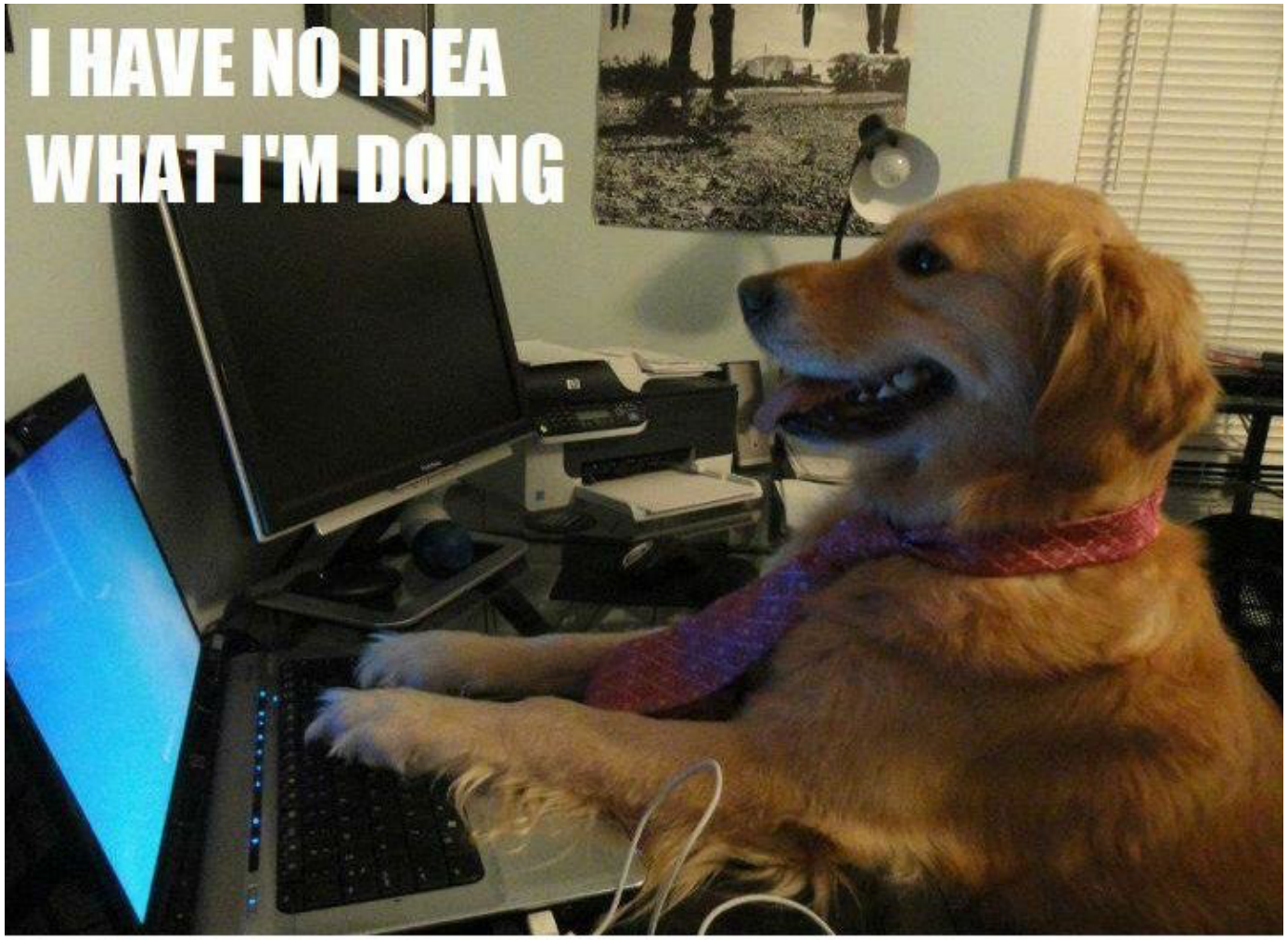
Can we break Neural Networks?

SFB 1294
Data Assimilation

Geo X

Disclaimer



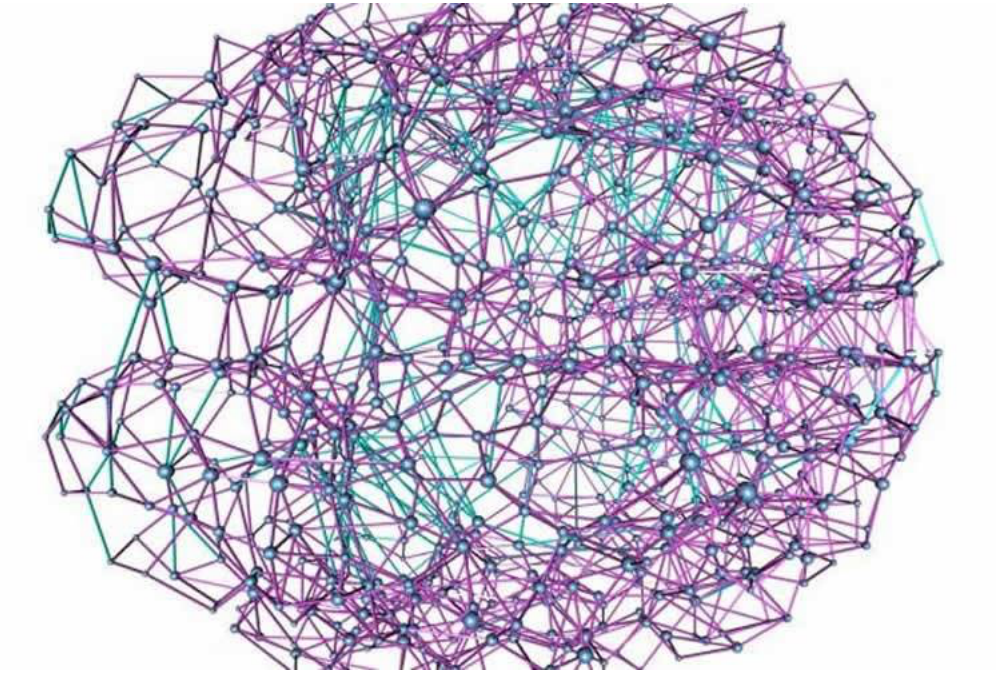
- Git clone https://github.com/MarkSomogyvari/NNexercises.git

