Artificial Neural Networks and Deep Learning

Week 1

Logistic regression and feed forward NN

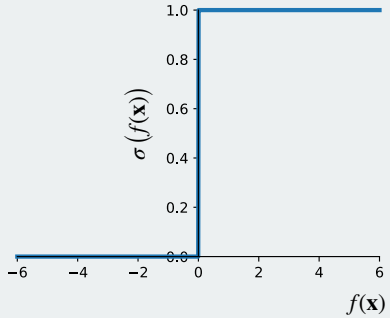
Linear regression

$$w_0 + x_0 w_1 + x_1 w_2 + x_2 w_3 = f(\mathbf{x})$$

Linear regression classifier

$$w_0 + x_0 w_1 + x_1 w_2 + x_2 w_3 = f(\mathbf{x})$$

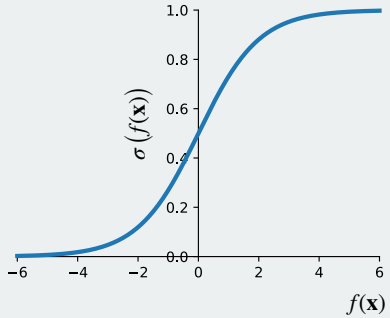
$$\sigma\left(f(\mathbf{x})\right) = \begin{cases} 1, & \textbf{if } f(\mathbf{x}) \geq 0 \\ 0, & \textbf{otherwise} \end{cases}$$



Linear regression classifier

$$w_0 + x_0 w_1 + x_1 w_2 + x_2 w_3 = f(\mathbf{x})$$

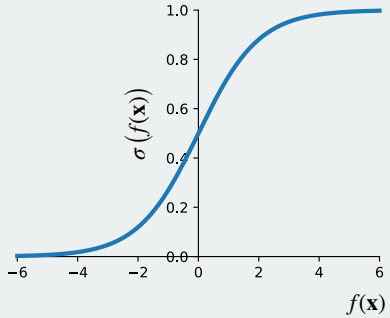
$$\sigma\left(f(\mathbf{x})\right) = \frac{1}{1 + exp\left(-f(x)\right)}$$



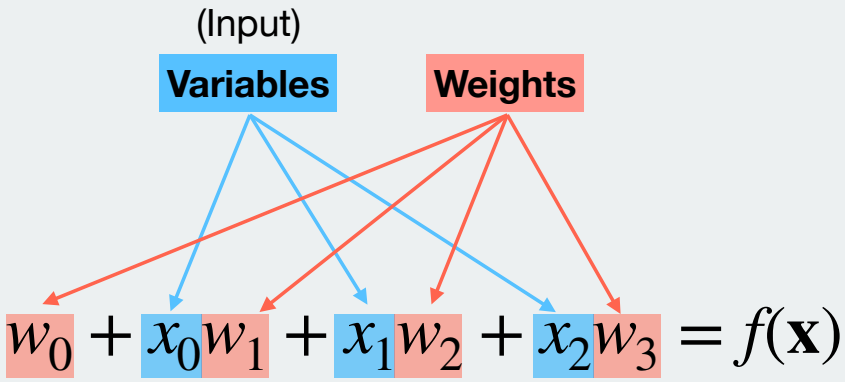
Linear regre ssifier

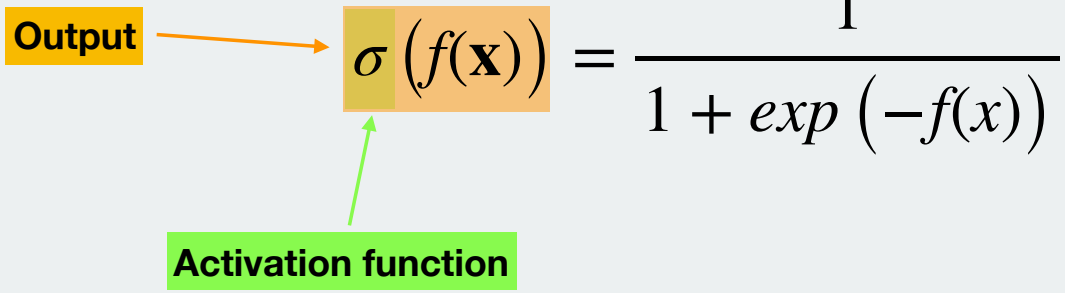
$$w_0 + x_0 w_1 + x_1 w_2 + x_2 w_3 = f(\mathbf{x})$$

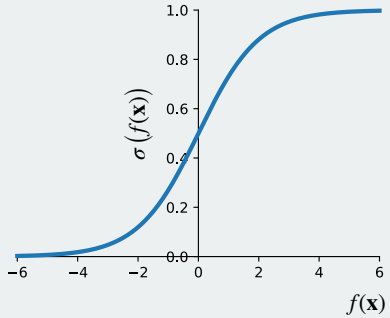
$$\sigma\left(f(\mathbf{x})\right) = \frac{1}{1 + exp\left(-f(x)\right)}$$



