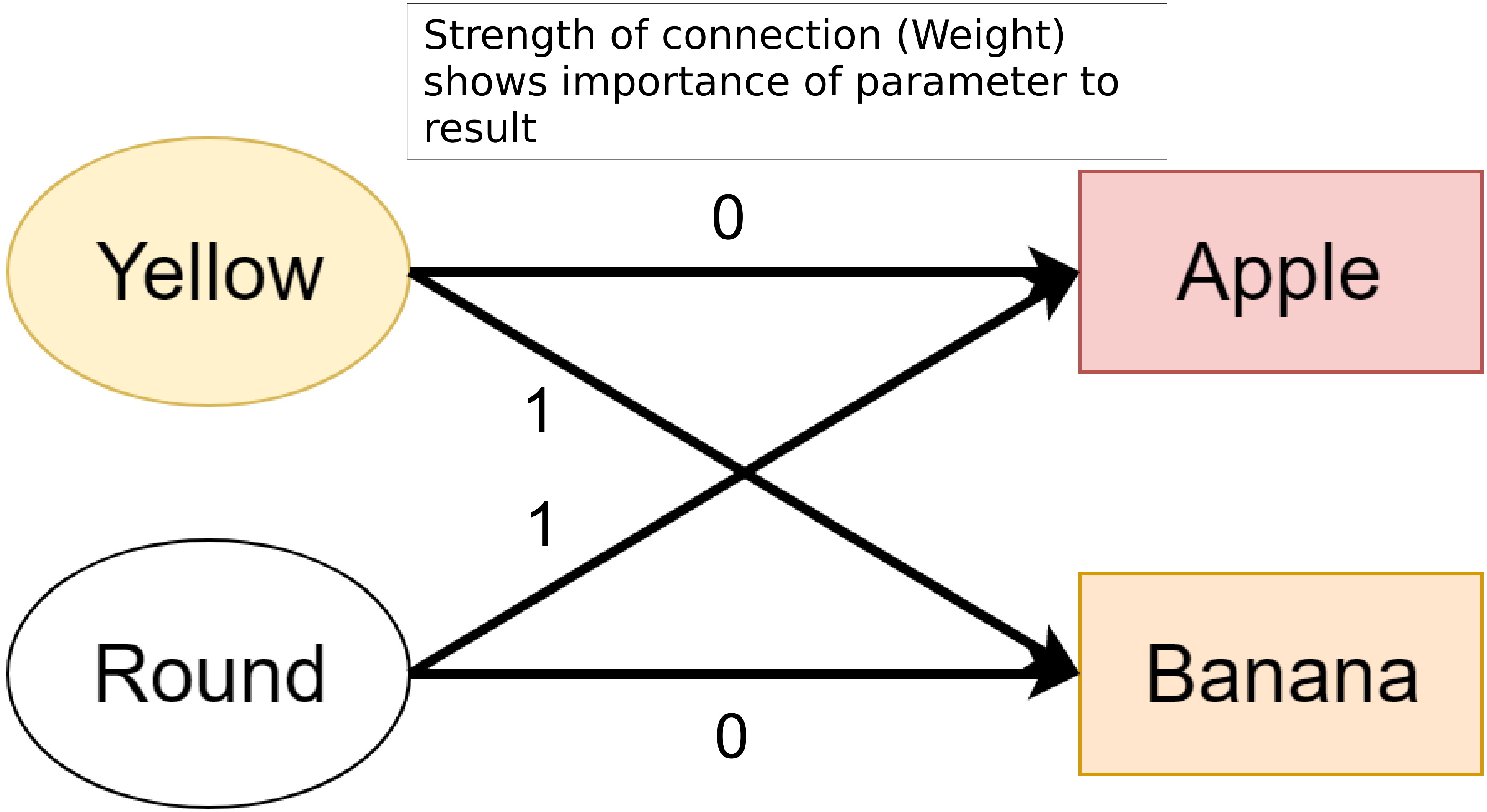
Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result



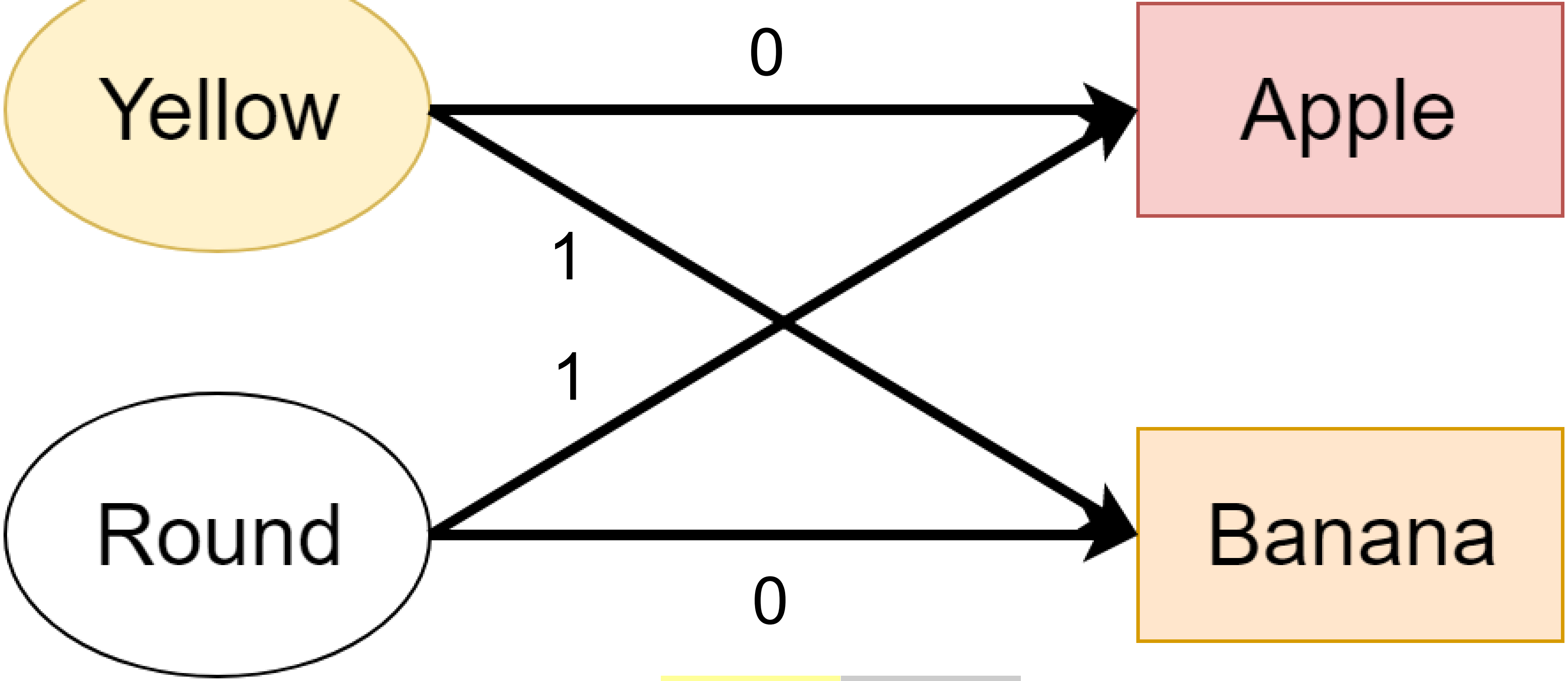
Training the Network!

Banana or Apple?



Banana or Apple?

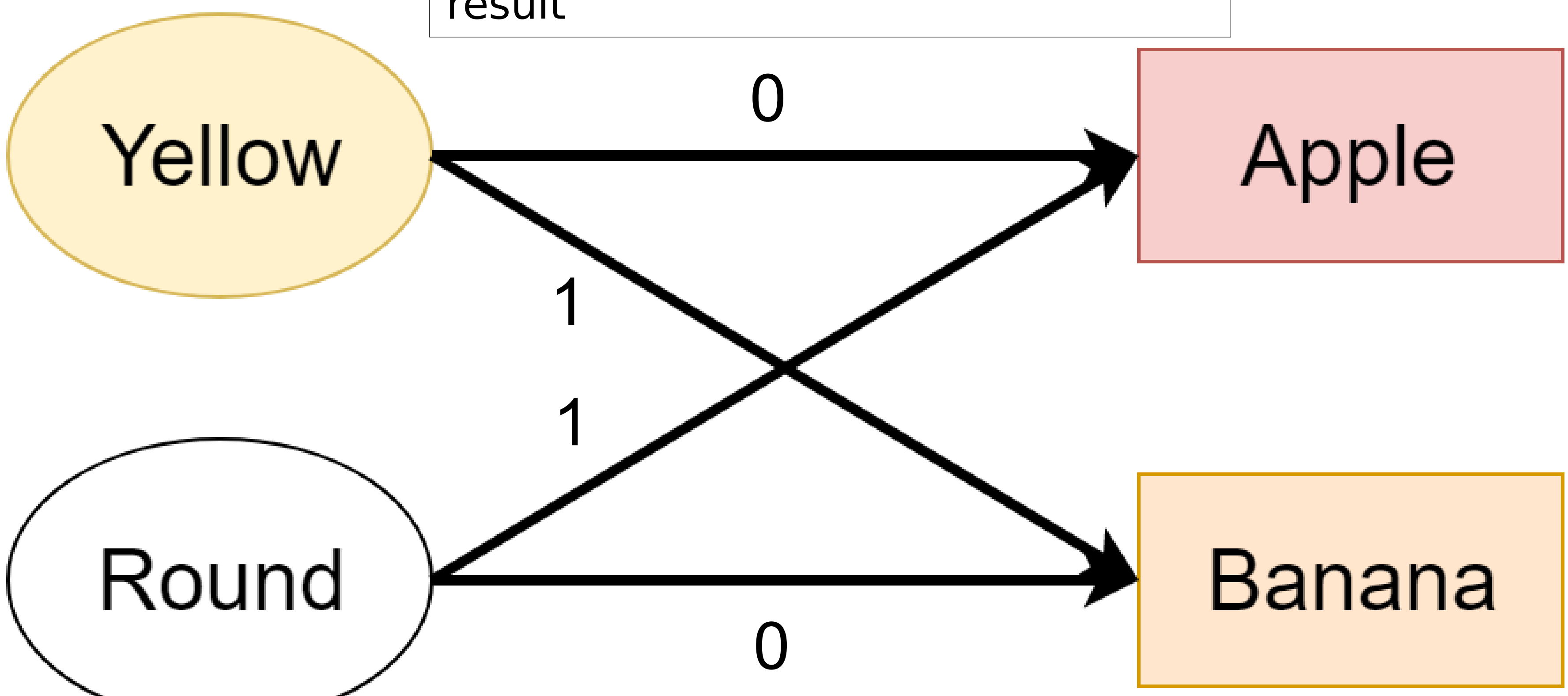
Strength of connection (Weight) shows importance of parameter to result



	Yellow	Round
Apple	0	1
Banana	1	0

Banana or Apple?