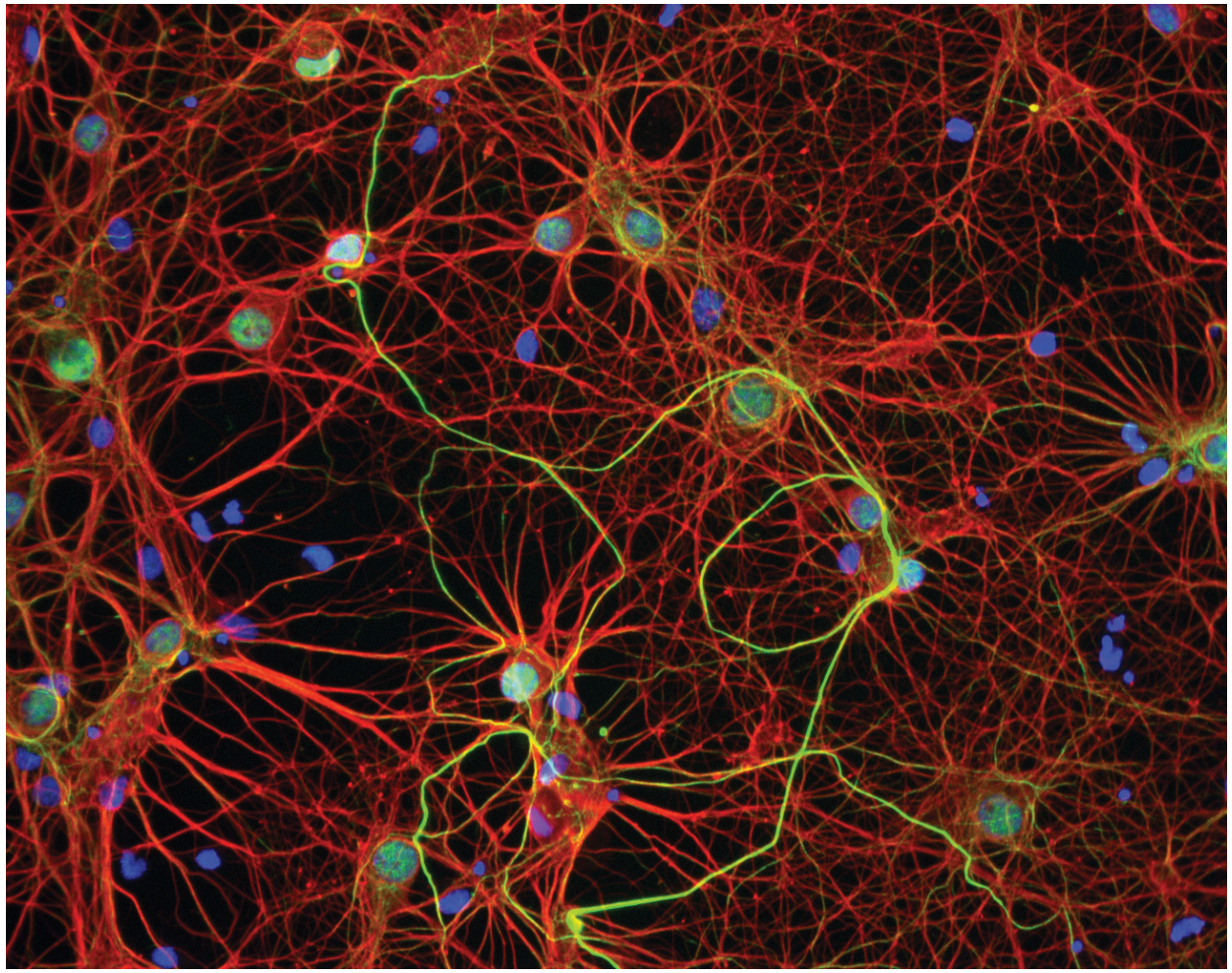
What is a Neural Network?

- a computer system modelled on the human brain and nervous system.

- *"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.*

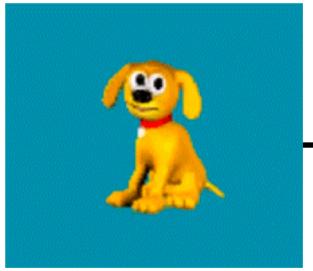
- **Artificial neural networks** (**ANN**) or **connectionist systems** are computing systems vaguely inspired by the biological neural networks that constitute animal brains.





Opentextbc.ca

How do Neural Networks work?



$(R_{1,1}, G_{1,1}, B_{1,1}, \dots)$

NN

DOG

$(CAT = 0.3, DOG = 0.7)$

It's a function!