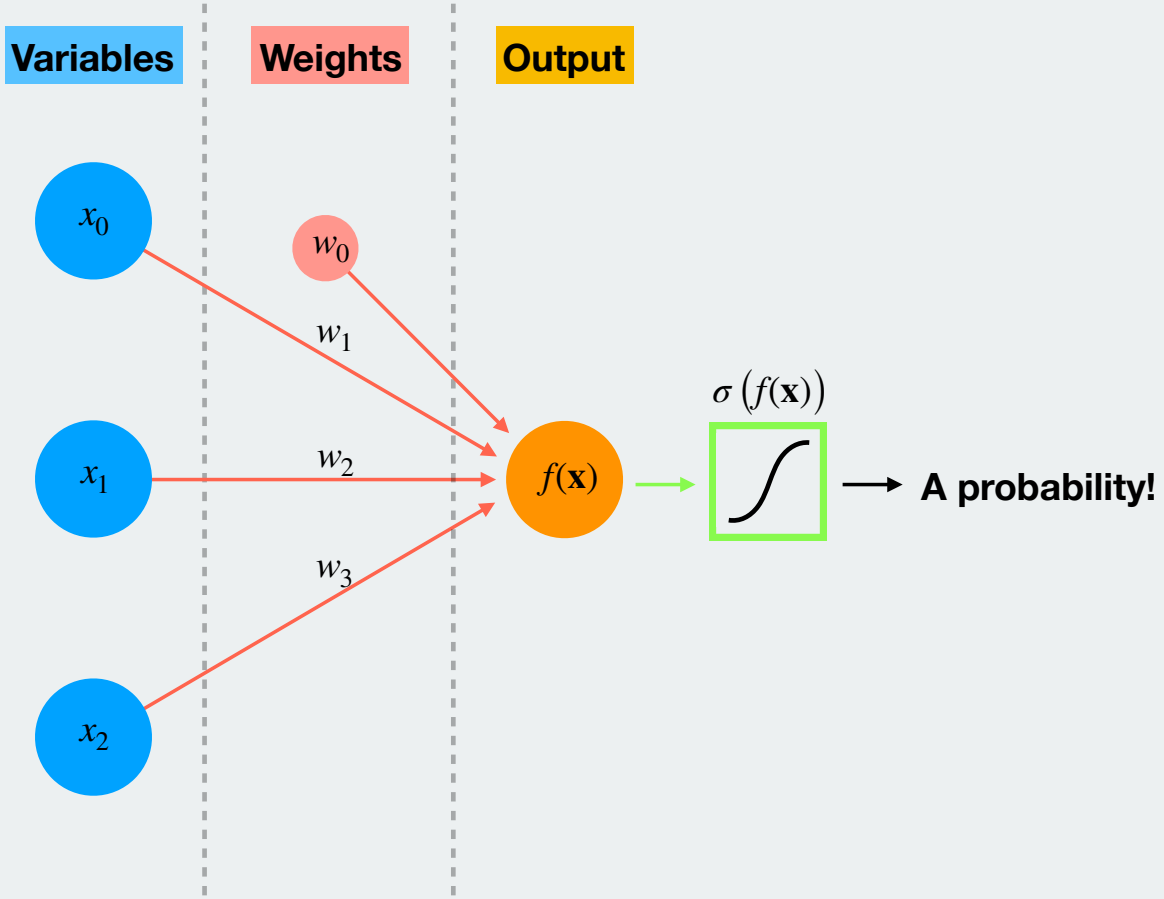
Logistic regression classifier

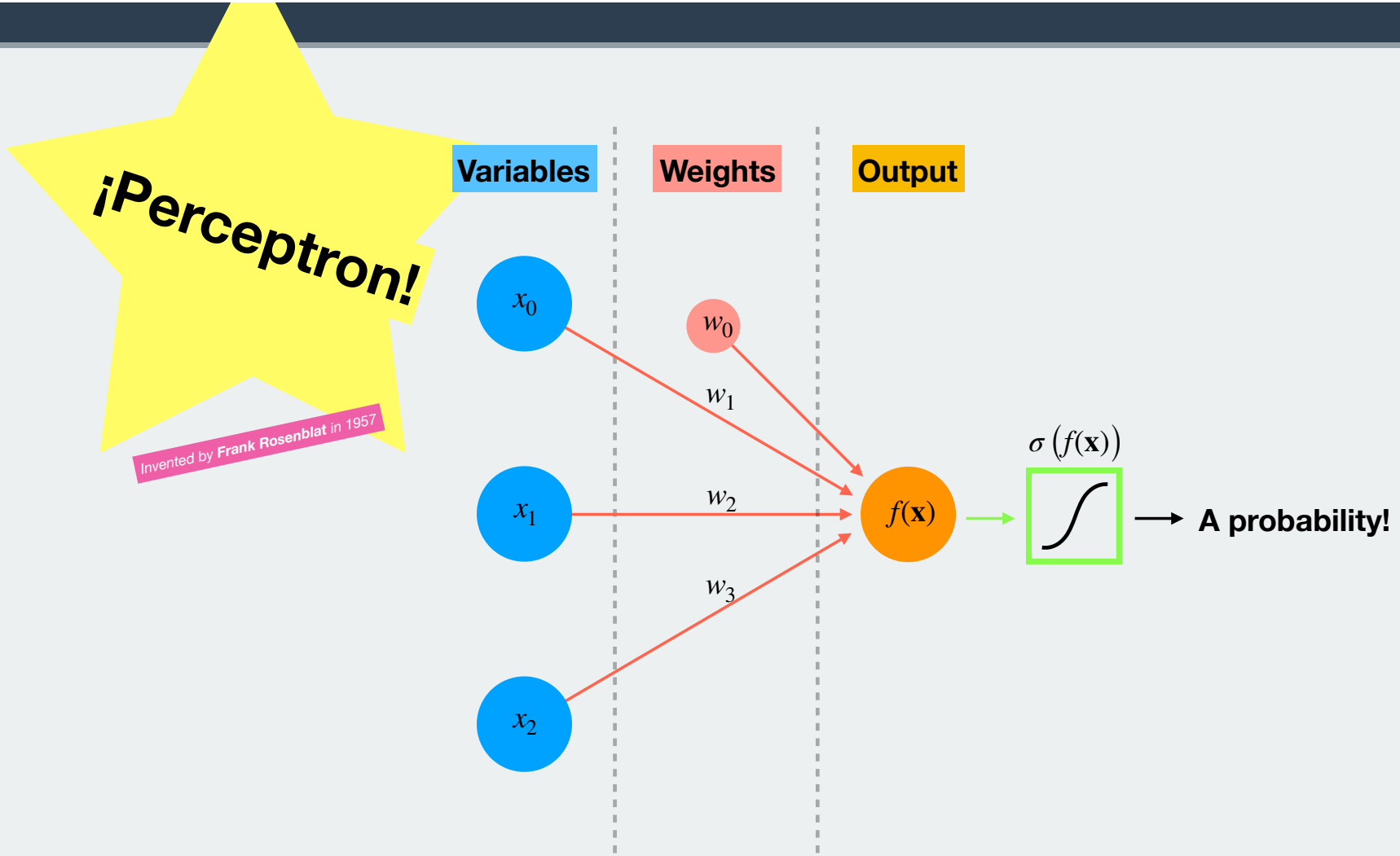






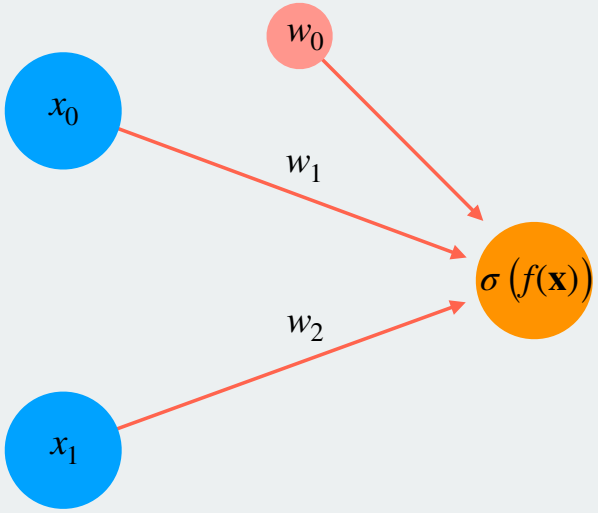
Logistic regression schematic illustration



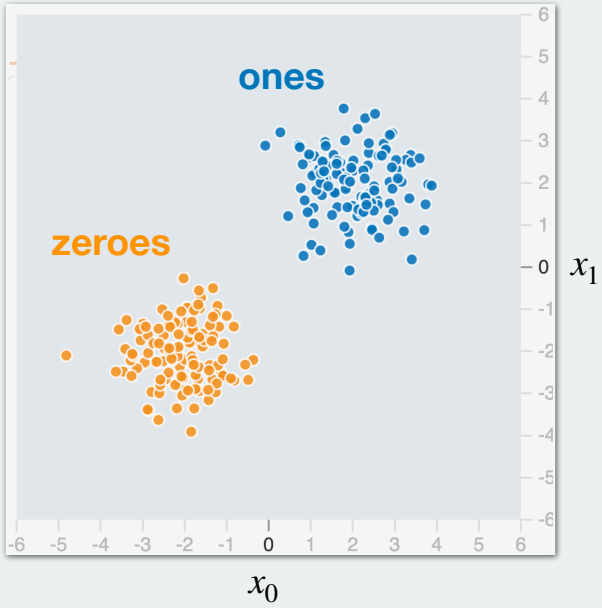

¡Perceptron!
Invented by Frank Rosenblat in 1957
Variables
Weights
Output
x_0
x_1
x_2
w_0
w_1
w_2
w_3
$f(\mathbf{x})$
$\sigma\left(f(\mathbf{x})\right)$
A probability!

Logistic regression simple problem

> Find values of $\{w_0, w_1, w_2\}$ that minimize $\sum_n (\tilde{y}_n - y_n)^2$



$$\mathbf{X} = \begin{array}{|c|c|} \hline x_0 & x_1 \\ \hline 1{,}2 & 2{,}5 \\ 2{,}1 & 3{,}4 \\ \ldots & \ldots \\ -3{,}3 & -2{,}0 \\ -1{,}9 & -2{,}5 \\ \hline \end{array} \quad , \quad \mathbf{y} = \begin{array}{|c|} \hline y \\ \hline 1 \\ 1 \\ \ldots \\ 0 \\ 0 \\ \hline \end{array}$$

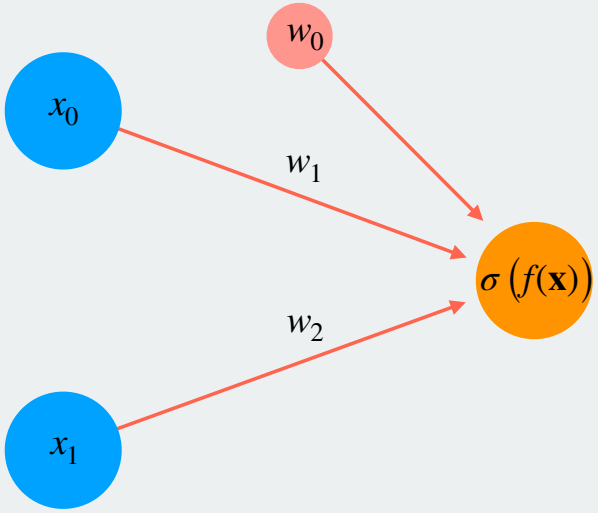


Colors are NOT related

Reminder: $w_0 + x_0 w_1 + x_1 w_2 = f(\mathbf{x})$

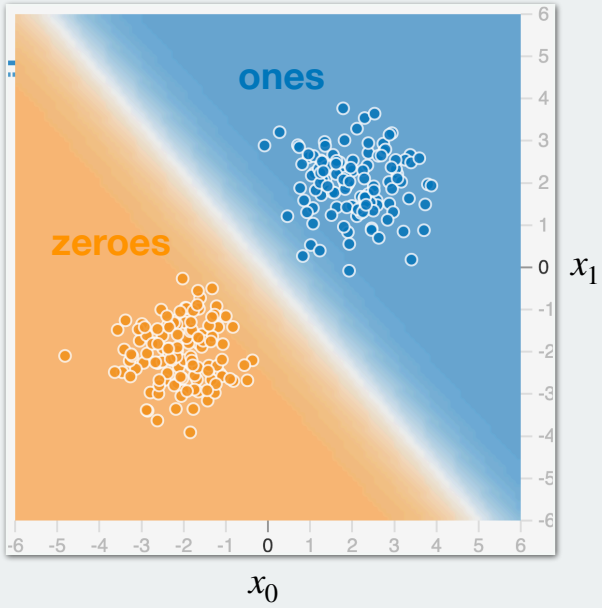
Logistic regression simple problem

> Find values of $\{w_0, w_1, w_2\}$ that minimizes $\sum_n (\tilde{y}_n - y_n)^2$