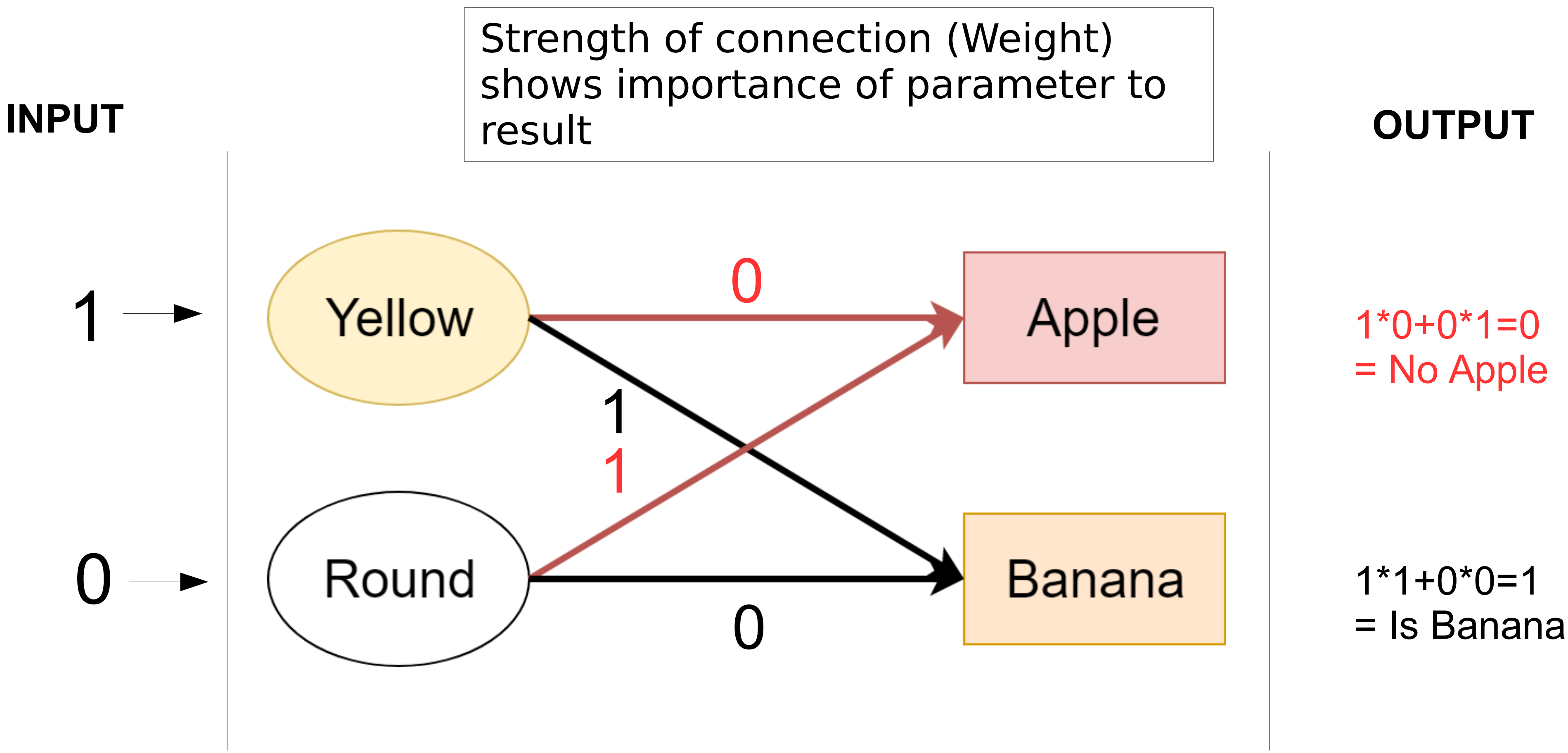
Strength of connection (Weight) shows importance of parameter to result



	Yellow	Round
Apple	0	1
Banana	1	0

Apple = Yellow * 0 + Round *1
Banana = Yellow * 1 + Round *0

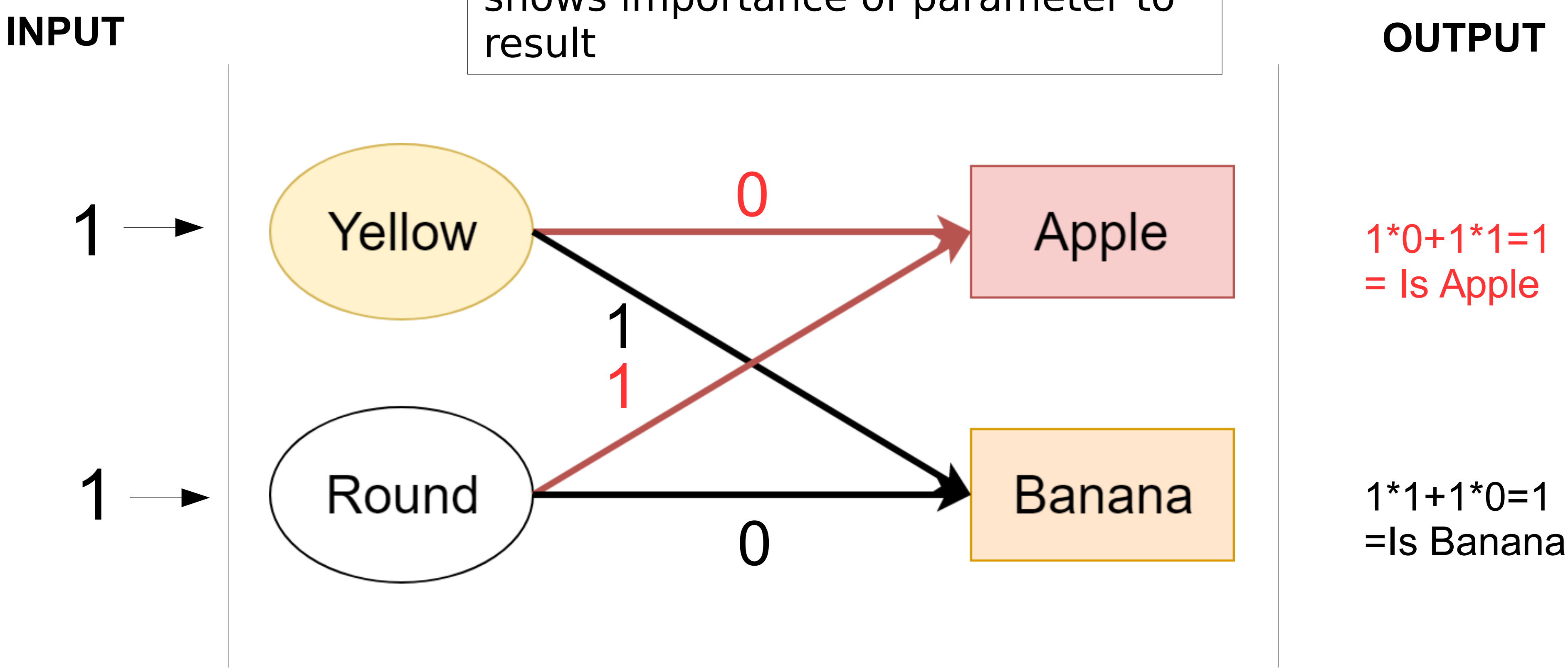
Banana or Apple?



Apple = Yellow * 0 + Round *1
Banana = Yellow * 1 + Round *0

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result



Our model is too simple

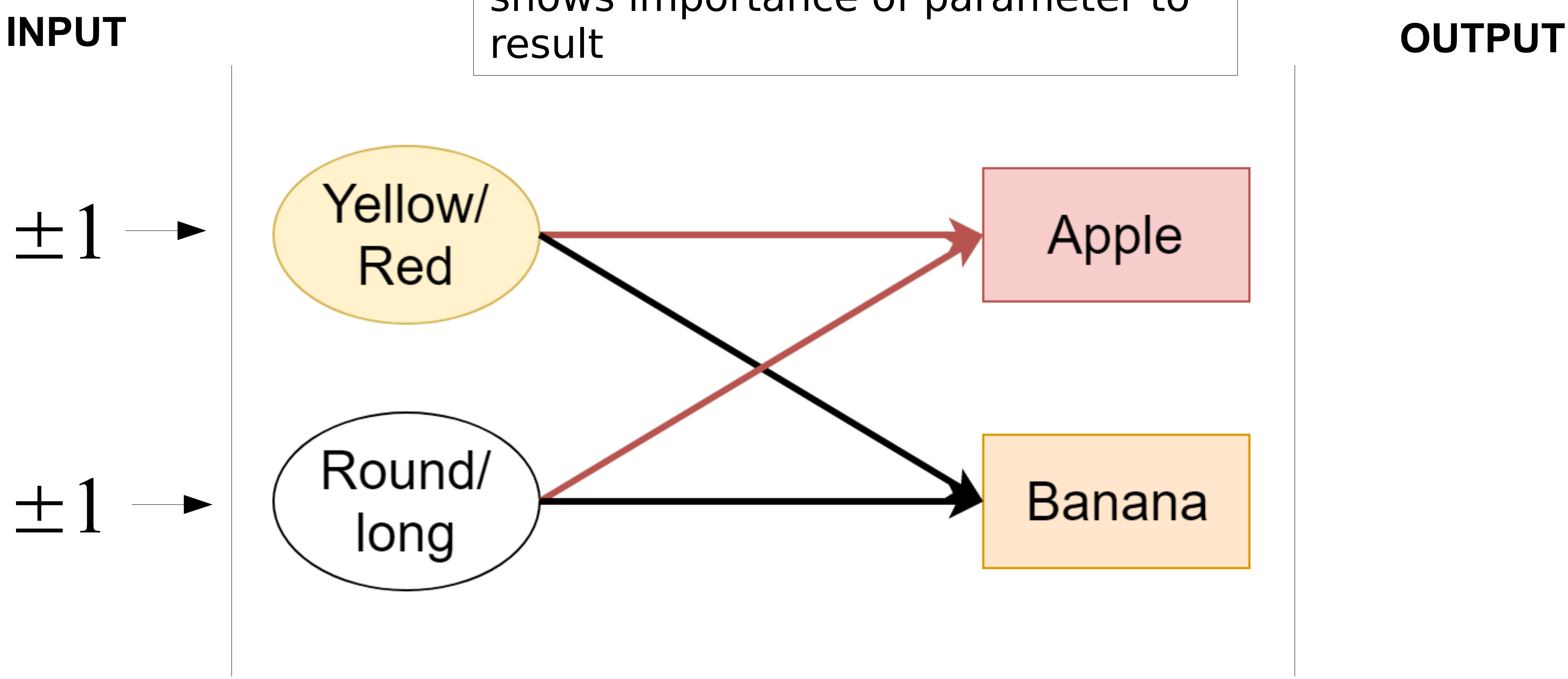
Apple = Yellow * 0 + Round *1
Banana = Yellow * 1 + Round *0

Solution:

- Expand inputs and weight range
- Hopefully this will add the needed Complexity to the model

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result

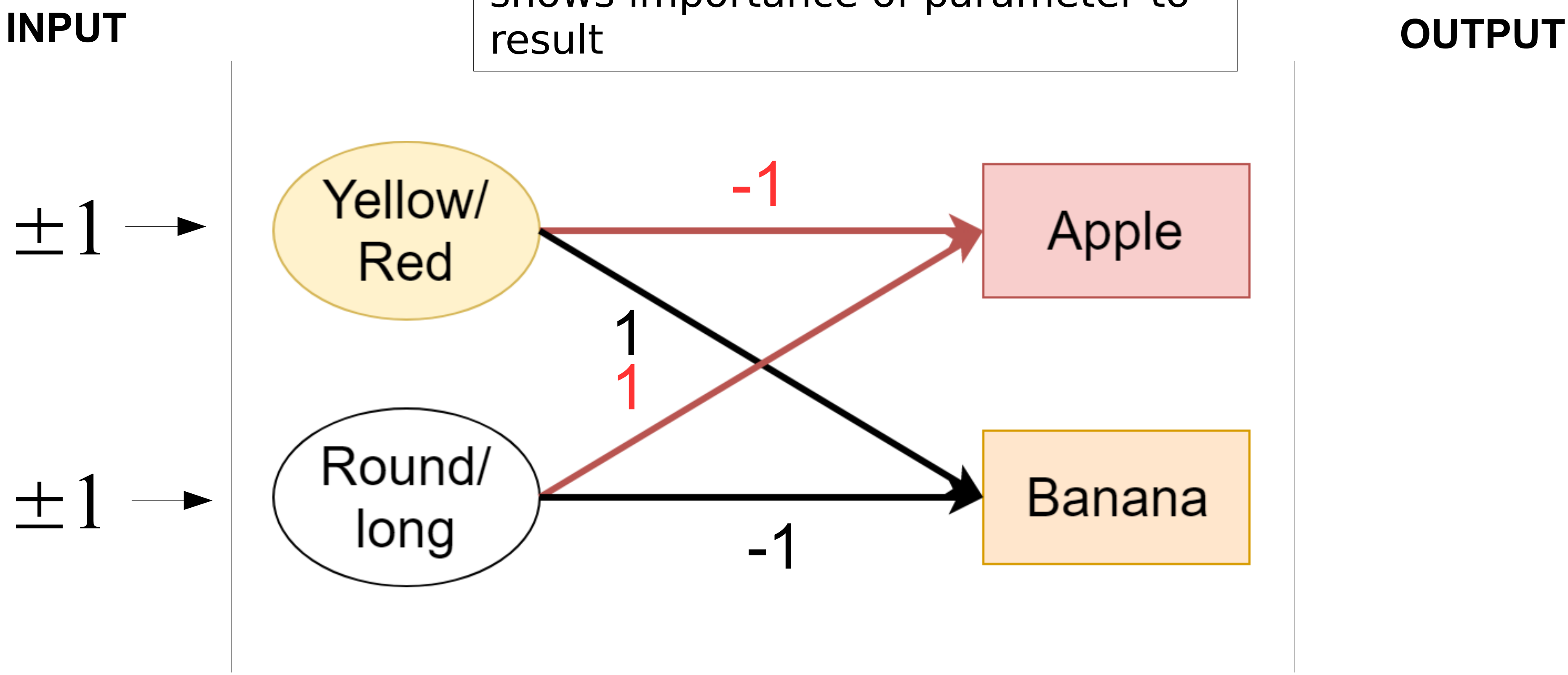


+1 is Yellow or Round like before,
-1 is Green or Long,
0 is lack of data.