NN

These are all dogs

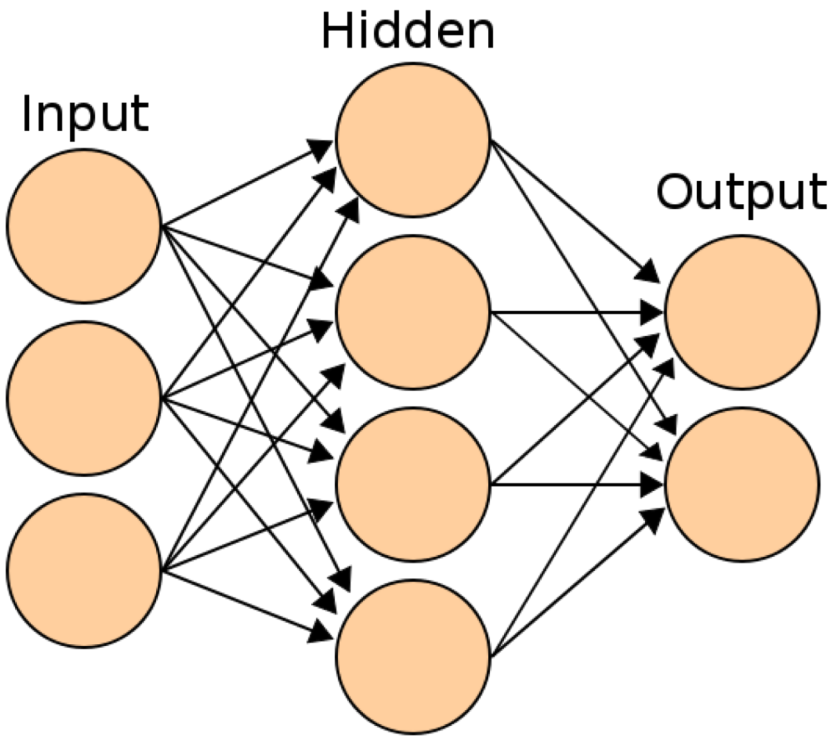
Like calibration

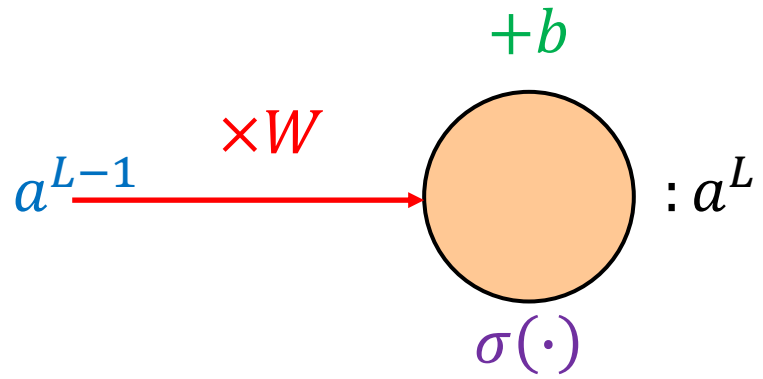
NN

Still a dog

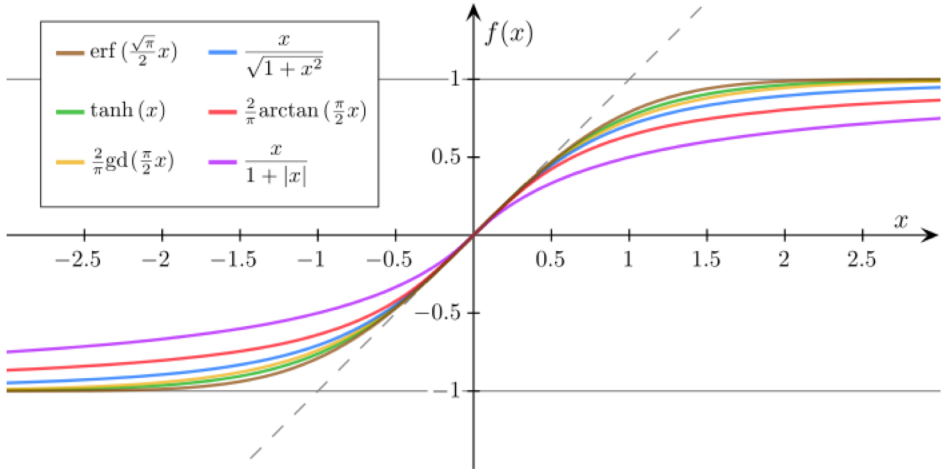
$(CAT = 0, DOG = 1)$

This is great!

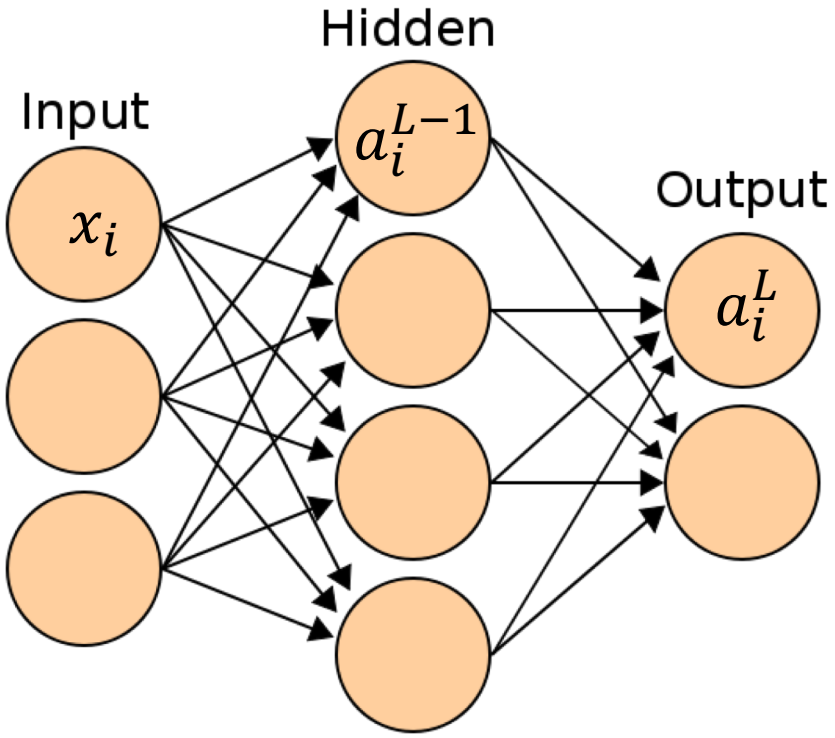




$$a^L = \sigma(\underbrace{W a^{L-1} + b}_{z})$$



Sigmoid functions



Linear combination of vectors
+
nonlinearity

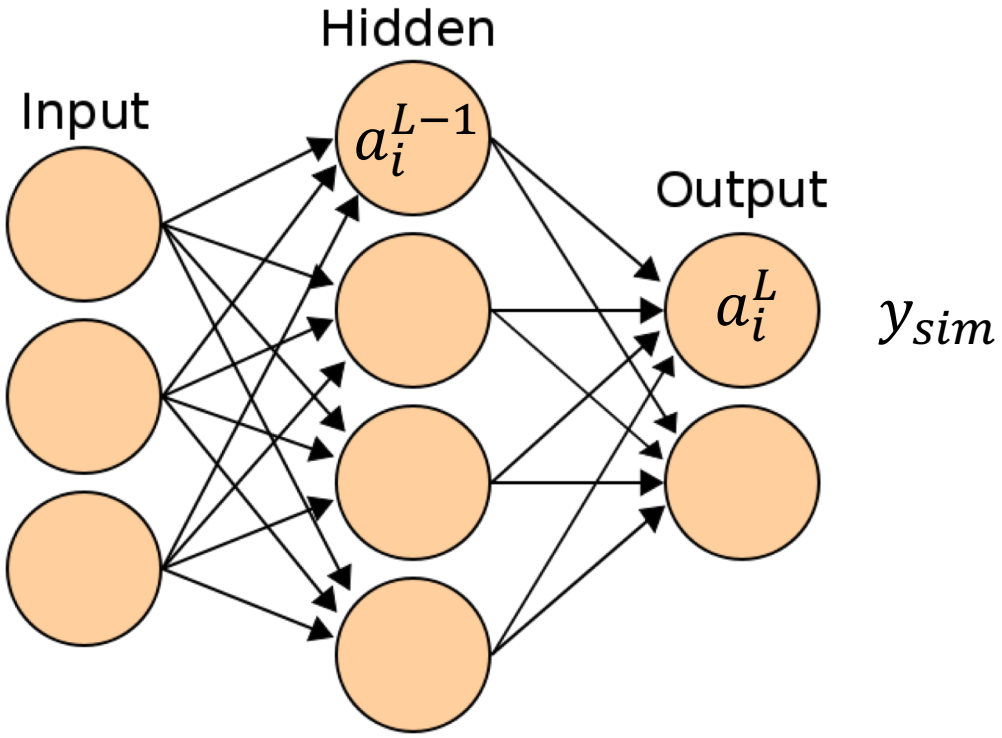
$$a_i^L = \sigma(z_i^L) = \sigma\left(\sum_j W_{ij} a_j^{L-1} + b_i\right)$$