f has amazing weights gives **perfect prediction**

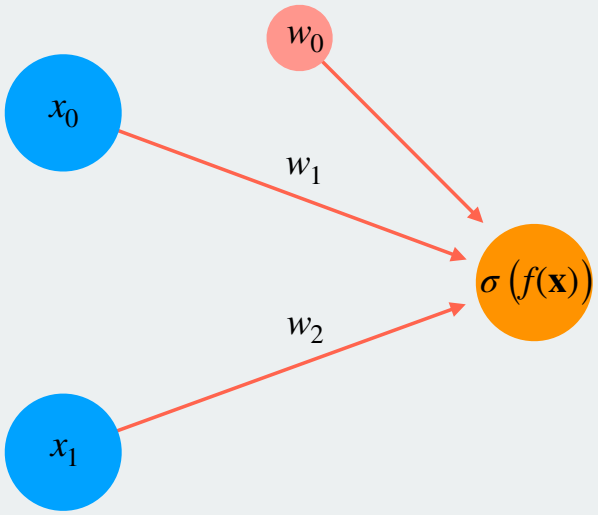
$$\mathbf{X} = \begin{array}{|c|c|}\hline x_0 & x_1 \\ \hline 1{,}2 & 2{,}5 \\ 2{,}1 & 3{,}4 \\ \ldots & \ldots \\ -3{,}3 & -2{,}0 \\ -1{,}9 & -2{,}5 \\ \hline \end{array}, \quad \sigma\left(f(\mathbf{X})\right) = \begin{array}{|c|}\hline \tilde{y} \\ \hline 1 \\ 1 \\ \ldots \\ 0 \\ 0 \\ \hline \end{array}, \quad \mathbf{y} = \begin{array}{|c|}\hline y \\ \hline 1 \\ 1 \\ \ldots \\ 0 \\ 0 \\ \hline \end{array}$$

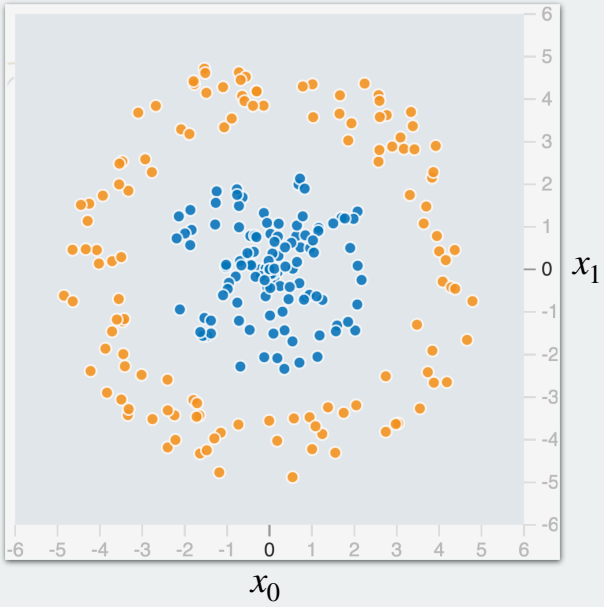


Reminder: $w_0 + x_0 w_1 + x_1 w_2 = f(\mathbf{x})$

Logistic regression not so simple problem

> Find values of $\{w_0, w_1, w_2\}$ that minimize $\sum_n (\tilde{y}_n - y_n)^2$



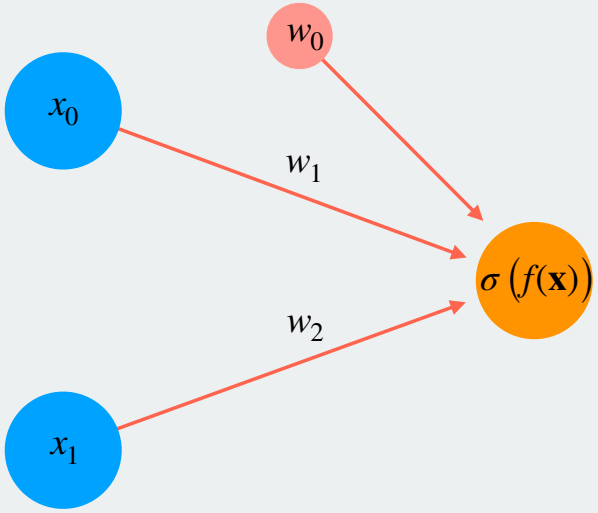


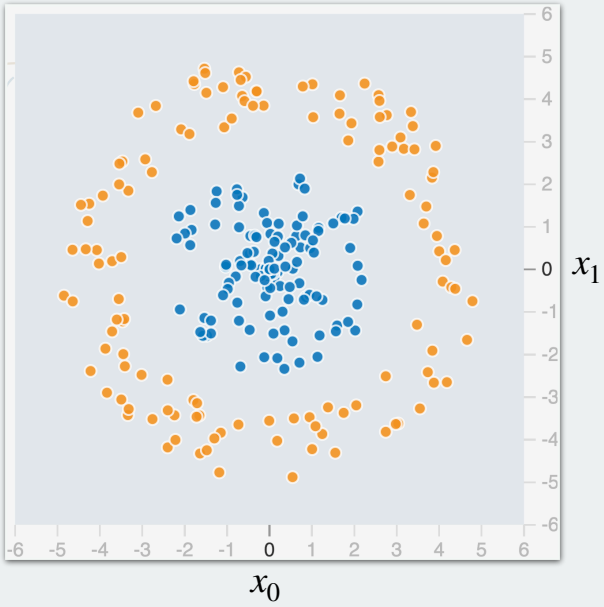
Reminder: $w_0 + x_0 w_1 + x_1 w_2 = f(\mathbf{x})$

Logistic regression not so simple problem

> Find values of $\{w_0, w_$ ...minimizes $\sum_n (\tilde{y}_n - y_n)^2$

Model too simple

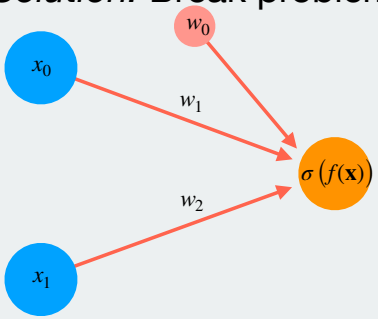




Reminder: $w_0 + x_0 w_1 + x_1 w_2 = f(\mathbf{x})$

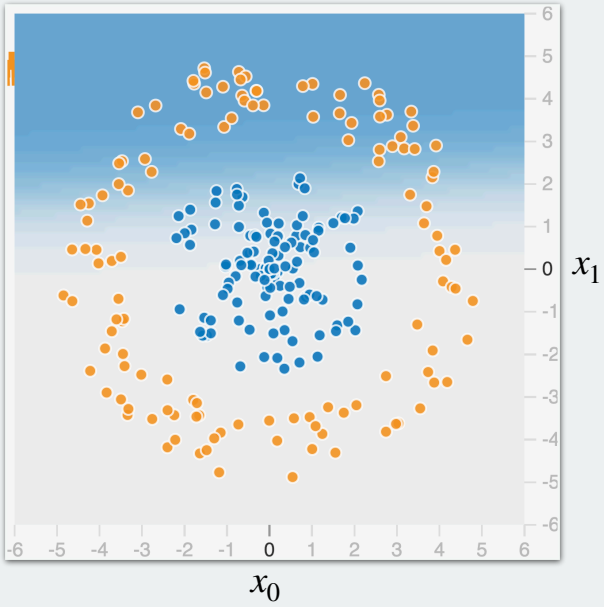
Logistic regression not so simple problem