Value	Input 1	Input 2
+1	Yellow	Round
-1	Red	Long
0	No Data	No Data

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result



+1 is Yellow or Round like before,
-1 is Green or Long,
0 is lack of data.

Banana or Apple?

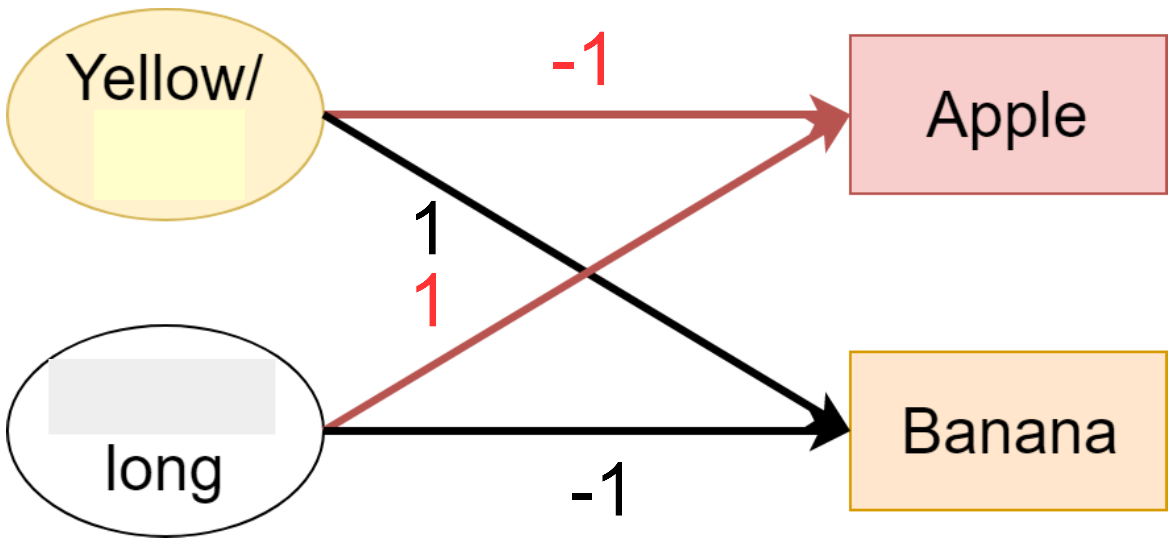
Strength of connection (Weight) shows importance of parameter to result

INPUT

OUTPUT

$+1 \rightarrow$

$-1 \rightarrow$



1*-1+-1*1=-2
=> very not apple

1*1+-1*-1=2
=> very banana

+1 is Yellow or Round like before,
-1 is Green or Long,
0 is lack of data.

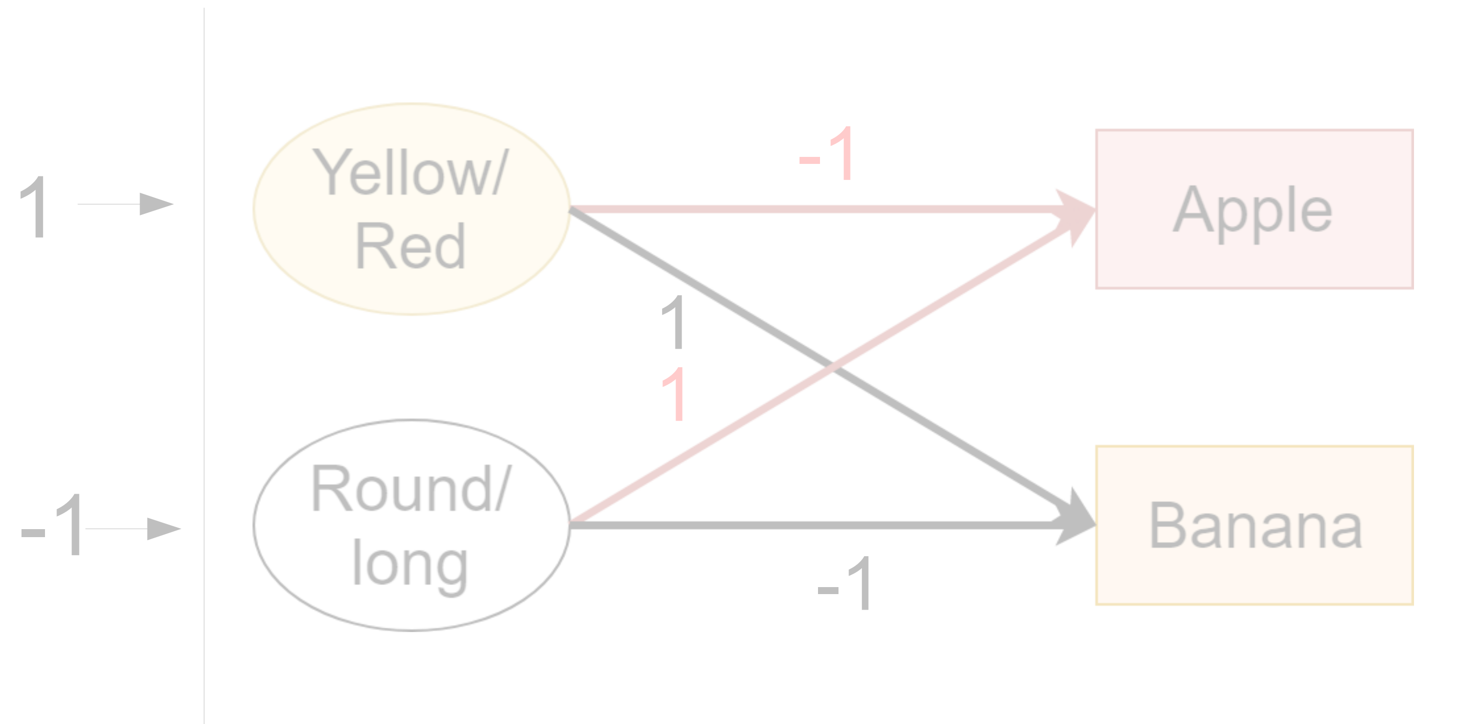
Apple = Yellow * 0 + Round *1
Banana = Yellow * 1 + Round *0

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result

INPUT

OUTPUT



1*-1+-1*1=-2
=> very not apple

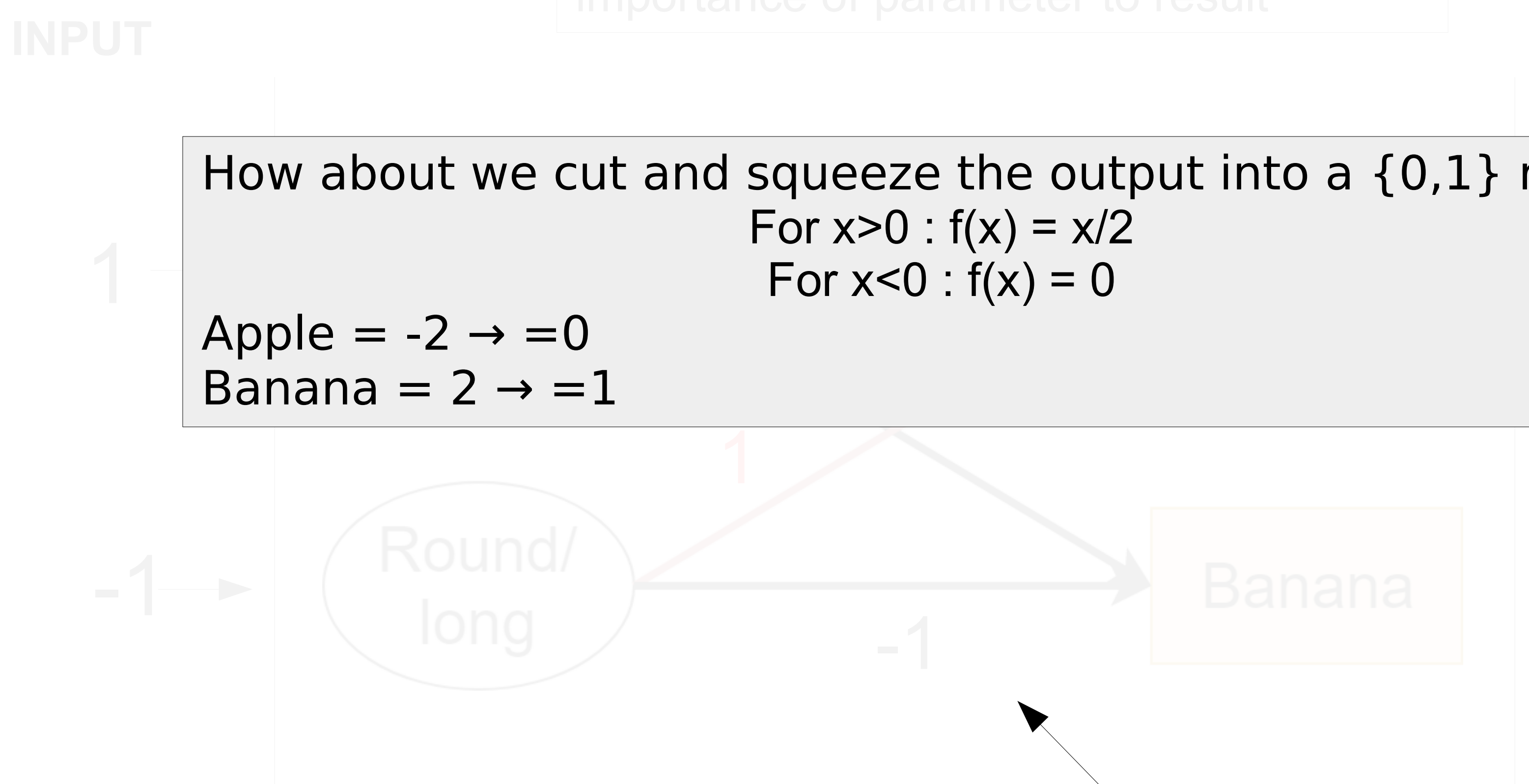
1*1+-1*-1=2
=> very banana

+1 is Yellow or Round like before,
-1 is Green or Long,
0 is lack of data.

Normalize to {0,1} range
to ease comparison

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result



How about we cut and squeeze the output into a {0,1} range?

For $x>0$: $f(x) = x/2$

For $x<0$: $f(x) = 0$

Apple = -2 → =0

Banana = 2 → =1

+-1*1=-2
very not
apple

1*1+-1*-1=2
=> very
banana

+1 is Yellow or Round like before,
-1 is Green or Long,
0 is lack of data.

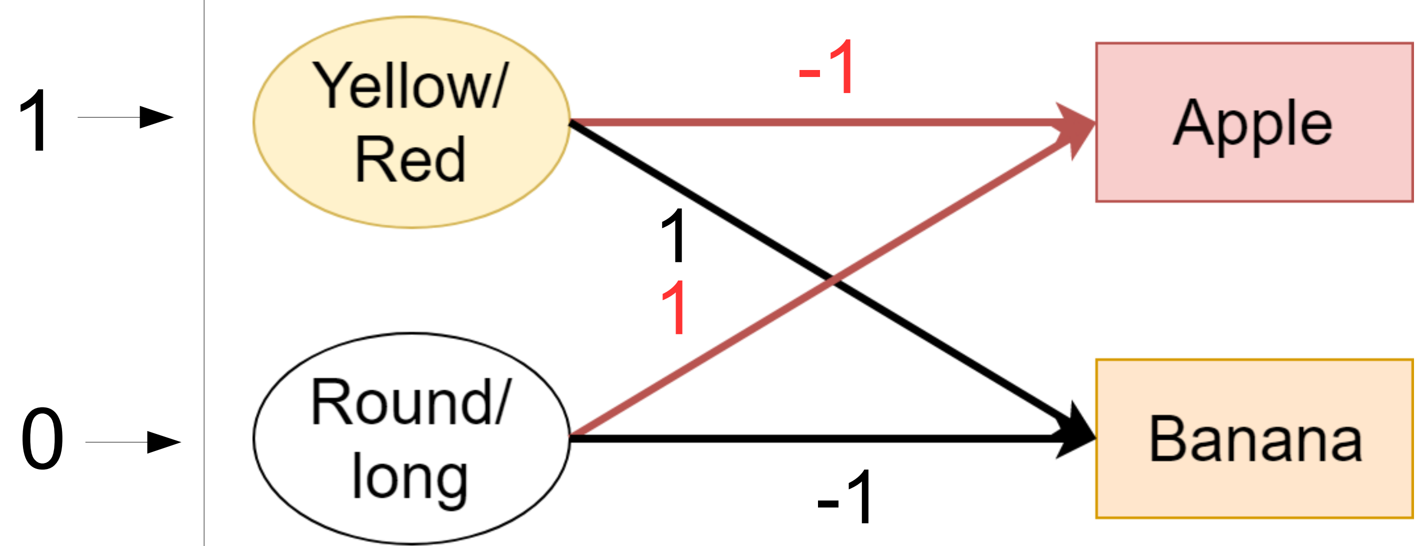
Normalize to {0,1} range
to ease comparison

Banana or Apple?