Hyperparameters: n_{layers}, n_{nodes} …

Parameters: $W_{ji}^{(l)}, b_i^{(l)}$

Training

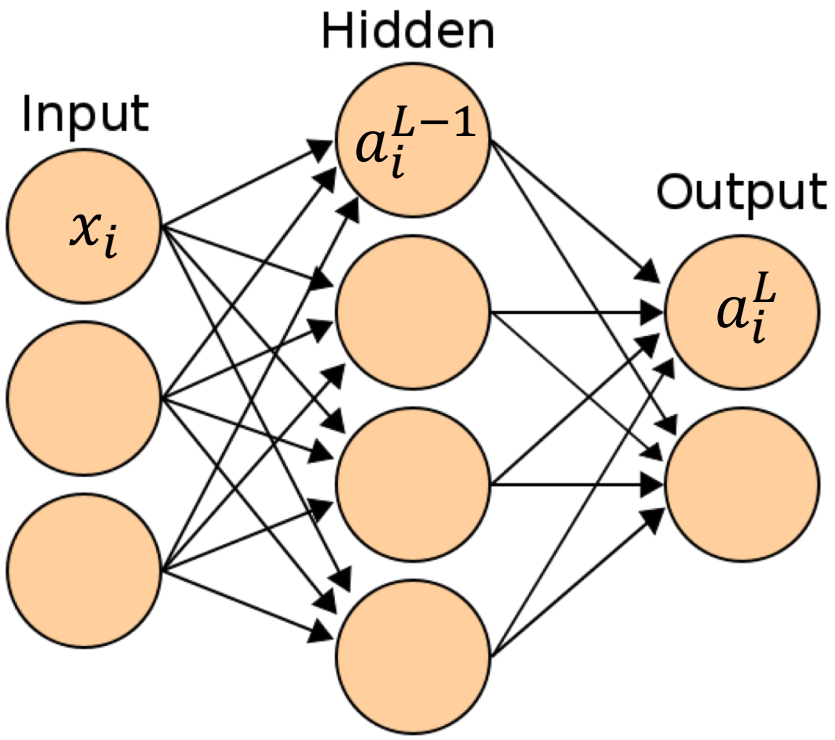
Training



$$C = f(y_{sim}, y_{des})$$

$$C = \frac{1}{N}\sum_i (y_{sim} - y_{des})^2$$

Adjust $\boldsymbol{W}$ and $\boldsymbol{b}$ to minimize cost function.