> *Solution:* Break problem into subproblems



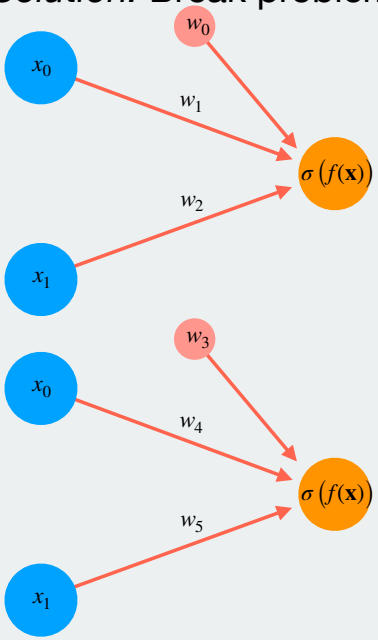
$$\sigma(w_0 + x_0 w_1 + x_1 w_2) = z_0(\mathbf{x})$$

$$\mathbf{X} = \begin{array}{|c|c|}\hline x_0 & x_1 \\ \hline 0{,}5 & 1{,}5 \\ 2{,}3 & -1{,}7 \\ \ldots & \ldots \\ 4{,}2 & -0{,}2 \\ -1{,}9 & 2{,}3 \\ \hline \end{array}, \quad z_0(\mathbf{X}) = \begin{array}{|c|}\hline z_0 \\ \hline 0{,}3 \\ 0{,}25 \\ \ldots \\ 0{,}79 \\ 0{,}34 \\ \hline \end{array}$$



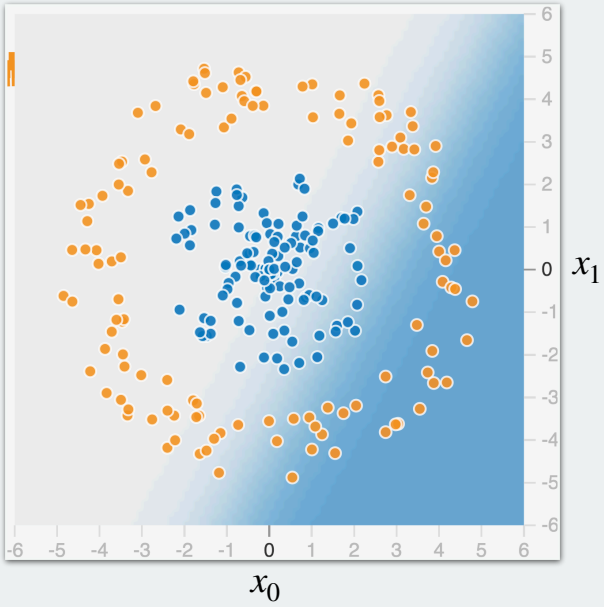
Logistic regression not so simple problem

> *Solution:* Break problem into subproblems



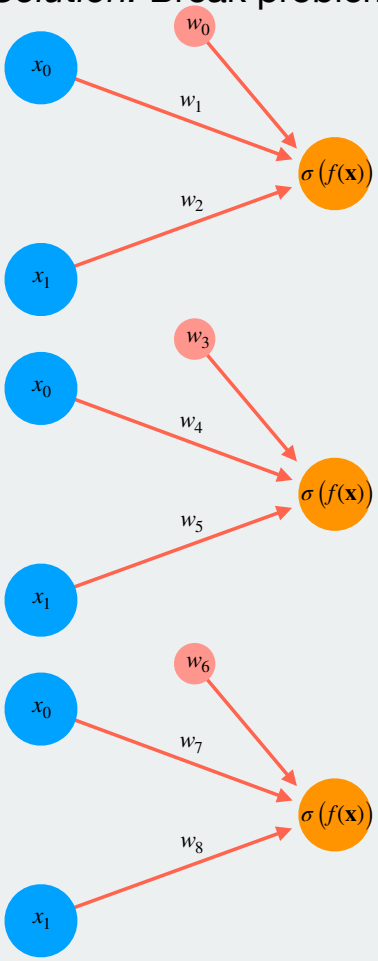
$$\sigma(w_0 + x_0 w_1 + x_1 w_2) = z_0(\mathbf{x})$$

$$\sigma(w_3 + x_0 w_4 + x_1 w_5) = z_1(\mathbf{x})$$



Logistic regression not so simple problem

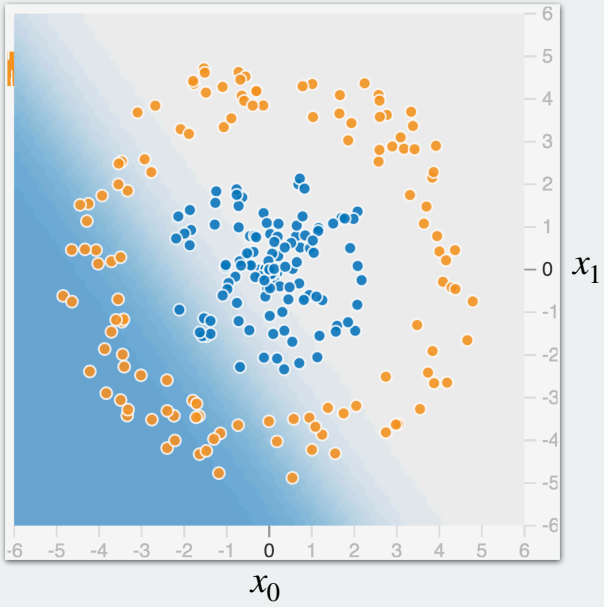
> *Solution:* Break problem into subproblems



$$\sigma(w_0 + x_0 w_1 + x_1 w_2) = z_0(\mathbf{x})$$

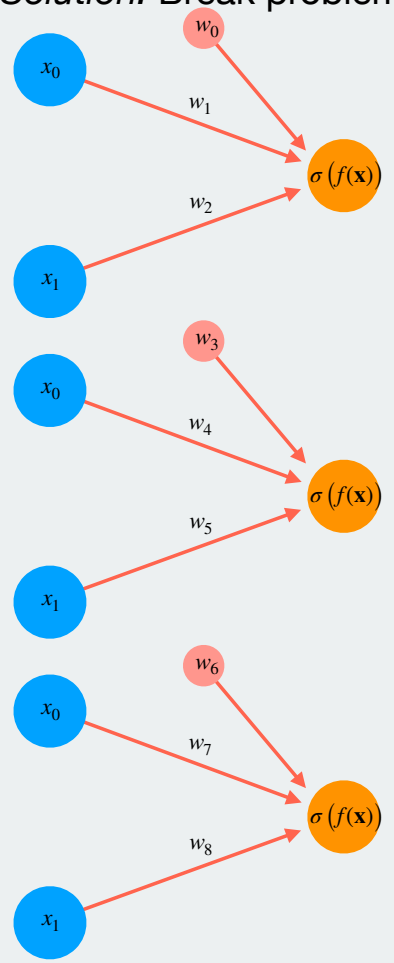
$$\sigma(w_3 + x_0 w_4 + x_1 w_5) = z_1(\mathbf{x})$$

$$\sigma(w_6 + x_0 w_7 + x_1 w_8) = z_2(\mathbf{x})$$



Logistic regression not so simple problem

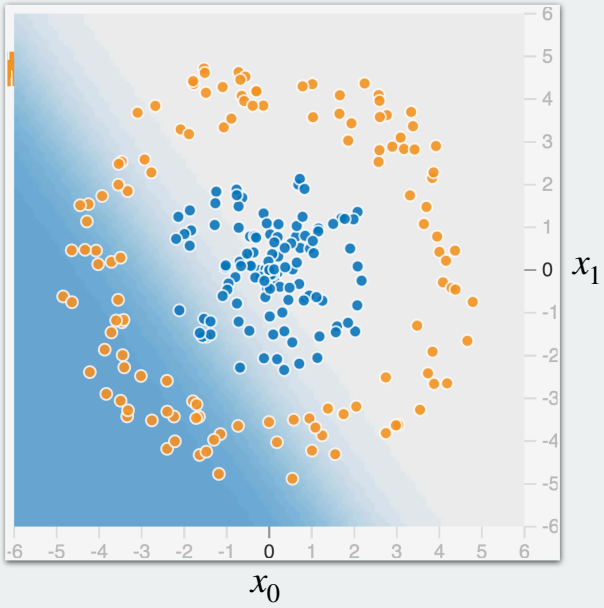
> *Solution:* Break problem into subproblems