Strength of connection (Weight) shows importance of parameter to result

INPUT

OUTPUT
f(x)=abs(x/2)



1*-1+0*1=-1
f(-1)=0
=>not apple

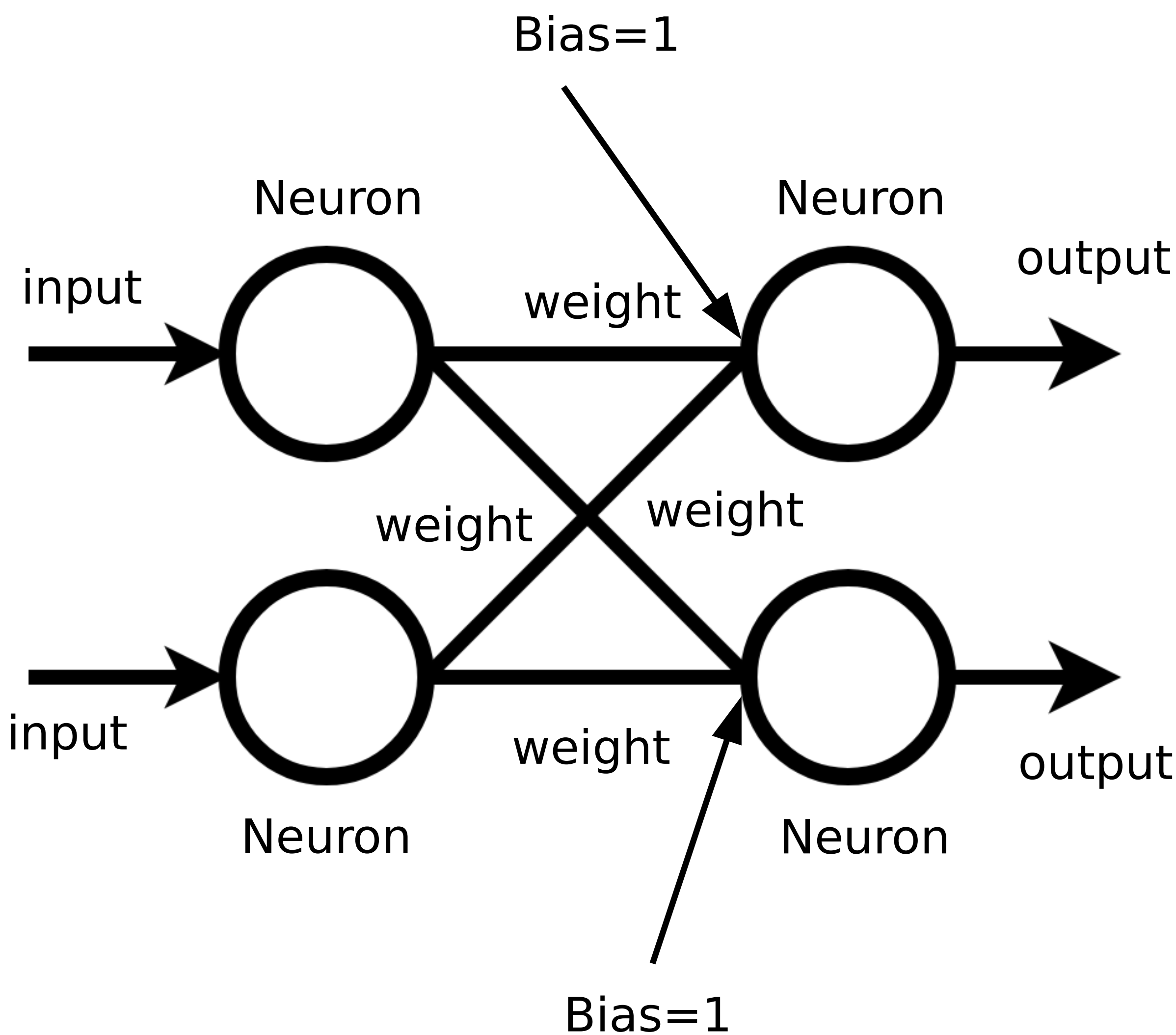
1*1+0*-1=1
F(1)=0.5
=> 50% banana

Apple = Yellow * 0 + Round *1
Banana = Yellow * 1 + Round *0

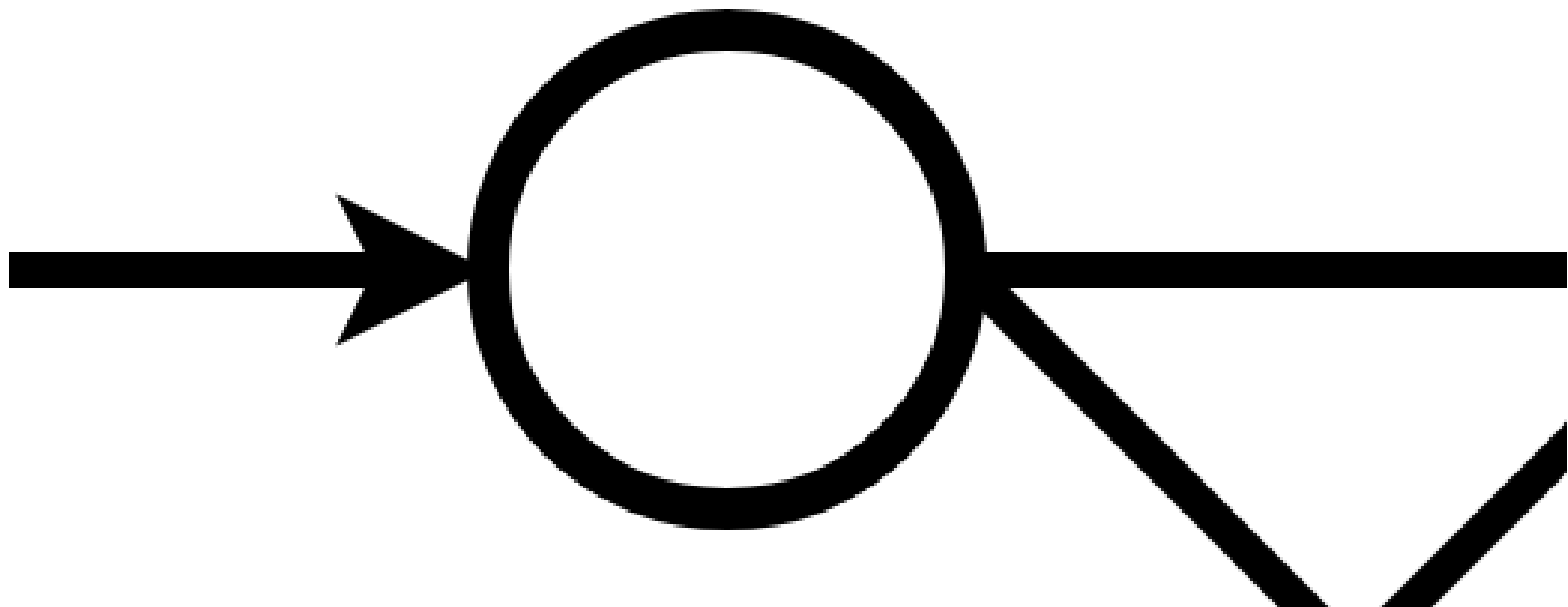
=>

For x>0 : f(x) = x/2
For x<0 : f(x) = 0

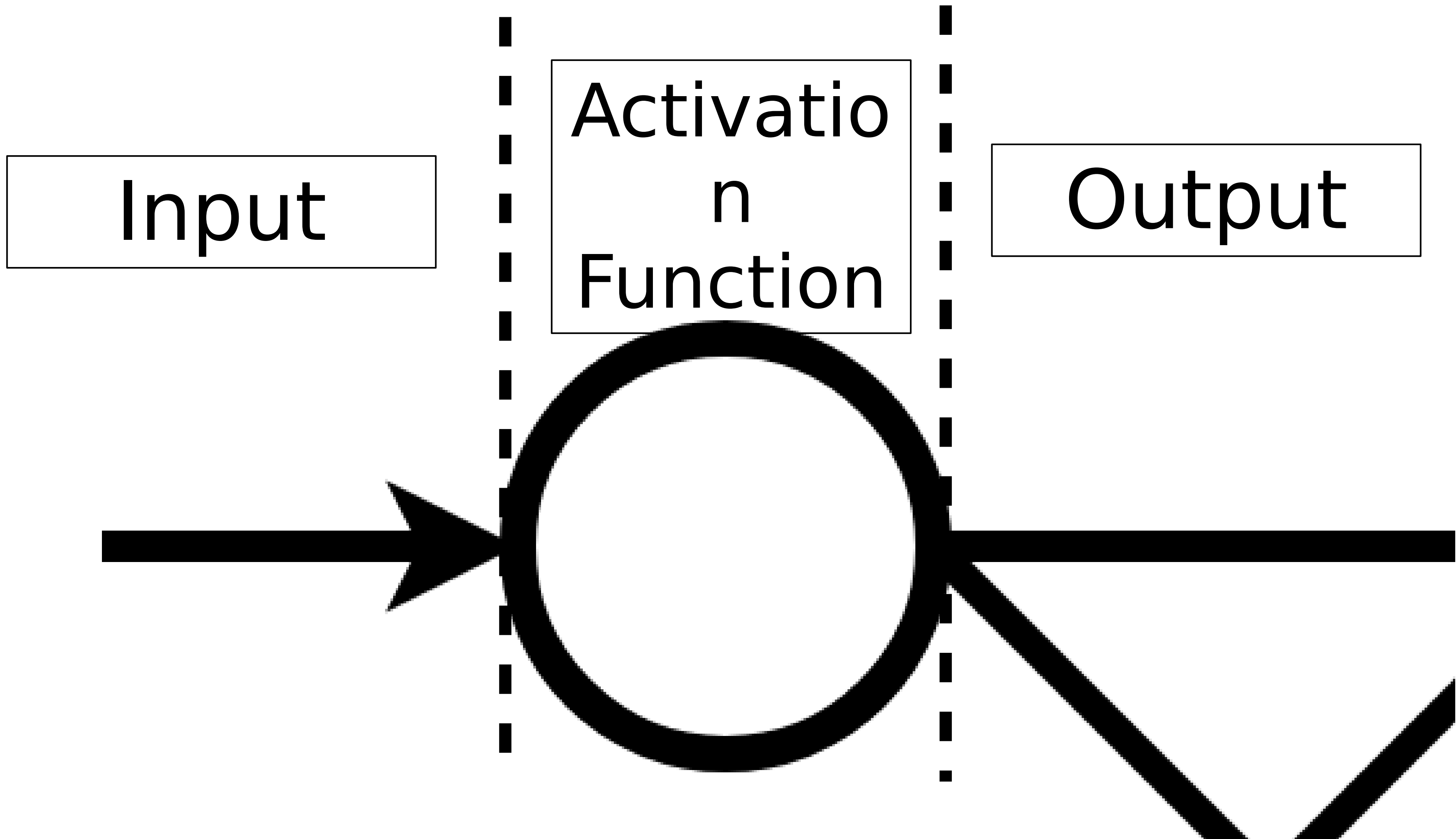
Multilayer Perceptrons



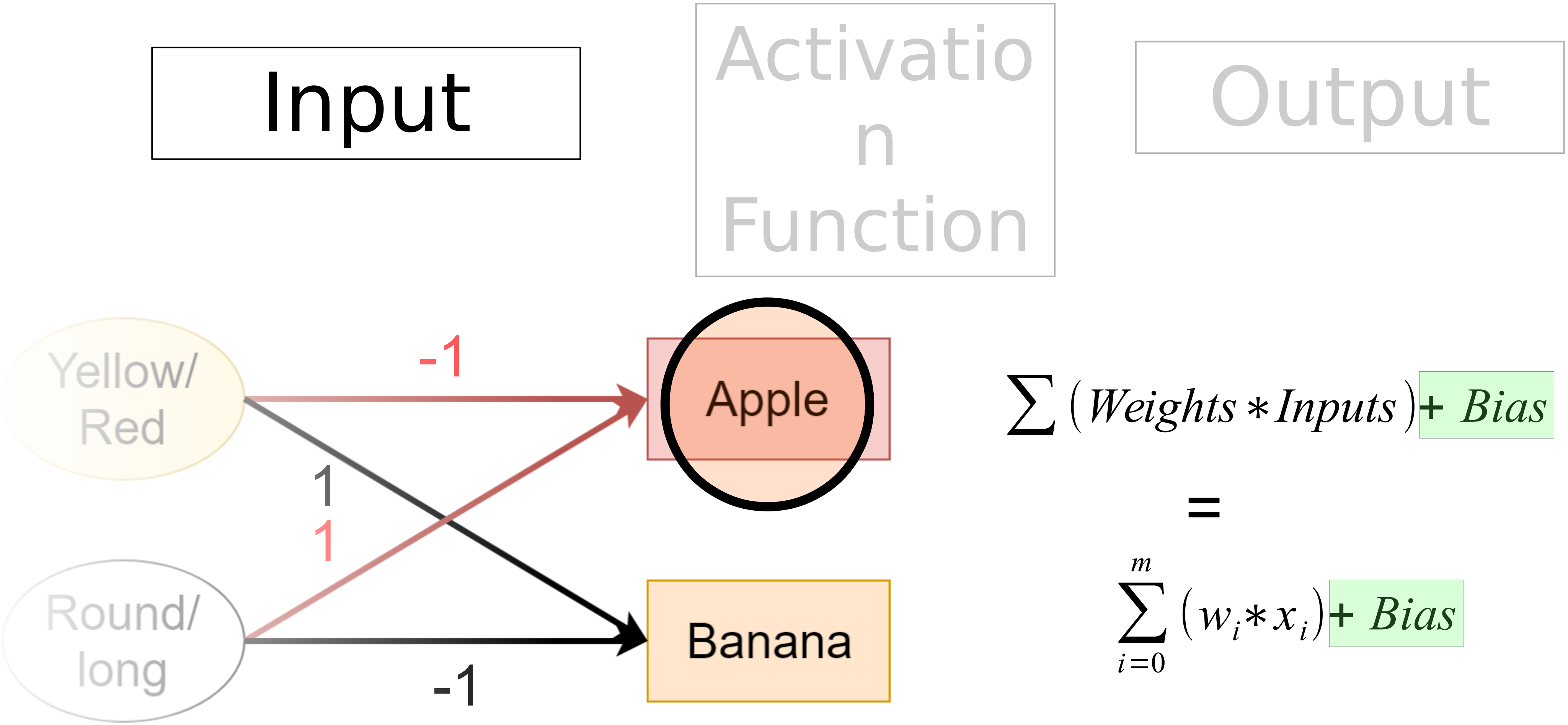
Multilayer Perceptrons



Multilayer Perceptrons



Multilayer Perceptrons



$$\sum\left(Weights * Inputs\right) + Bias$$

=

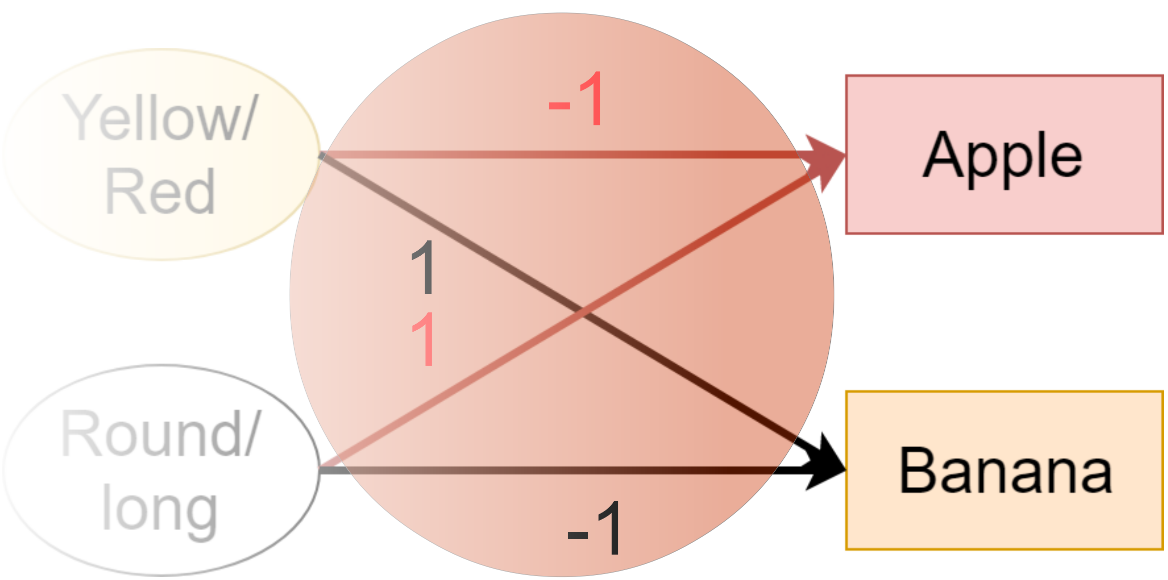
$$\sum_{i=0}^{m}\left(w_i * x_i\right) + Bias$$

Multilayer Perceptrons

Input

Activation Function

Output



1*-1+-1*1=-2
=> very not apple

1*1+-1*-1=2
=> very banana

$$\sum(Weights * Inputs)$$

=

$$\sum_{i=0}^{m}(w_i * x_i)$$

Multilayer Perceptrons

*Image credits here

Input

Activation Function

Output