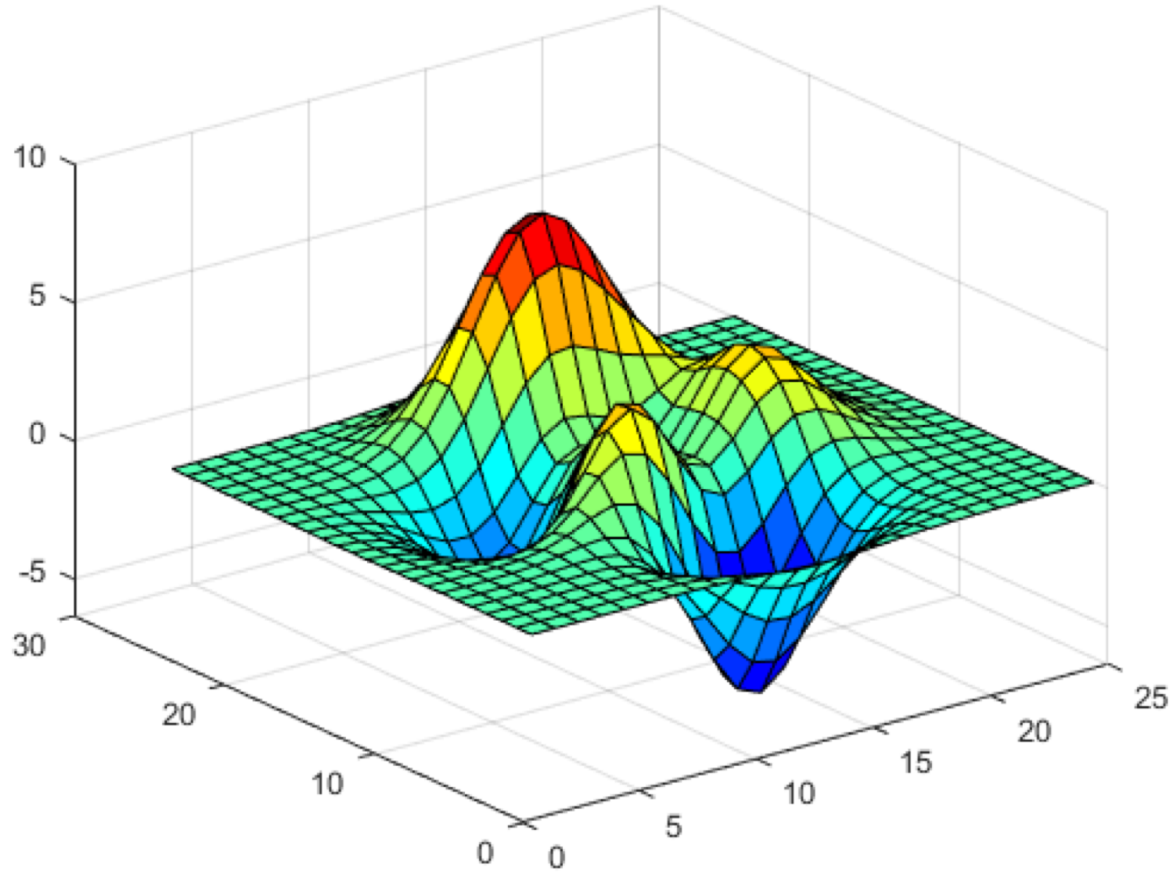


$$a_i^L = \sigma\left(\sum_j W_{ij} a_j^{L-1} + b_i\right)$$

$$C = \frac{1}{N}(y_{sim} - y_{des})^2$$

$$C = f\left(W_{ij}^l, b_i^l\right)$$

Gradient descent



$$C = \frac{1}{N}(y_{sim} - y_{des})^2$$

$$C = f\left(W_{ij}^{l}, b_{i}^{l}\right)$$

$$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial W}$$

$$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial b}$$

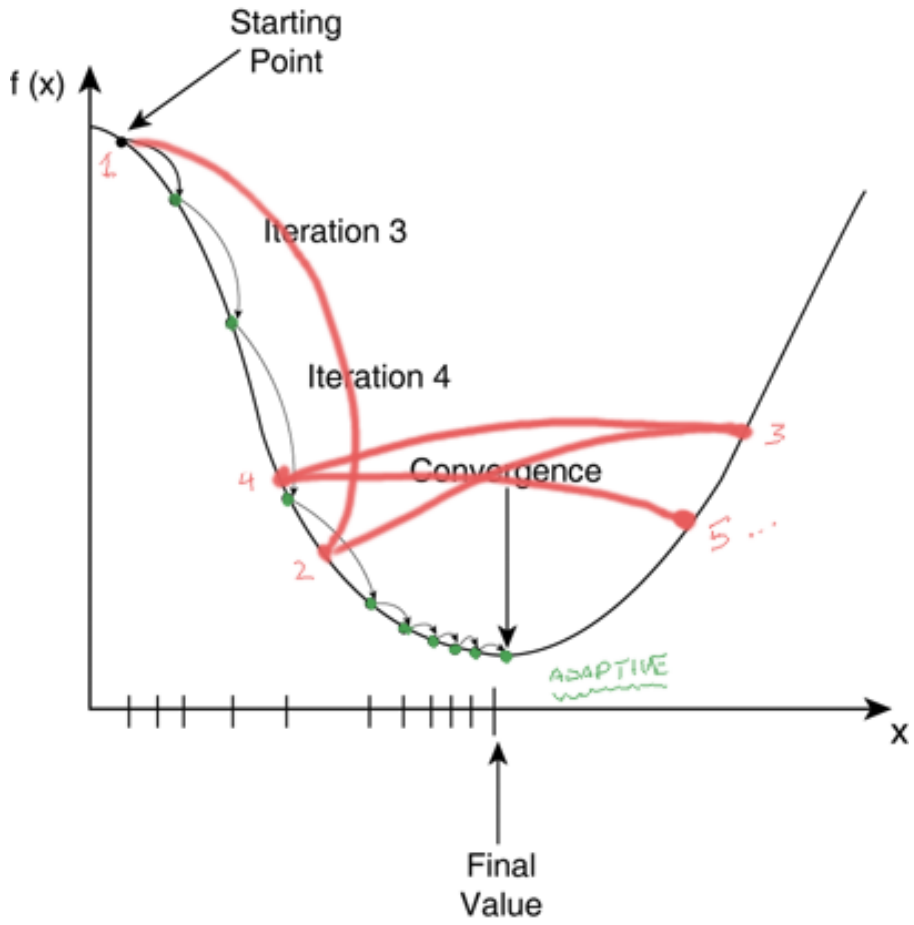
Backpropagation algorithm

1. Input data
2. Feedforward $z^l = w^l a^{l-1} + b^l \qquad a^l = \sigma(z^l)$
3. Output error $\delta^L = \nabla_a C \odot \sigma'(z^L)$
4. Backpropagate error $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$
5. Gradients: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \qquad \frac{\partial C}{\partial b_j^l} = \delta_j^l$