$$\sigma(w_0 + x_0 w_1 + x_1 w_2) = z_0(\mathbf{x})$$

$$\sigma(w_3 + x_0 w_4 + x_1 w_5) = z_1(\mathbf{x})$$

$$\sigma(w_6 + x_0 w_7 + x_1 w_8) = z_2(\mathbf{x})$$

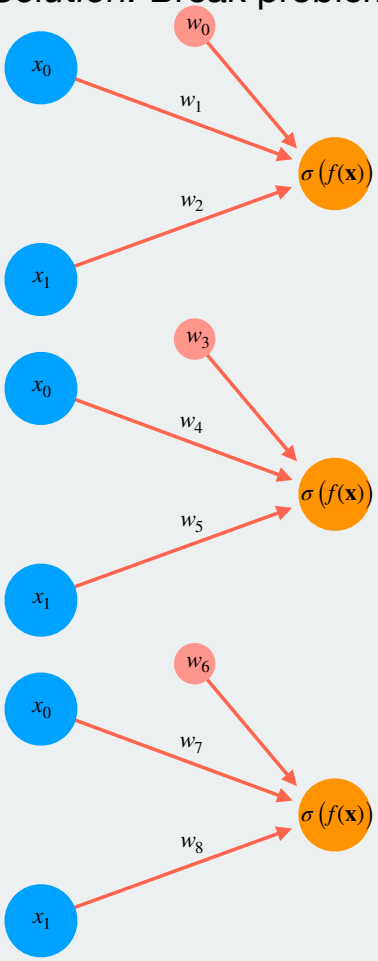


New problem: Given **Z**, predict **y**

$$\mathbf{Z} = \begin{array}{|c|c|c|} \hline z_0 & z_1 & z_2 \\ \hline 0{,}3 & 0{,}75 & 0{,}78 \\ 0{,}25 & 0{,}1 & 0{,}95 \\ \ldots & \ldots & \\ 0{,}79 & 0{,}99 & 0{,}3 \\ 0{,}34 & 0{,}6 & 0{,}1 \\ \hline \end{array} \qquad \mathbf{y} = \begin{array}{|c|} \hline y \\ \hline 0 \\ 1 \\ \ldots \\ 1 \\ 0 \\ \hline \end{array}$$

Logistic regression not so simple problem

> *Solution:* Break problem into subproblems



$$\sigma(w_0 + x_0 w_1 + x_1 w_2) = z_0(\mathbf{x})$$

$$\sigma(w_3 + x_0 w_4 + x_1 w_5) = z_1(\mathbf{x})$$

$$\sigma(w_6 + x_0 w_7 + x_1 w_8) = z_2(\mathbf{x})$$

New problem: Given **Z**, predict **y**

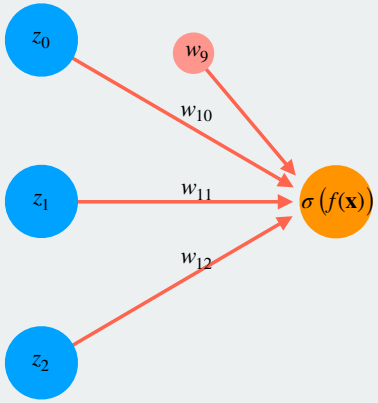
$\mathbf{Z} =$

z_0	z_1	z_2
0,3	0,75	0,78
0,25	0,1	0,95
...	...	
0,79	0,99	0,3
0,34	0,6	0,1

$\mathbf{y} =$

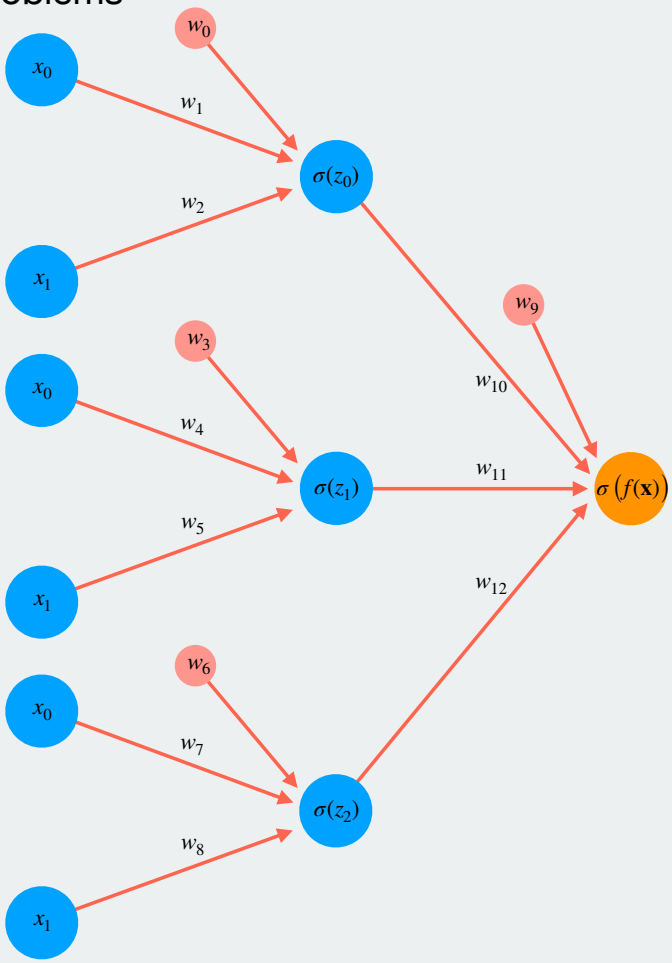
y
0
1
...
1
0

Solution: Why not use a logistic regression?



Logistic regression not so simple problem

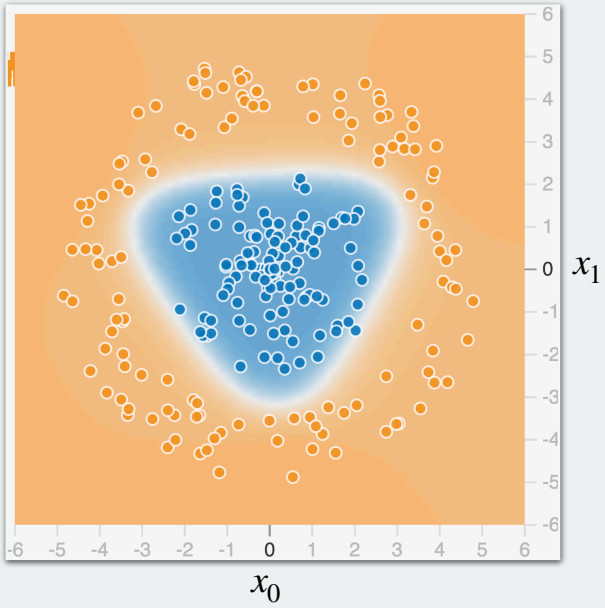
> *Solution:* Connect subproblems



Mathematical form

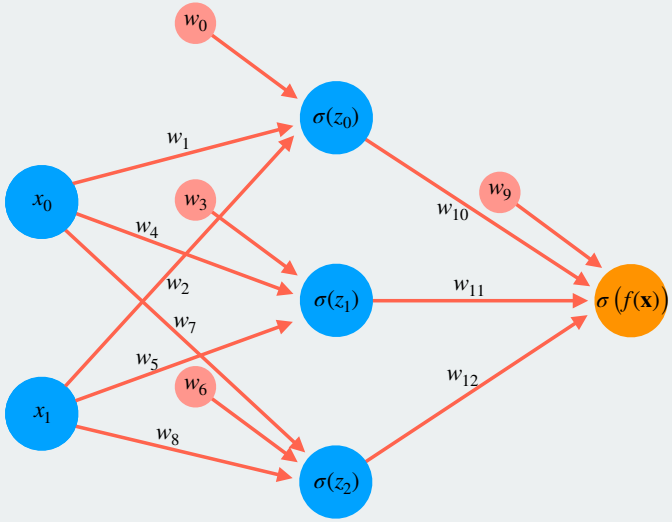
$$\sigma\Big(w_9 + \sigma(w_0 + x_0 w_1 + x_1 w_2) w_{10} \quad + \\ \sigma(w_3 + x_0 w_4 + x_1 w_5) w_{11} \quad + \\ \sigma(w_6 + x_0 w_7 + x_1 w_8) w_{12}\Big) = \sigma\big(f(x)\big)$$

Solution



Logistic regression not so simple problem

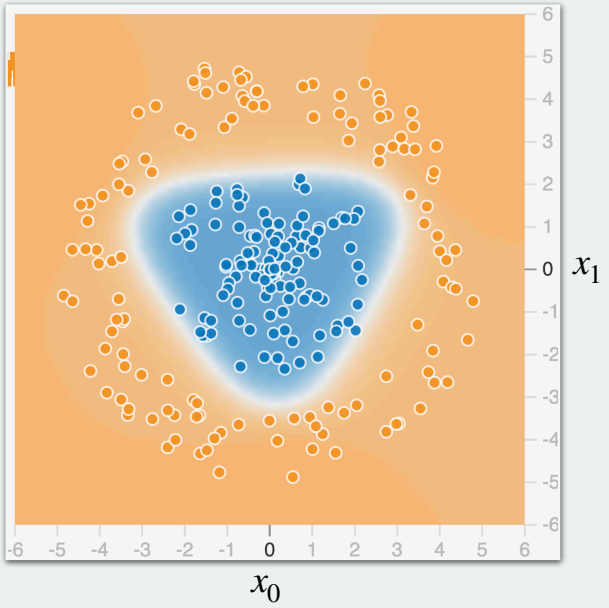
> *Solution:* Connect subproblems



Mathematical form

$$\sigma\Big(w_9 + \sigma(w_0 + x_0 w_1 + x_1 w_2) w_{10} \quad + \\ \sigma(w_3 + x_0 w_4 + x_1 w_5) w_{11} \quad + \\ \sigma(w_6 + x_0 w_7 + x_1 w_8) w_{12}\Big) = \sigma\big(f(x)\big)$$