How about we cut and squeeze the output into a {0,1} range?

For x>0 : f(x) = x/2

For x<0 : f(x) = 0

Apple = -2 → =0

Banana = 2 → =1

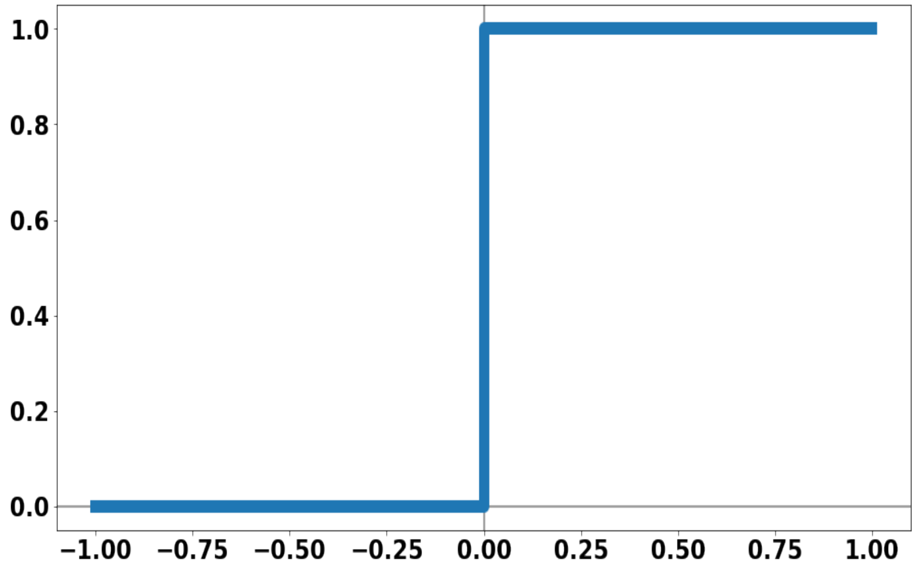
$$f(Inputs)$$

Multilayer Perceptrons

Input

Activation Function

Output



Step function (Heaviside)

$$f(x) = \begin{cases} 0 \ for \ x \leqslant 0 \\ 1 \ for \ x > 0 \end{cases}$$

1*-1+-1*1=-2
=> very not Apple
=> f(x)=0

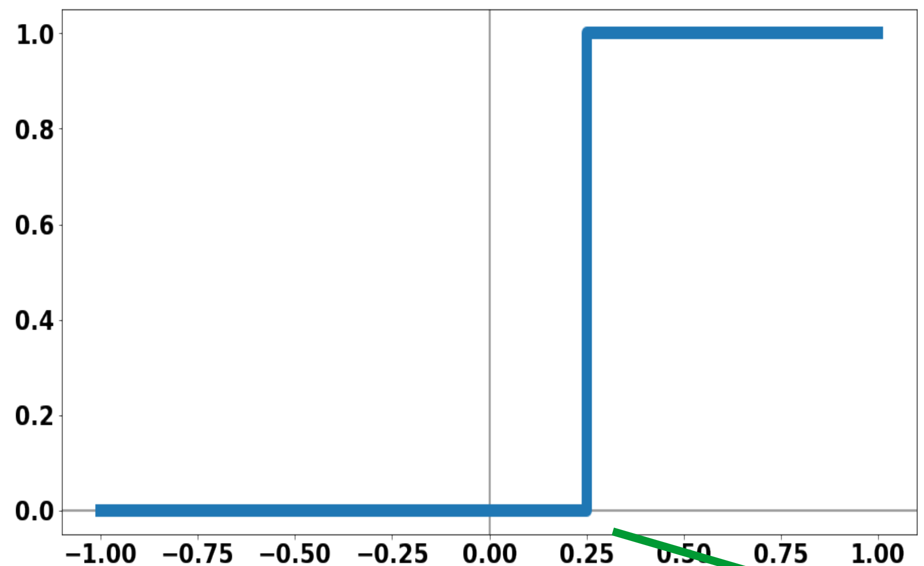
1*1+-1*-1=2
=> very Banana
=> f(x)=1

Multilayer Perceptrons

Input

Activation Function

Output



Step function (Heaviside)

Bias

$$f(x) = \begin{cases} 0 \ for \ x \leqslant 0 + 0.25 \\ 1 \ for \ x > 0 + 0.25 \end{cases}$$

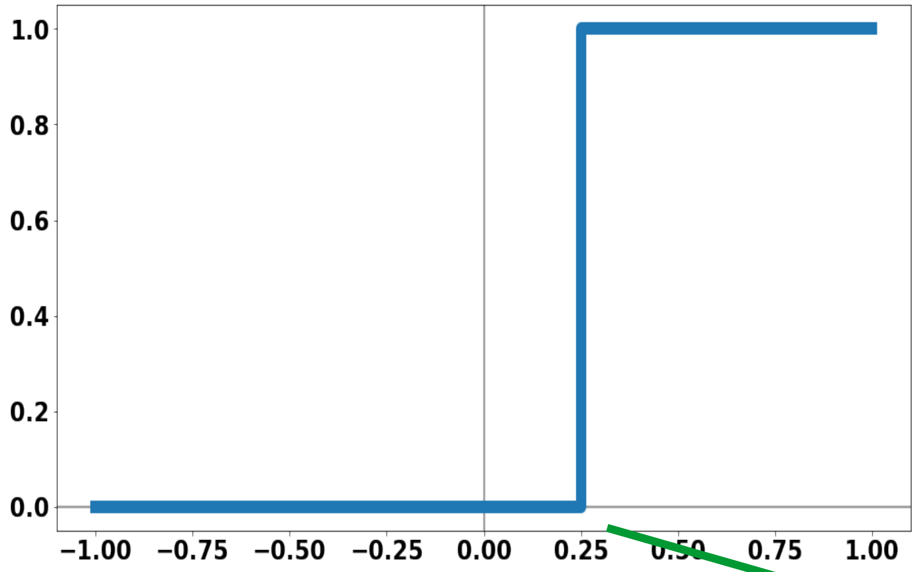
$$f(x) = step\ function(x - 0.25)$$

Multilayer Perceptrons

Input

Activation Function

Output



Step function (Heaviside)

$$Bias = 1 * w_b$$

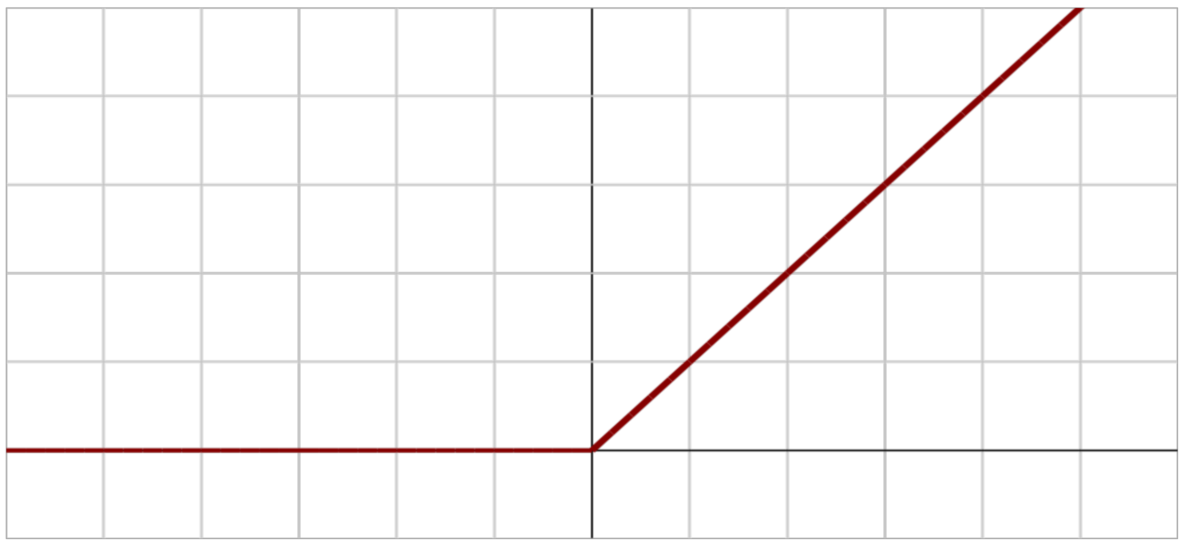
$$f(x) = \begin{cases} 0 \ for \ x \leqslant 0 \\ 1 \ for \ x > 0 \end{cases} + 0.25$$