Gradient descent

- Calculate gradients with data
- Change parameters

$$-\eta \nabla C$$

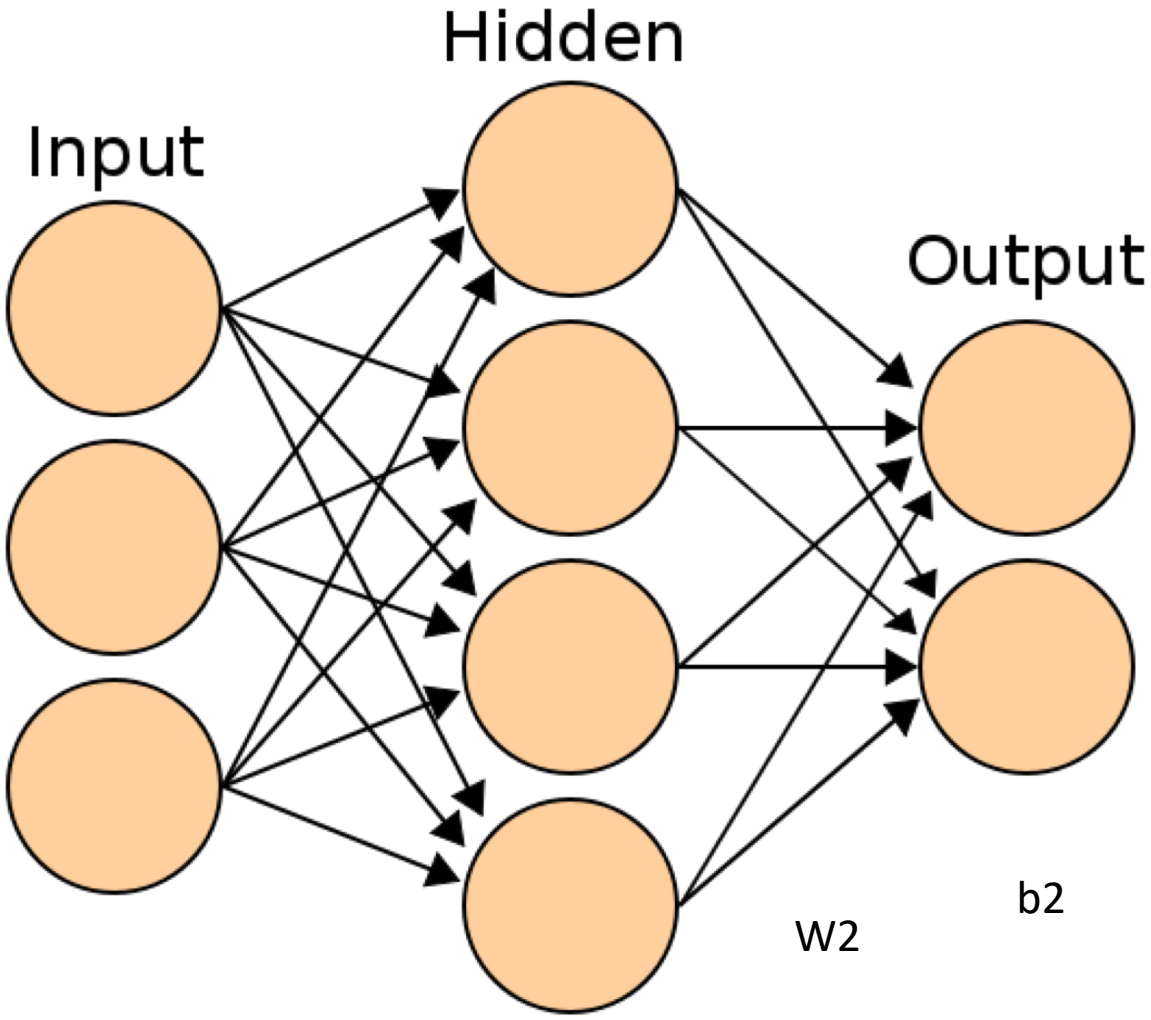


Example 1. - Clustering

Input: (x,y) coordinates
Output: 0 or 1

Open: Class_exercise.ipynb

X - coordinates

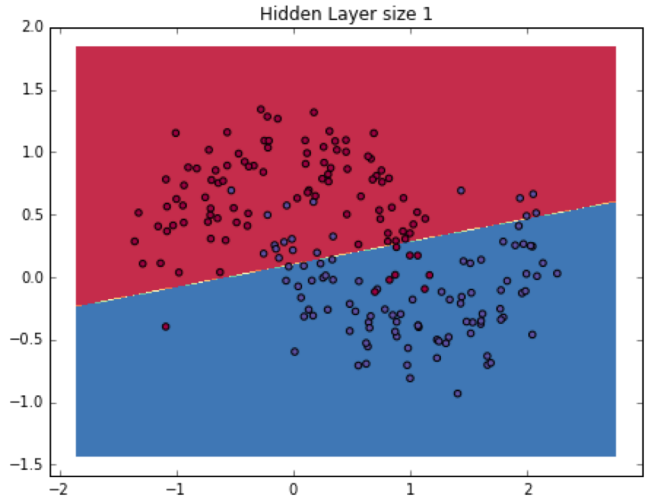


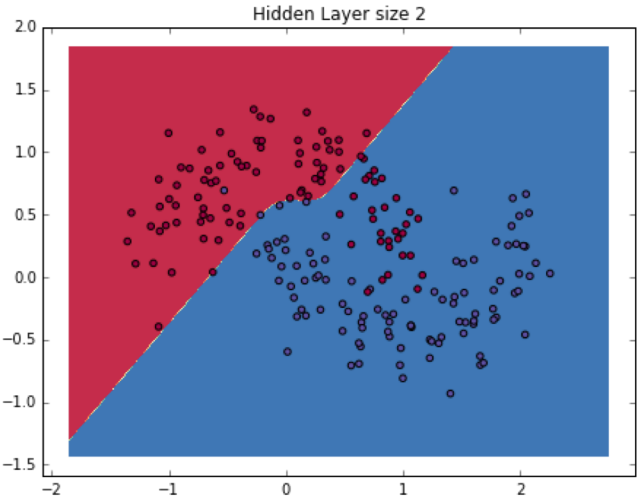
Y - binary

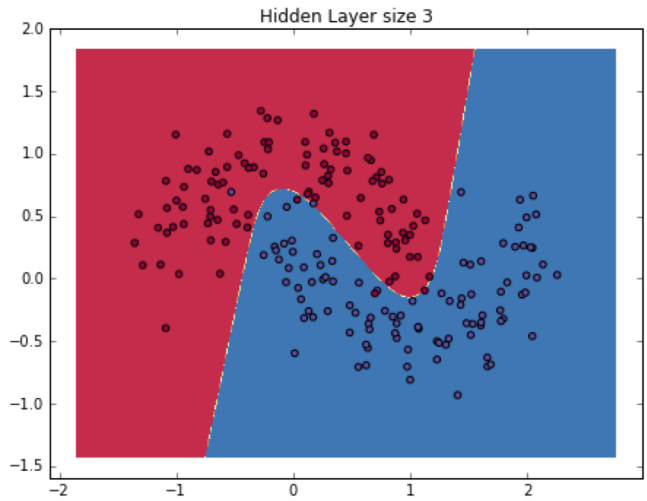
W1 b1 W2 b2

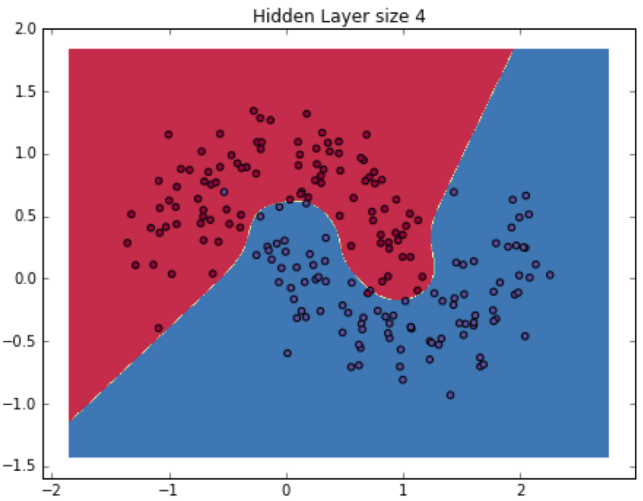
a1 = tanh(x*W1+b1)