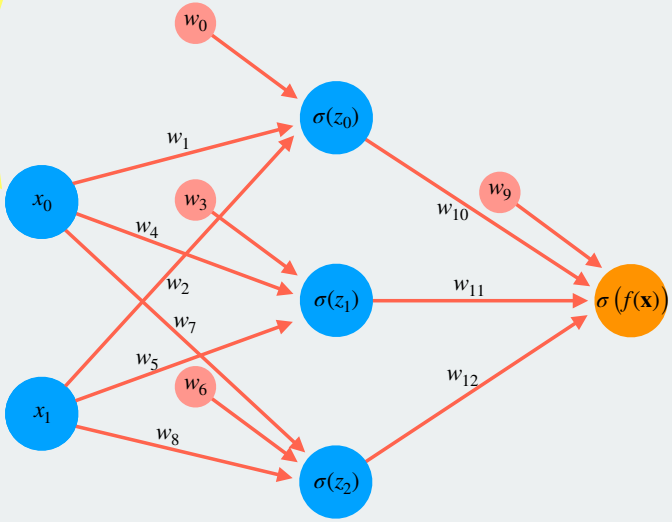
Solution



Logistic regression ot so simple problem

¡Multilayer perceptron!

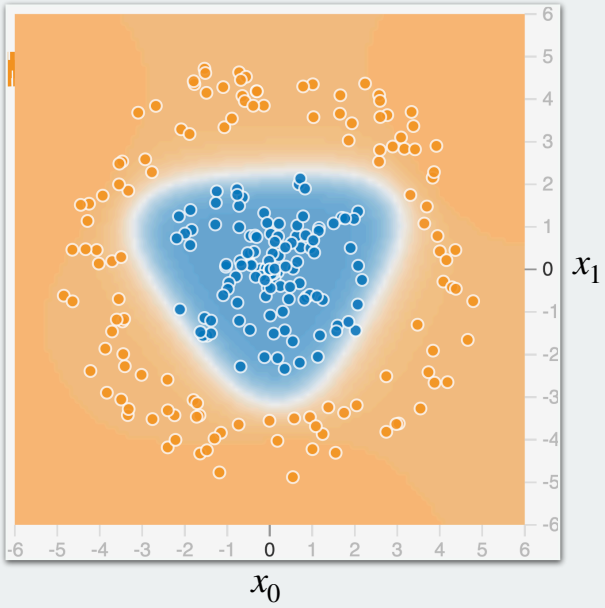
classifier



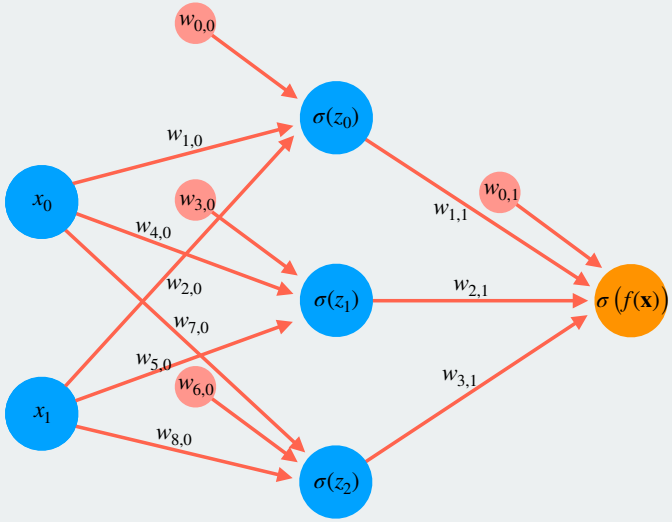
Mathematical form

$$\sigma\Big(w_9 + \sigma(w_0 + x_0 w_1 + x_1 w_2) w_{10} \quad + \\ \sigma(w_3 + x_0 w_4 + x_1 w_5) w_{11} \quad + \\ \sigma(w_6 + x_0 w_7 + x_1 w_8) w_{12}\Big) = \sigma\big(f(x)\big)$$

Solution



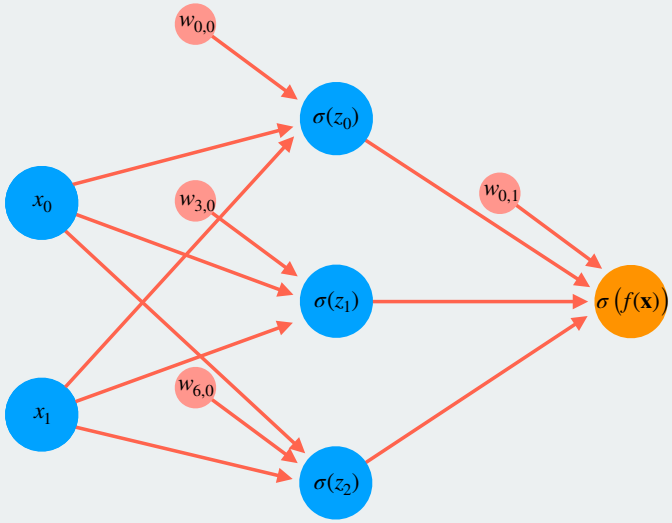
Multilayer perceptron – the math



Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0}) w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0}) w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0}) w_{3,1}\Big) = \sigma\Big(f(x)\Big)$$

Multilayer perceptron – the math

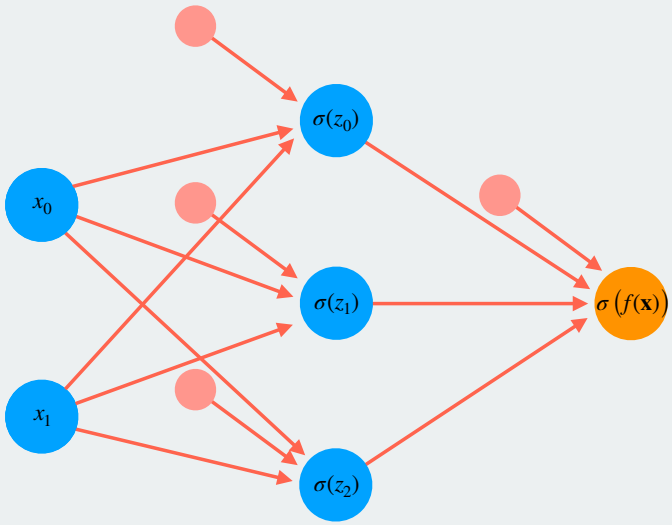


Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0}) w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0}) w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0}) w_{3,1}\Big) = \sigma\Big(f(x)\Big)$$

$$\mathbf{W_0} = \begin{bmatrix} w_{1,0} & w_{2,0} \\ w_{4,0} & w_{5,0} \\ w_{7,0} & w_{8,0} \end{bmatrix} \qquad \mathbf{W_1} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \end{bmatrix}$$

Multilayer perceptron – the math

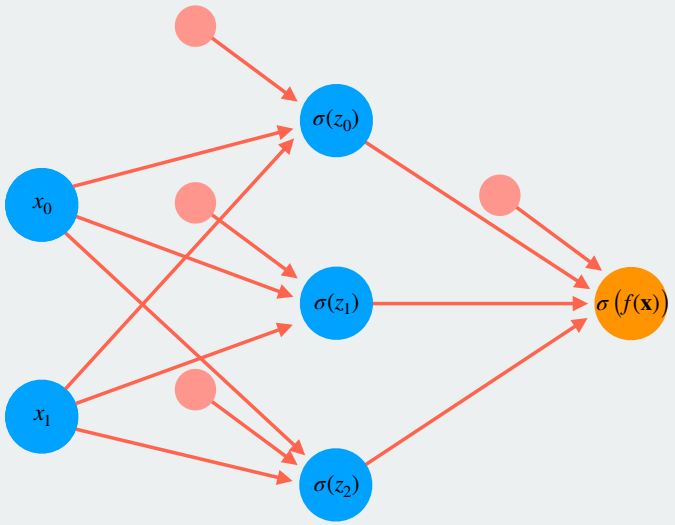


Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0}) w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0}) w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0}) w_{3,1}\Big) = \sigma\Big(f(x)\Big)$$

$$\mathbf{b_0} = \begin{bmatrix} w_{0,0} \\ w_{3,0} \\ w_{6,0} \end{bmatrix} \quad \mathbf{W_0} = \begin{bmatrix} w_{1,0} & w_{2,0} \\ w_{4,0} & w_{5,0} \\ w_{7,0} & w_{8,0} \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} w_{0,1} \end{bmatrix} \quad \mathbf{W_1} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \end{bmatrix}$$