Multilayer Perceptrons

Input

Activation Function

Output



RELU (Rectified Linear Units)

$$f(x) = \begin{cases} 0 \; for \; x \leqslant 0 \\ x \; for \; x > 0 \end{cases}$$

1*-1+-1*1=-2
=> very not Apple
=> f(x)=0

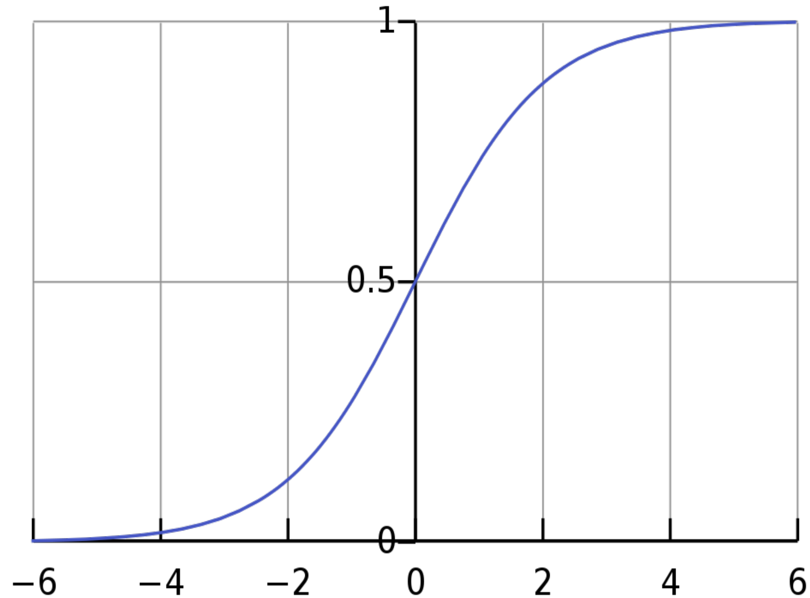
1*1+-1*-1=2
=> very Banana
=> f(x)=2

Multilayer Perceptrons

Input

Activation Function

Output



Sigmoid function (logistic function)

$$f(x) = \frac{1}{1+e^{-x}}$$

1*-1+-1*1=-2
=> very not Apple
=> f(x)=0.1192

1*1+-1*-1=2
=> very Banana
=> f(x)=0.8807

...aka Softstep..

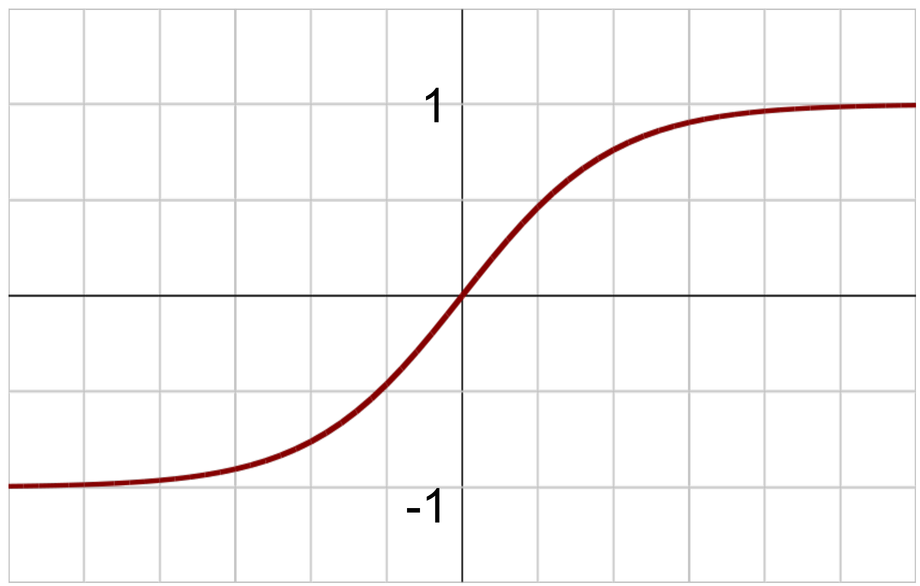
...aka Logistic Curve...

Multilayer Perceptrons

Input

Activation Function

Output



Tanh function

$$f(x) = \frac{2}{1+e^{-2x}} - 1 = 2\,sigmoid(2x) - 1$$

1*-1+-1*1=-2
=> very not Apple
=> f(x)=-0.964

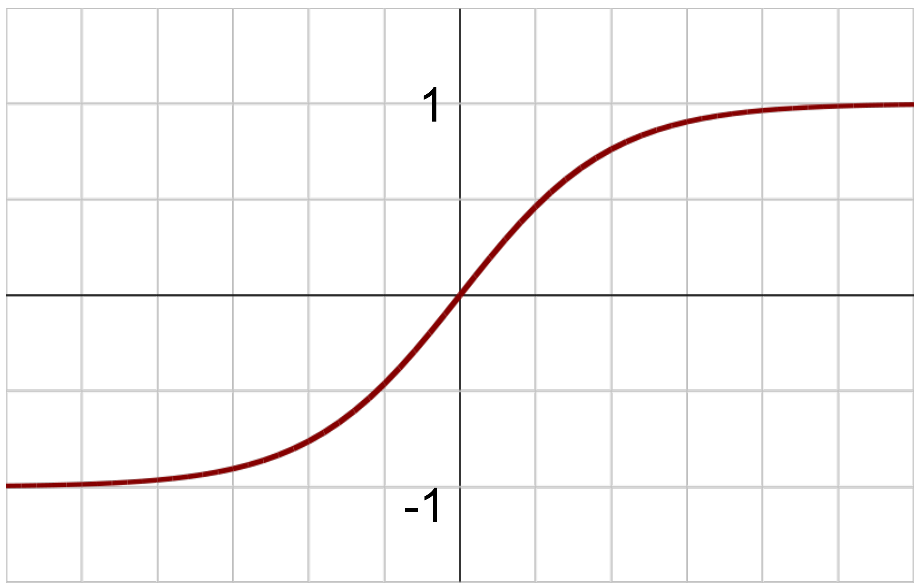
1*1+-1*-1=2
=> very Banana
=> f(x)=0.964

Multilayer Perceptrons

Input

Activation Function

Output



Tanh function

1*-1+-1*1=-2
=> very not apple

1*1+-1*-1=2
=> very banana

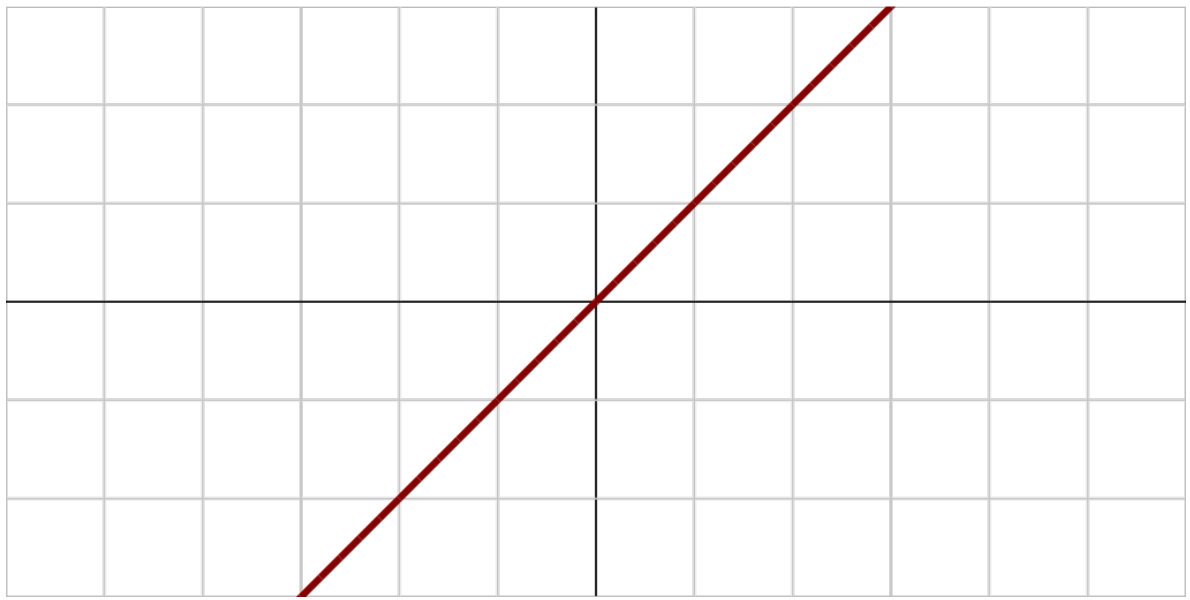
Range {-1,1} might be used for when you want to make negative predictions, such as Yellow + Long has not just a zero chance of being apple, but a negative chance.

Multilayer Perceptrons

Input

Activation Function

Output



Identity function

$$f(x)=x$$

1*-1+-1*1=-2
=> very not apple

1*1+-1*-1=2
=> very banana

A Lot'a Functions

- Monotonic – When the activation function is monotonic, the error surface associated with a single-layer model is guaranteed to be convex.[4]
- Smooth Functions with a Monotonic derivative – These have been shown to generalize better in some cases. The argument for these properties suggests that such activation functions are more consistent with Occam's razor.[5]
- Approximates identity near the origin – When activation functions have this property, the neural network will learn efficiently when its weights are initialized with small random values. When the activation function does not approximate identity near the origin, special care must be used when initializing the

The following table compares the properties of several activation functions that are functions of one fold x from the previous layer or layers:

Name	Plot	Equation	Derivative (with respect to x)	Range	Order of continuity	Monotonic	Derivative Monotonic	Approximates identity near the origin
Identity		$f(x)=x$	$f'(x)=1$	$(-\infty,\infty)$	C^∞	Yes	Yes	Yes
Binary step		$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x\neq 0\\ ? & \text{for } x=0\end{cases}$	$\{0,1\}$	C^{-1}	Yes	No	No
Logistic (a.k.a. Soft step)		$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$	$(0,1)$	C^∞	Yes	No	No
TanH		$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$	$(-1,1)$	C^∞	Yes	No	Yes
ArcTan		$f(x)=\tan^{-1}(x)$	$f'(x)=\frac{1}{x^2+1}$	$\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$	C^∞	Yes	No	Yes
Softsign[7][8]		$f(x)=\frac{x}{1+\lvert x\rvert}$	$f'(x)=\frac{1}{(1+\lvert x\rvert)^2}$	$(-1,1)$	C^1	Yes	No	Yes
Inverse square root unit (ISRU)[9]		$f(x)=\frac{x}{\sqrt{1+\alpha x^2}}$	$f'(x)=\left(\frac{1}{\sqrt{1+\alpha x^2}}\right)^3$	$\left(-\frac{1}{\sqrt{\alpha}},\frac{1}{\sqrt{\alpha}}\right)$	C^∞	Yes	No	Yes
Rectified linear unit (ReLU)[10]		$f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	$[0,\infty)$	C^0	Yes	Yes	No
Leaky rectified linear unit (Leaky ReLU)[11]		$f(x)=\begin{cases}0.01x & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0.01 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	$(-\infty,\infty)$	C^0	Yes	Yes	No
Parameteric rectified linear unit (PReLU)[12]		$f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'($[illegible] for $x<0$, for $x\geq 0$	$(-\infty,\infty)$	C^0	Yes iff $\alpha\geq 0$	Yes	Yes iff $\alpha=1$
Randomized leaky rectified linear unit (RReLU)[13]		$f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$ [1]	$f'(\alpha,$[illegible] $x<0$, ≥ 0	$(-\infty,\infty)$	[illegible]	Yes	Yes	No
Exponential linear unit (ELU)[14]		$f(\alpha,x)=\begin{cases}\alpha(e^x-1) & \text{[illegible]}\\ x & \text{[illegible]} x\geq\end{cases}$	$f'(\alpha,x)=\begin{cases}f(\alpha,x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	[illegible]	[illegible]en $\alpha=1$; C_0 [illegible]erwise	Yes iff $\alpha\geq 0$	Yes iff $0\leq\alpha\leq 1$	Yes iff $\alpha=1$
Scaled exponential linear unit (SELU)[15]		$f(\alpha,x)=$ [illegible] $\alpha(e^x-1)$ [illegible] x; with $\lambda=1$[illegible] and [illegible] 1.67326[illegible]	$f'(\alpha,x)=$ [illegible]	[illegible]$,\infty)$	[illegible]	Yes	No	No
S-shaped rectified linear activation unit (SReLU)[16]		$f_{t_l,a_l,t_r,a_r}(x)=$ [illegible] $(x-t_l)$ for $x\leq t_l$; for $t_l<x<t_r$; $t_r+a_r(x-t_r)$ for $x\geq t_r$; [illegible] are parameters.	[illegible] $(x)=$ [illegible] for $t_l<x$ [illegible]	$(-\infty,\infty)$	C^0	No	No	No
Inverse square root linear unit (ISRLU)[9]		$f(x)=\begin{cases}\frac{x}{\sqrt{1+\alpha x^2}} & \text{for } x<0\\ x & \text{for [illegible]}\end{cases}$	[illegible]$(x)=$ [illegible] for $x<0$; 1 for $x\geq 0$	$\left(-\frac{1}{\sqrt{\alpha}},\infty\right)$	C^2	Yes	Yes	Yes
Adaptive piecewise linear (APL)[17]		[illegible]	$f'(x)=H(x)-\sum_{s=1}^{S}a_i^s H(-x+b_i^s)$ [2]	$(-\infty,\infty)$	C^0	No	No	No
SoftPlus[18]		[illegible] $\ln($[illegible]$e^x)$	$f'(x)=\frac{1}{1+e^{-x}}$	$(0,\infty)$	C^∞	Yes	Yes	No
Bent identity		[illegible]$)=\frac{\sqrt{x^2+1}-1}{2}+x$	$f'(x)=\frac{x}{2\sqrt{x^2+1}}+1$	$(-\infty,\infty)$	C^∞	Yes	Yes	Yes
SoftExponential[19]		$f(\alpha,x)=\begin{cases}-\frac{\ln(1-\alpha(x+\alpha))}{\alpha} & \text{for } \alpha<0\\ x & \text{for } \alpha=0\\ \frac{e^{\alpha x}-1}{\alpha}+\alpha & \text{for } \alpha>0\end{cases}$	$f'(\alpha,x)=\begin{cases}\frac{1}{1-\alpha(\alpha+x)} & \text{for } \alpha<0\\ e^{\alpha x} & \text{for } \alpha\geq 0\end{cases}$	$(-\infty,\infty)$	C^∞	Yes	Yes	Yes iff $\alpha=0$
Sinusoid[20]		$f(x)=\sin(x)$	$f'(x)=\cos(x)$	$[-1,1]$	C^∞	No	No	Yes
Sinc		$f(x)=\begin{cases}1 & \text{for } x=0\\ \frac{\sin(x)}{x} & \text{for } x\neq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x=0\\ \frac{\cos(x)}{x}-\frac{\sin(x)}{x^2} & \text{for } x\neq 0\end{cases}$	$[\approx -.217234, 1]$	C^∞	No	No	No
Gaussian		$f(x)=e^{-x^2}$	$f'(x)=-2xe^{-x^2}$	$(0,1]$	C^∞	No	No	No

^ Here, H is the Heaviside step function.

Multilayer Perceptrons

Input

Activatio
n
Function

Output

$$f\left(\sum_{i=0}^{m}\left(w_i * x_i\right) + Bias\right)$$

Multilayer Perceptrons

Input

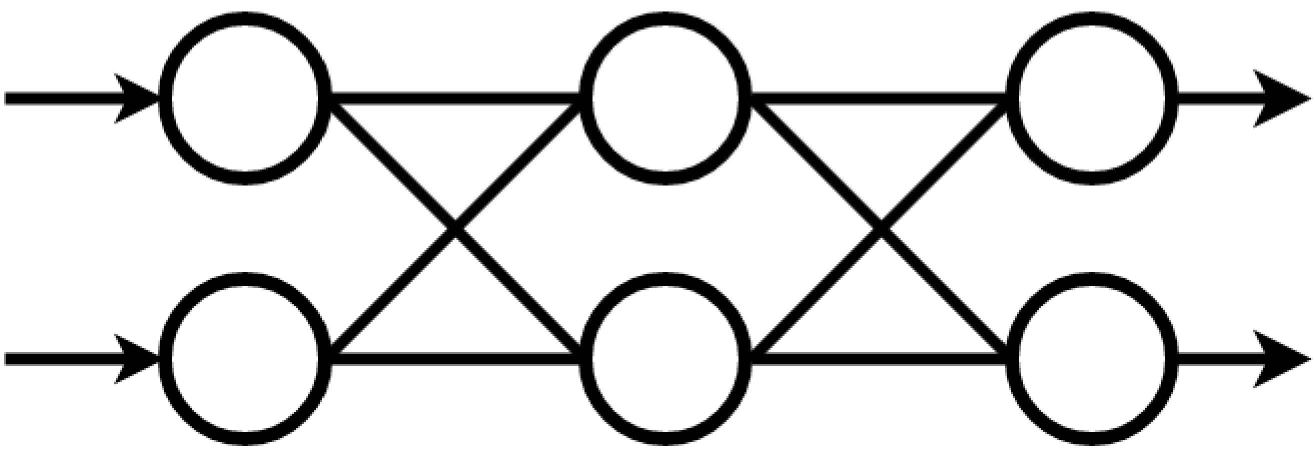
Activation Function

Output

$$f\left(\sum_{i=0}^{m}\left(w_i * x_i\right) + Bias\right)$$

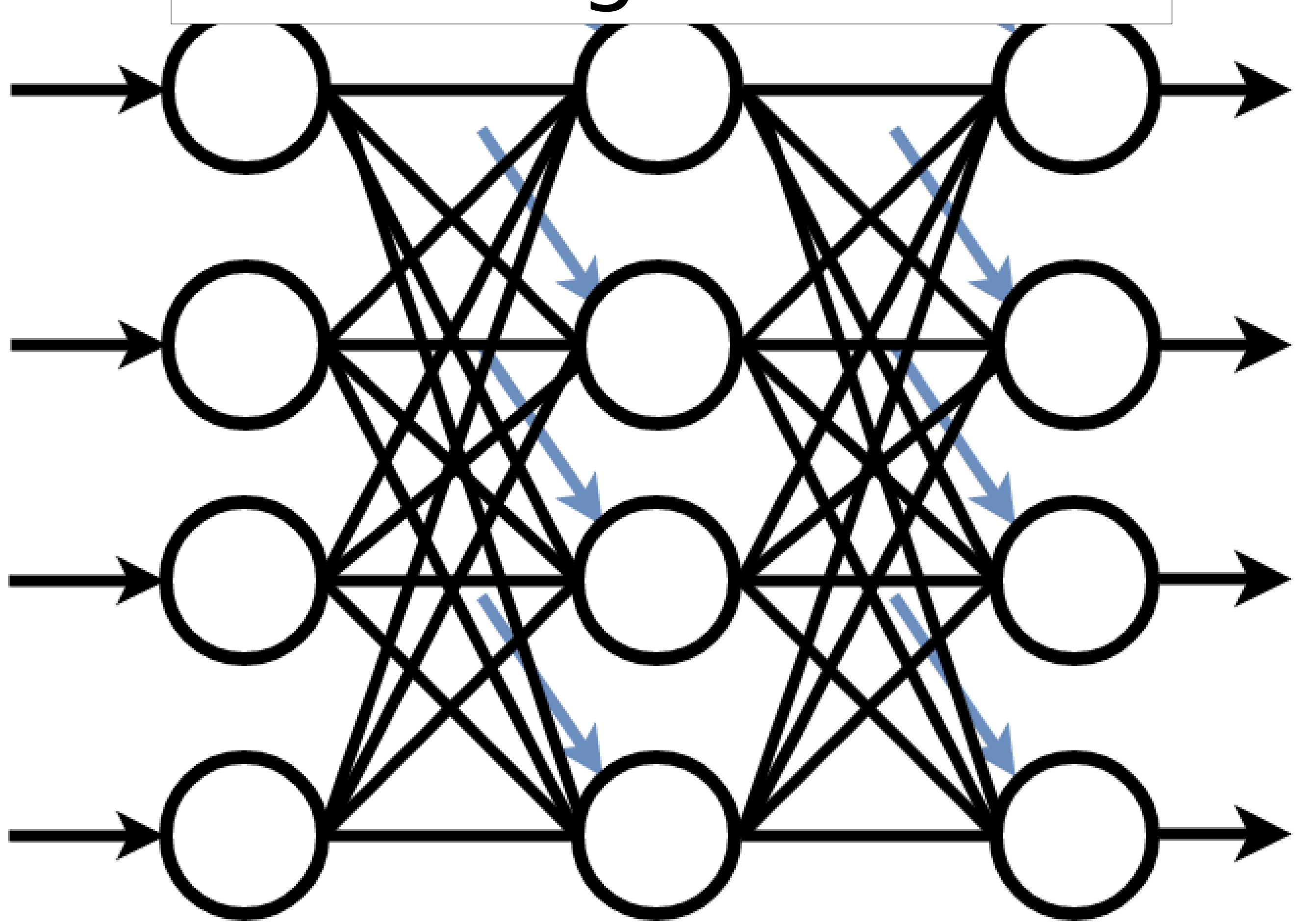
Multilayer Perceptrons

Where's the
Advantage?



Multilayer Perceptrons

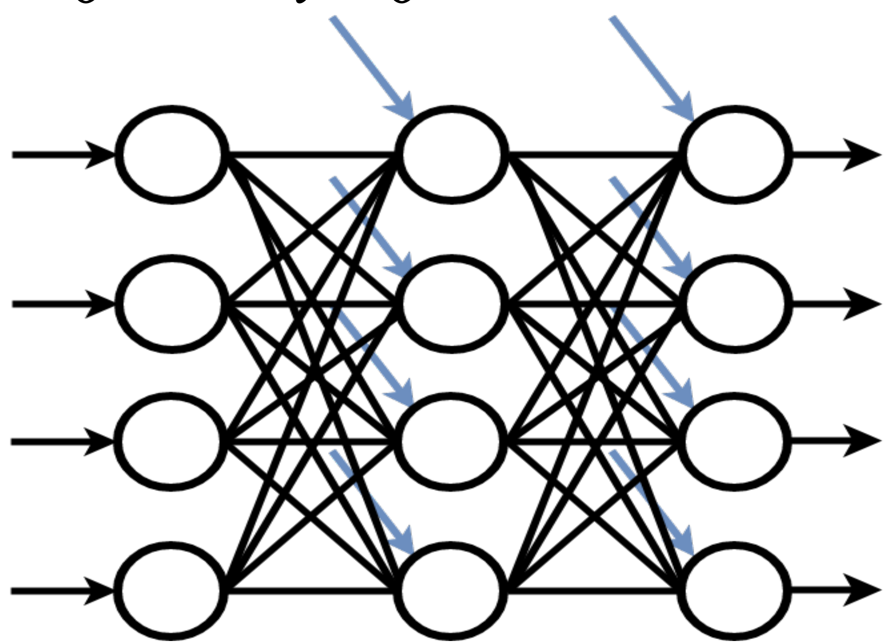
Where's the Advantage?