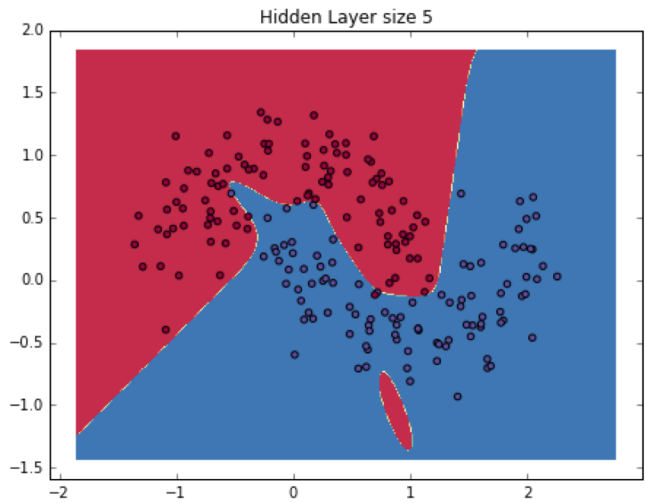
y=softmax(a1*W2+b2)

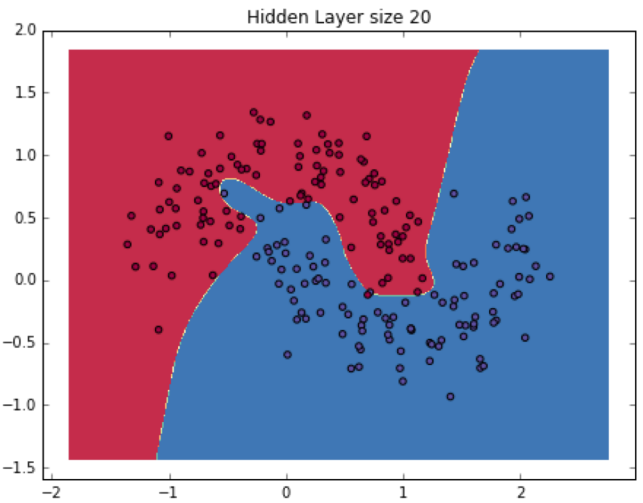

Hidden Layer size 1

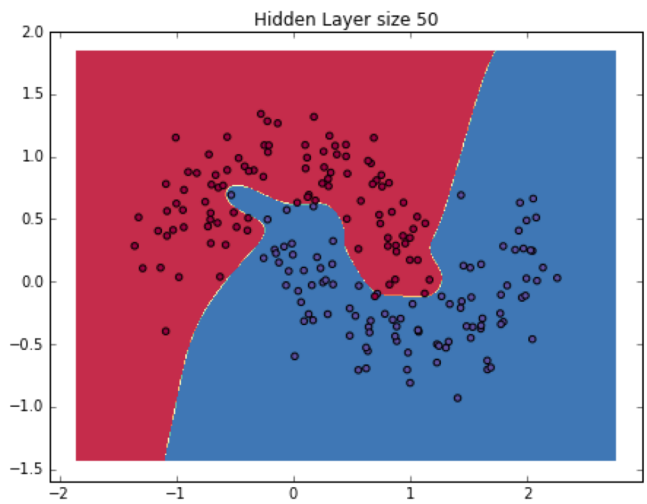

Hidden Layer size 2


Hidden Layer size 3