Multilayer perceptron – the math



Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0}) w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0}) w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0}) w_{3,1}\Big) = \sigma\big(f(x)\big)$$

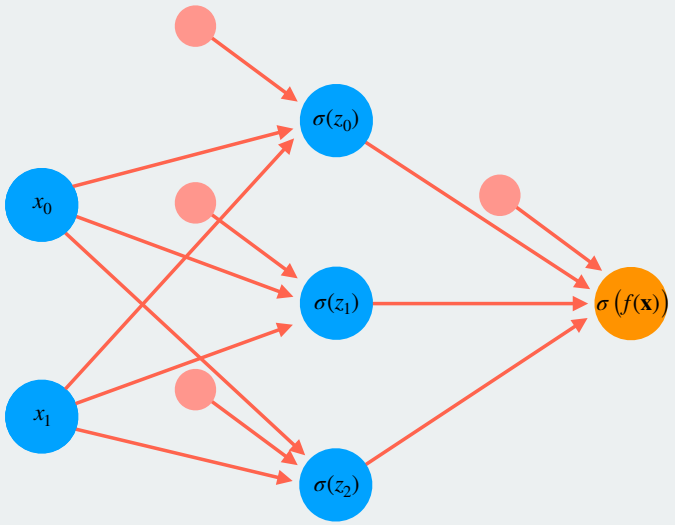
$$\Leftrightarrow$$

$$\sigma\Big(\mathbf{b_1} + \mathbf{W_1}\sigma\big(\mathbf{b_0} + \mathbf{W_0}\mathbf{x}\big)\Big) = \sigma\big(f(x)\big)$$

$$\mathbf{b_0} = \begin{bmatrix} w_{0,0} \\ w_{3,0} \\ w_{6,0} \end{bmatrix} \quad \mathbf{W_0} = \begin{bmatrix} w_{1,0} & w_{2,0} \\ w_{4,0} & w_{5,0} \\ w_{7,0} & w_{8,0} \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} w_{0,1} \end{bmatrix} \quad \mathbf{W_1} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \end{bmatrix}$$

and $\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$

Multilayer perceptron – the math



$$\mathbf{b_0} = \begin{bmatrix} w_{0,0} \\ w_{3,0} \\ w_{6,0} \end{bmatrix} \quad \mathbf{W_0} = \begin{bmatrix} w_{1,0} & w_{2,0} \\ w_{4,0} & w_{5,0} \\ w_{7,0} & w_{8,0} \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} w_{0,1} \end{bmatrix} \quad \mathbf{W_1} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \end{bmatrix}$$

Mathematical form

$$\begin{aligned} \sigma\big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0}) w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0}) w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0}) w_{3,1}\big) = \sigma\big(f(x)\big) \end{aligned}$$

$$\Leftrightarrow$$

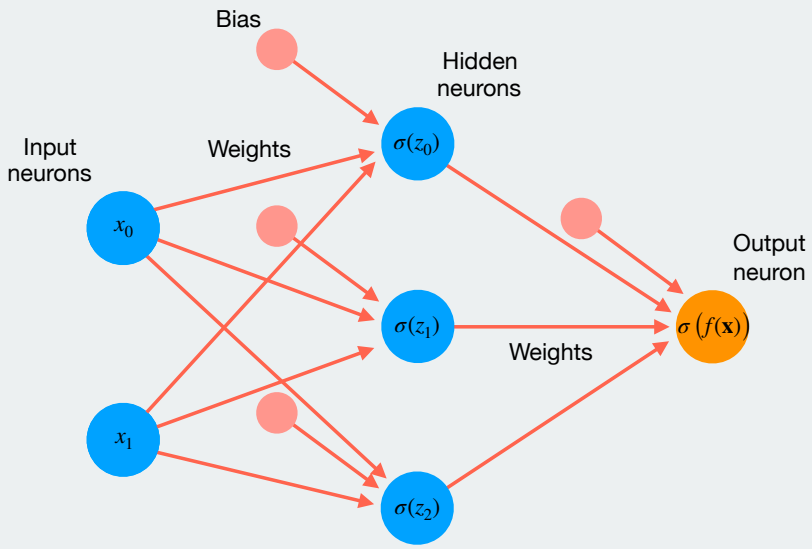
$$\sigma\Big(\mathbf{b_1} + \mathbf{W_1}\sigma\big(\mathbf{b_0} + \mathbf{W_0}\mathbf{x}\big)\Big) = \sigma\big(f(x)\big)$$

$$\Leftrightarrow$$

$$\sigma\big(\mathbf{b_n} + \mathbf{W_n}\mathbf{a_{n-1}}\big) = \mathbf{a_n}$$

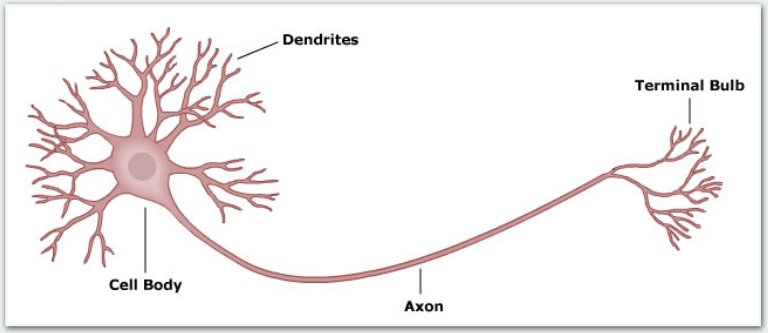
and $\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$

Multilayer perceptron – the structure



Multilayer perceptron – why “Neural”?

Brain neuron

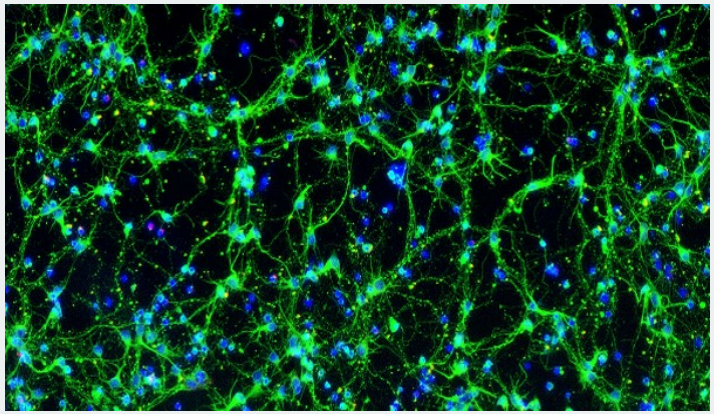


https://neurofantastic.com/brain/2017/4/13/brain-computation-is-a-lot-more-analog-than-we-thought

3D artist impression

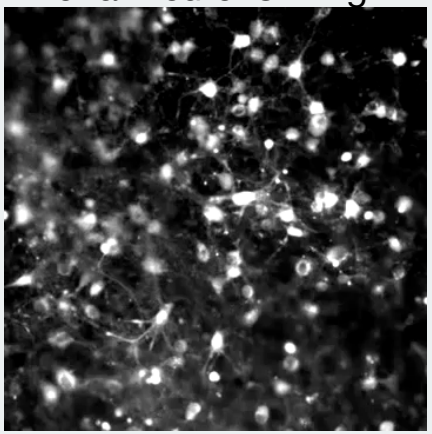
https://alsnewstoday.com/2017/07/25/als-study-shows-how-excessive-dna-repetitions-trigger-neuron-deaths/

Rat neuron image



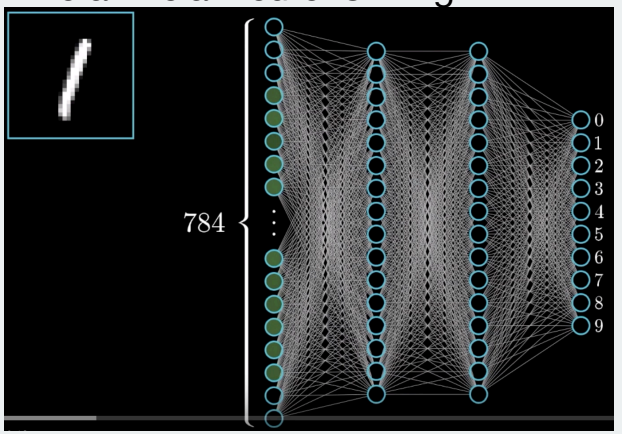
https://www.cellapplications.com/expanded-neuron-offering

Live rat neurons firing



https://www.youtube.com/watch?v=yy994HpFudc

Live artificial neurons firing



https://www.youtube.com/watch?v=aircAruvnKk&t=1s

Multilayer perceptron – why use non-linearities?