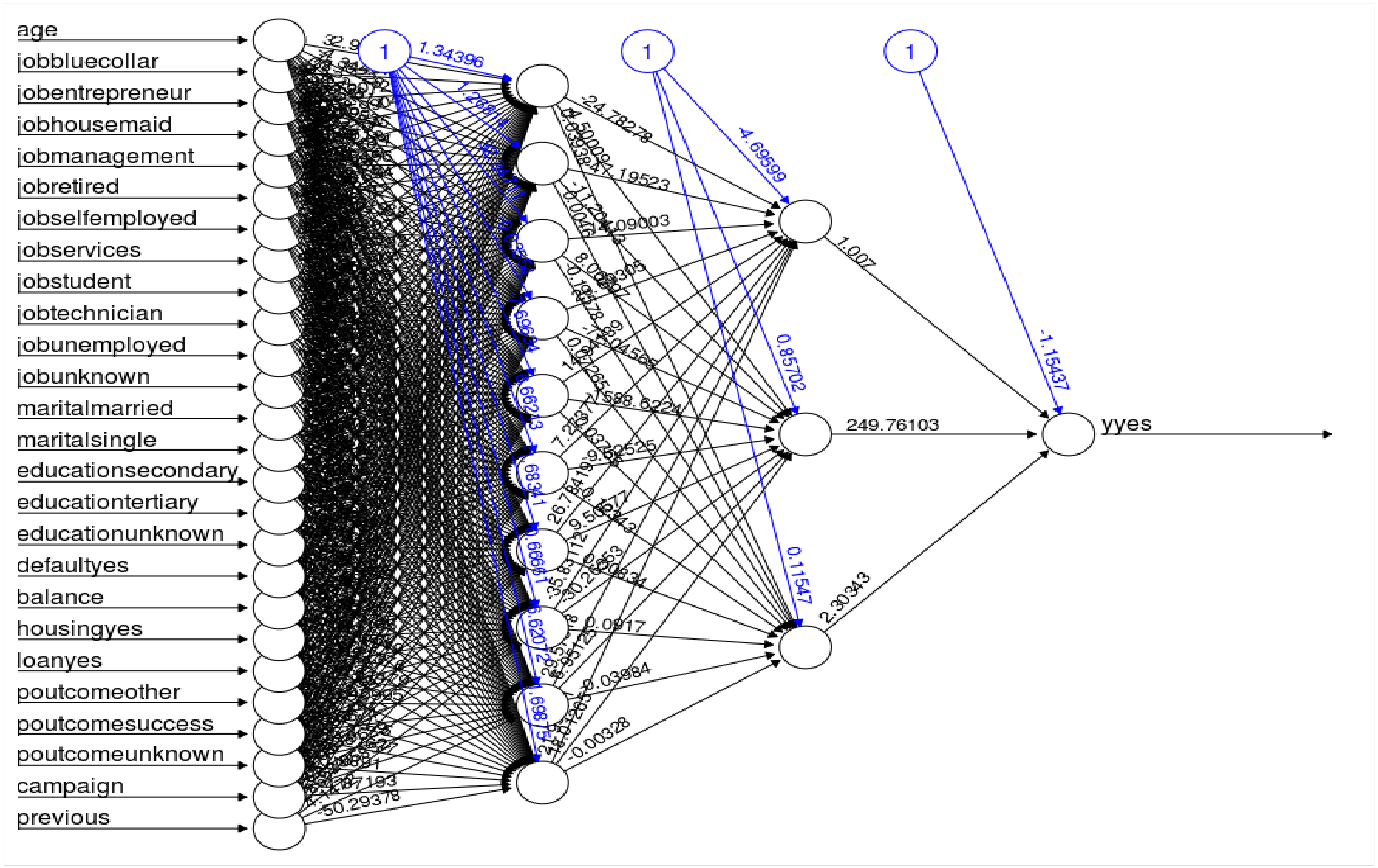
Multilayer Perceptrons

Where's the Advantage?

$$f\left(\sum_{j=0}^{m} f\left(\sum_{i=0}^{m} w_{i,j} * x_{i,j}\right) + Bias\right)$$



Multilayer Perceptrons

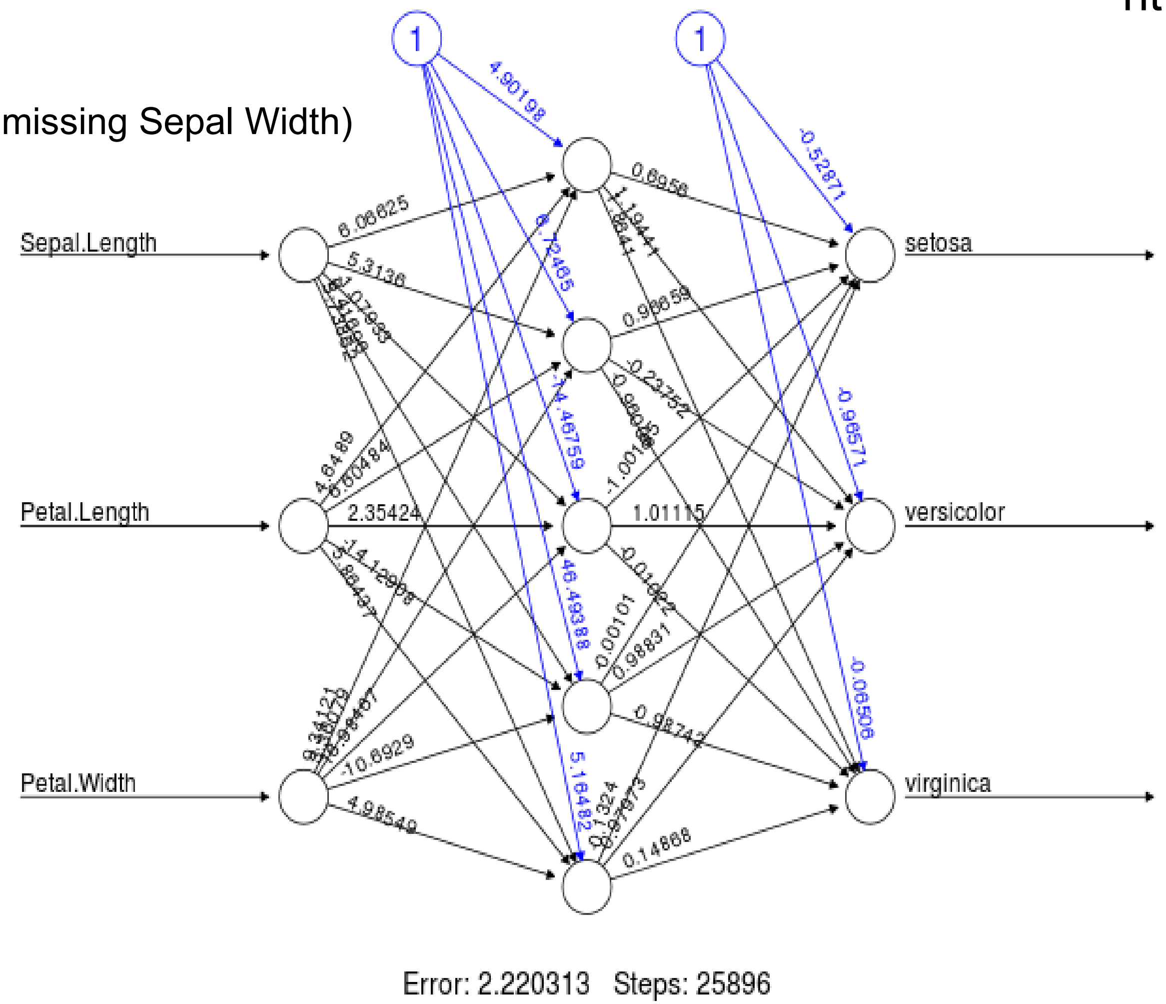


http://www.learnbymarketing.com/wp-content/uploads/2016/09/neural-net-r-multi-layer-tut.png

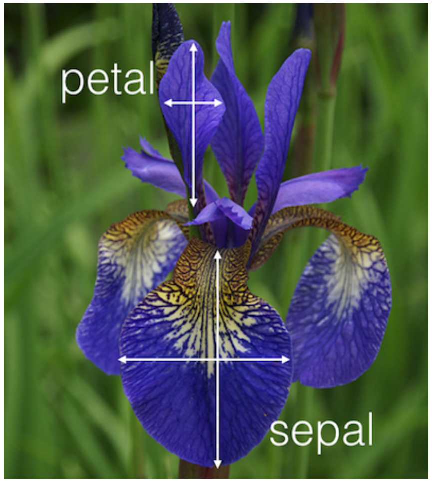
Multilayer Perceptrons

https://archive.ics.uci.edu/ml/datasets/Iris/

(missing Sepal Width)



Famous Iris dataset which measured the Different Length of Iris petals and classifies them as one of 3 Species

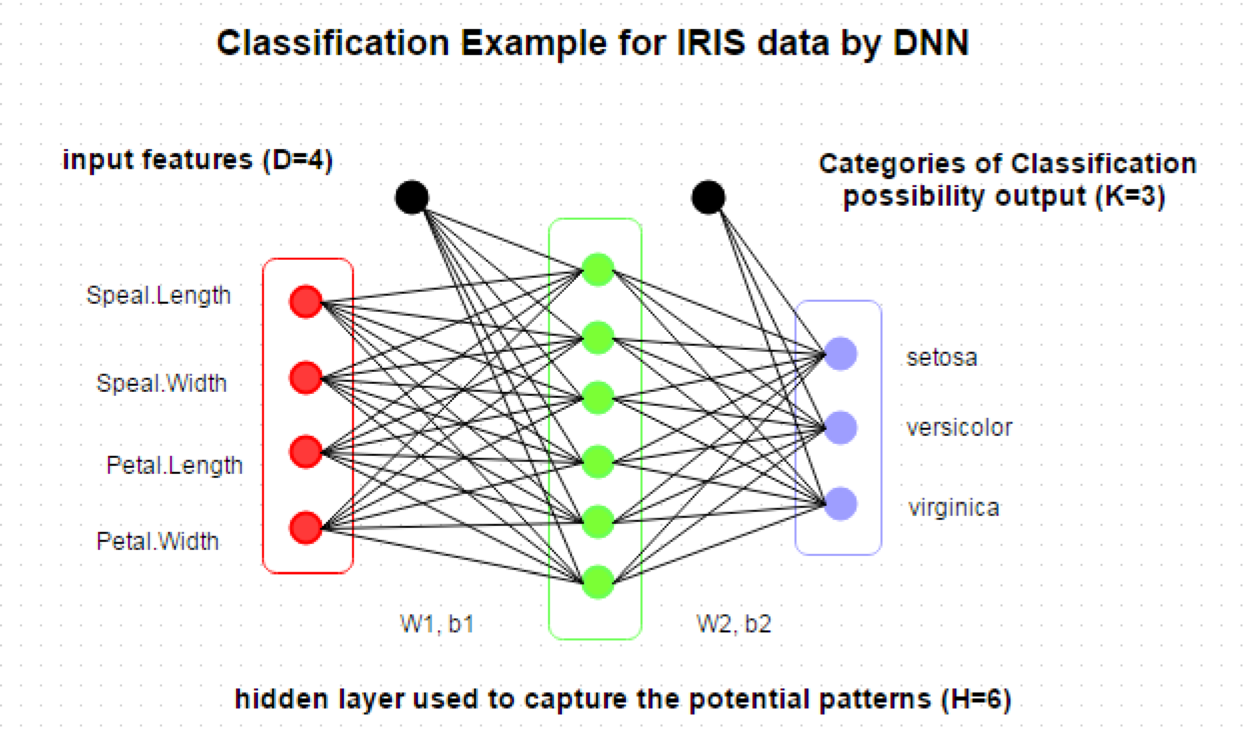


http://www.learnbymarketing.com/tutorials/neural-networks-in-r-tutorial/

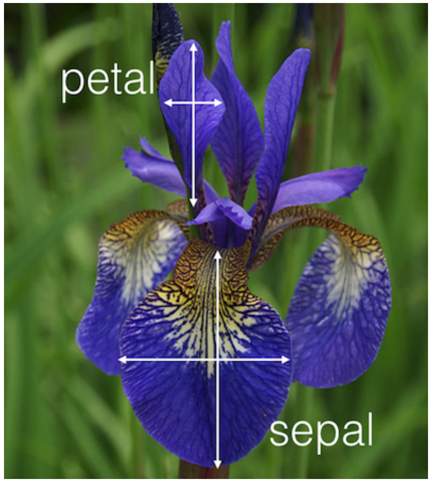
http://blog.kaggle.com/2015/04/22/scikit-learn-video-3-machine-learning-first-steps-with-the-iris-dataset/

Multilayer Perceptrons

https://archive.ics.uci.edu/ml/datasets/Iris/



Famous Iris dataset which measured the Different Length of Iris petals and classifies them as one of 3 Species



https://stats.stackexchange.com/questions/268202/backpropagation-algorithm-nn-with-rectified-linear-unit-relu-activation

http://blog.kaggle.com/2015/04/22/scikit-learn-video-3-machine-learning-first-steps-with-the-iris-dataset/

Machine Learning, An introduction 1: Conclusion

Machine Learning is the training of a model which learns based on data X to predict Y

It imitates the system which generated data X and Y, hopefully well...

See you next time for:

- Math!
- Optimization methods!
- More Math!!
- Eating habits of Neural Networks!
- Descending Gradients!

Written by Nicolas Symeou

for the Physical-Digital Affordances Group of the University of Regensburg