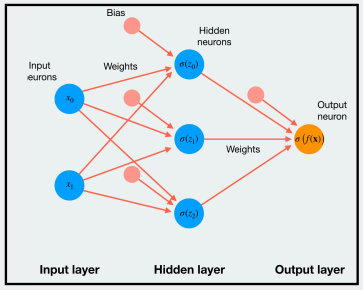
Mathematical form

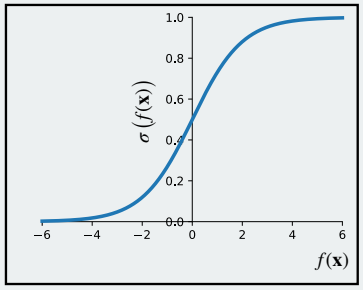
$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1}\Big) = \sigma\Big(f(x)\Big)$$

Mathematical form without non-linearities

$$w_{0,1} + (w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} \quad + \\ (w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} \quad + \\ (w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1} = f(x)$$

Reminders





Multilayer perceptron – why use non-linearities?

Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1}\Big) = \sigma\Big(f(x)\Big)$$

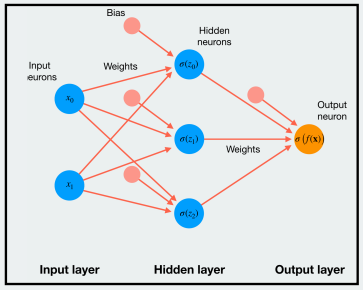
Mathematical form without non-linearities

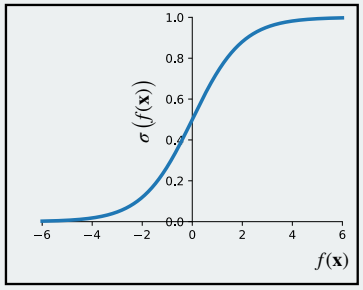
$$w_{0,1} + (w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} + \\ (w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} + \\ (w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1} = f(x)$$

$$\Leftrightarrow$$

$$x_0(w_{1,0}w_{1,1} + w_{4,0}w_{2,1} + w_{7,0}w_{3,1}) + \\ x_1(w_{2,0}w_{1,1} + w_{5,0}w_{2,1} + w_{8,0}w_{3,1}) + \\ w_{0,1} + w_{0,0}w_{1,1} + w_{3,0}w_{2,1} + w_{6,0}w_{3,1} = f(x)$$

Reminders





Multilayer perceptron – why use non-linearities?

Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1}\Big) = \sigma\big(f(x)\big)$$

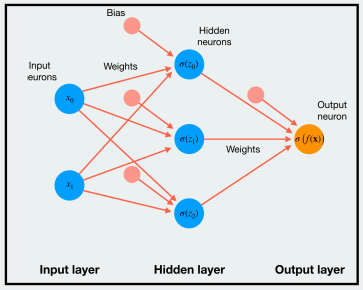
Mathematical form without non-linearities

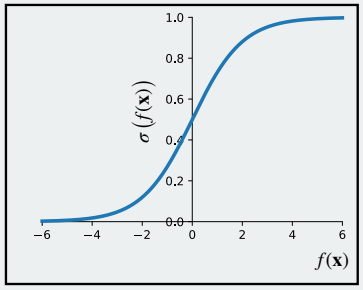
$$w_{0,1} + (w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} + \\ (w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} + \\ (w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1} = f(x)$$

$$\Leftrightarrow$$

$$x_0(w_{1,0}w_{1,1} + w_{4,0}w_{2,1} + w_{7,0}w_{3,1}) + \\ x_1(w_{2,0}w_{1,1} + w_{5,0}w_{2,1} + w_{8,0}w_{3,1}) + \\ w_{0,1} + w_{0,0}w_{1,1} + w_{3,0}w_{2,1} + w_{6,0}w_{3,1} = f(x)$$

Reminders





Multilayer perceptron – why use non-linearities?

Mathematical form

$$\sigma\Big(w_{0,1} + \sigma(w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} \quad + \\ \sigma(w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} \quad + \\ \sigma(w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1}\Big) = \sigma\big(f(x)\big)$$

Mathematical form without non-linearities

$$w_{0,1} + (w_{0,0} + x_0 w_{1,0} + x_1 w_{2,0})w_{1,1} + \\ (w_{3,0} + x_0 w_{4,0} + x_1 w_{5,0})w_{2,1} + \\ (w_{6,0} + x_0 w_{7,0} + x_1 w_{8,0})w_{3,1} = f(x)$$

$$\Leftrightarrow$$

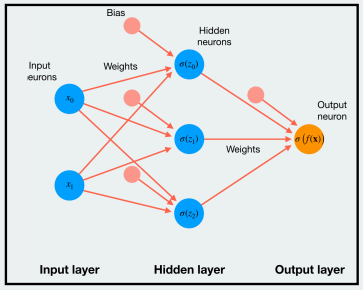
$$x_0(w_{1,0}w_{1,1} + w_{4,0}w_{2,1} + w_{7,0}w_{3,1}) + \\ x_1(w_{2,0}w_{1,1} + w_{5,0}w_{2,1} + w_{8,0}w_{3,1}) + \\ w_{0,1} + w_{0,0}w_{1,1} + w_{3,0}w_{2,1} + w_{6,0}w_{3,1} = f(x)$$

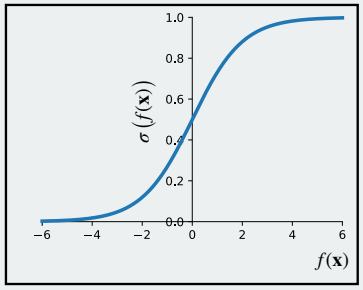
$$\Leftrightarrow$$

$$x_0 w_a + x_1 w_b + w_c = f(x)$$

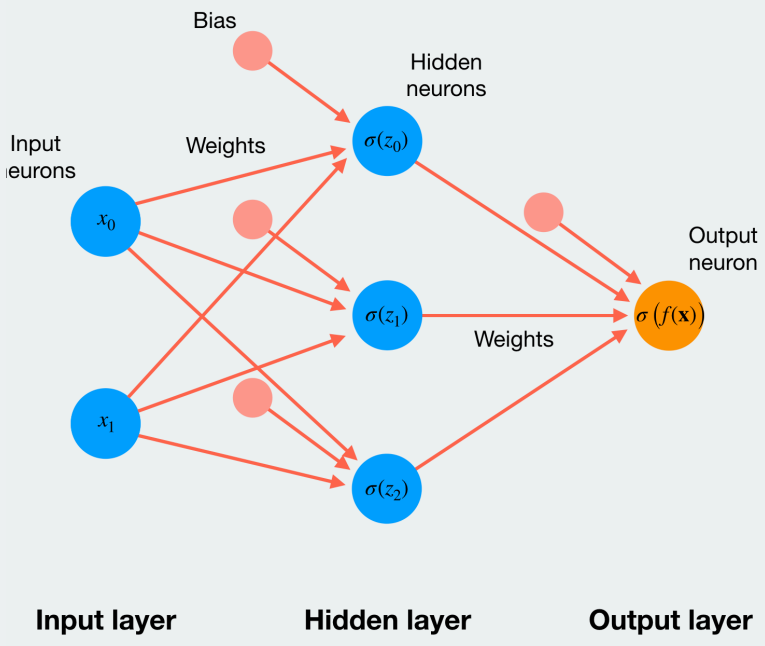
omg! **nested** linear regression is actually **just** linear regression

Reminders





Multilayer perceptron – overview



- Forward-coupled logistic regressions
- Activation function **necessary**
- Number of *layers* refers to number of layers of weights (left, 2)
- Can be *deep,* i.e. have many *hidden* layers
- Dense/fully-connected layers
- Can have multiple output neurons
- More general name: **Feed forward neural network**

Let's get started with the exercises!

1. If you haven't already: download and install Anaconda (Python 3.8 version) https://www.continuum.io/downloads.

2. Make a folder for this course. Open a terminal (Mac) or a console (PC) and navigate to that folder.

3. Run the following command in your terminal/console:

   ```
   git clone https://github.com/lucian979/neural_networks
   ```

4. In your terminal/console, navigate into the exercises folder inside of the newly created downloaded directory, and run the following command:

   ```
   jupyter notebook week1_exercises.ipynb
   ```