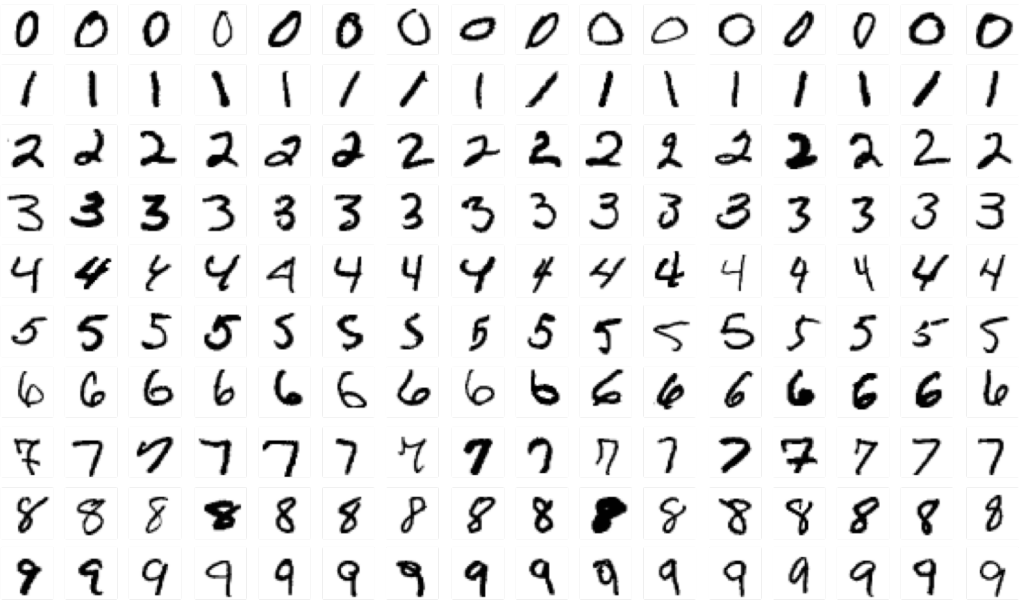

Hidden Layer size 4


Hidden Layer size 5

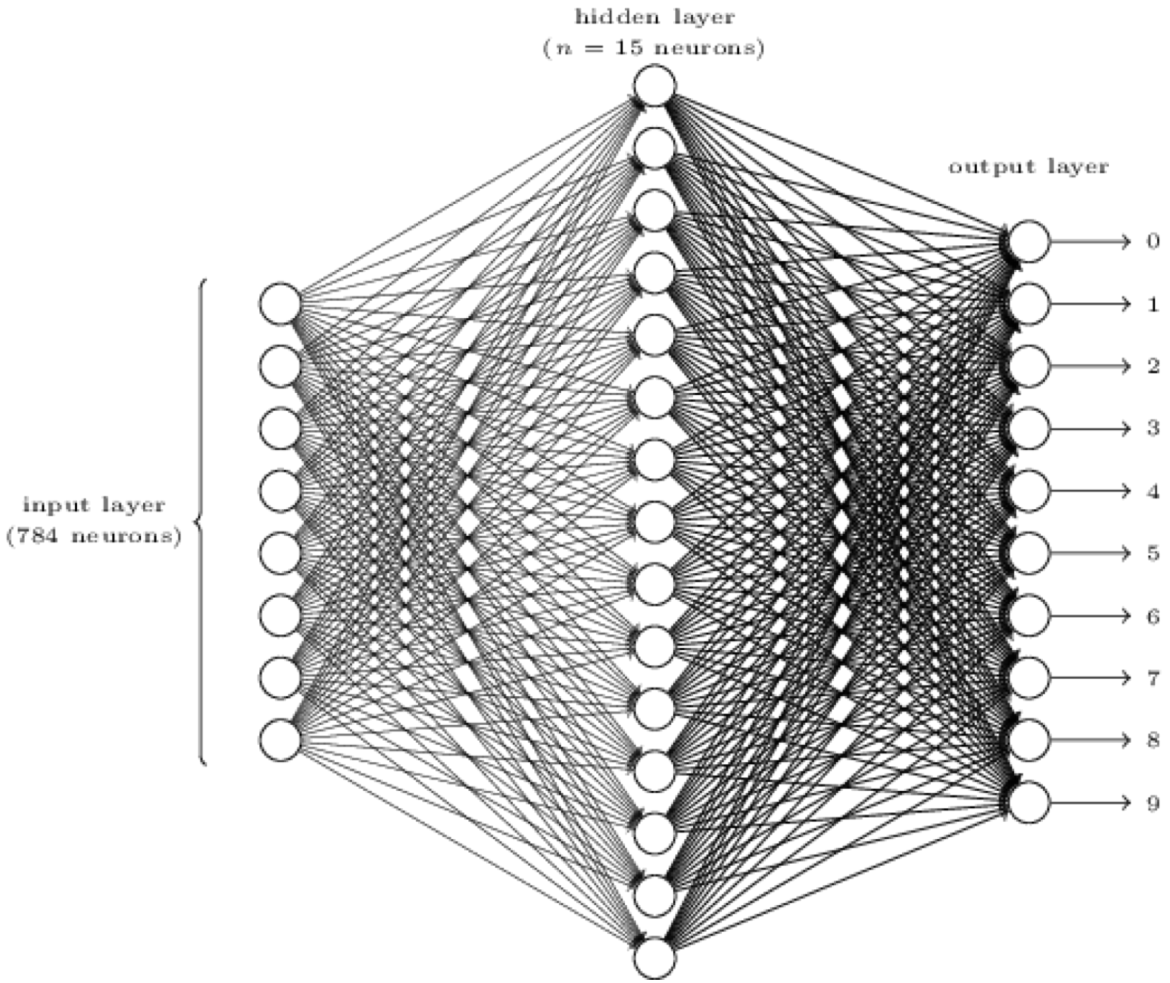

Hidden Layer size 20


Hidden Layer size 50

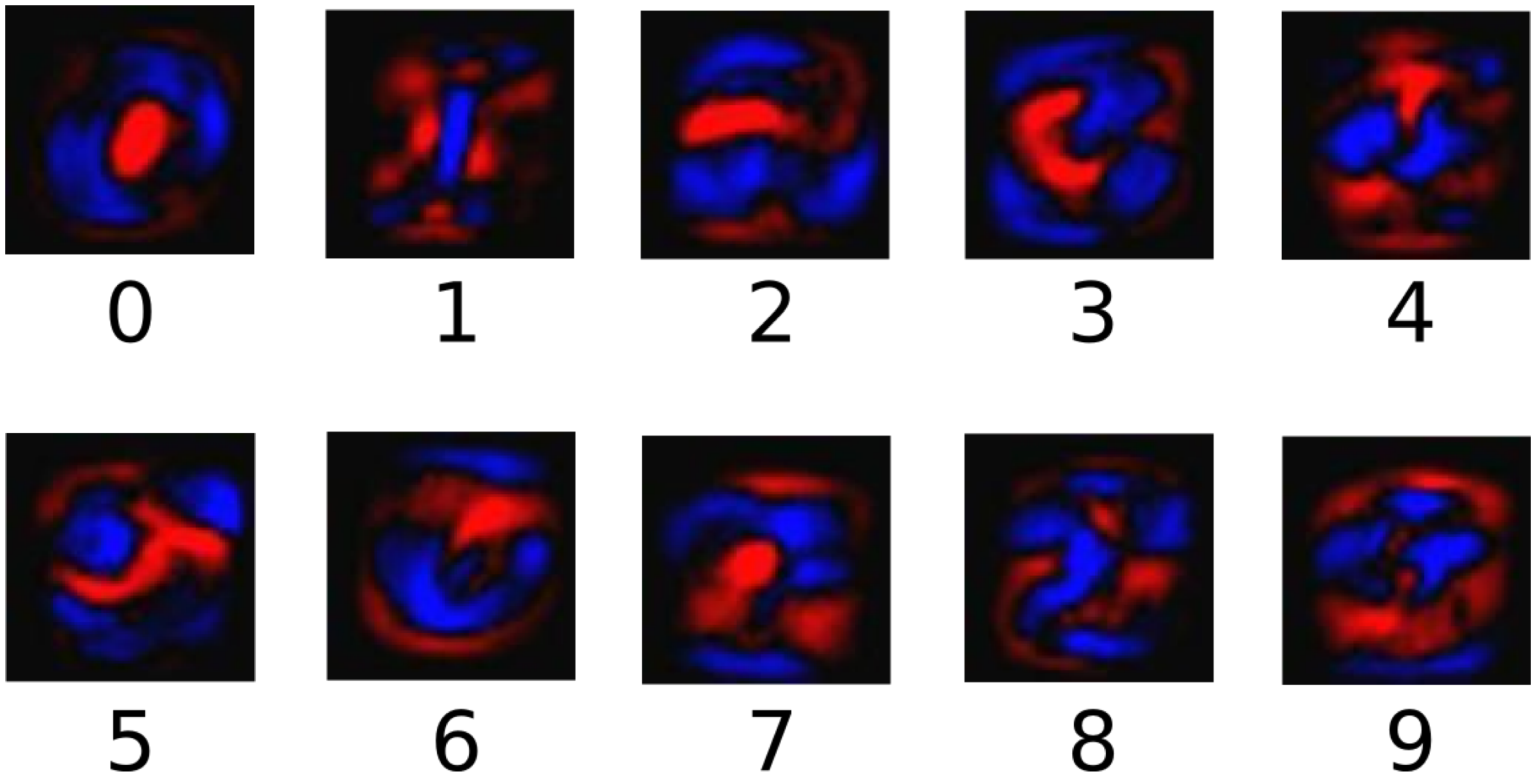
Example 2. – Digit recognition

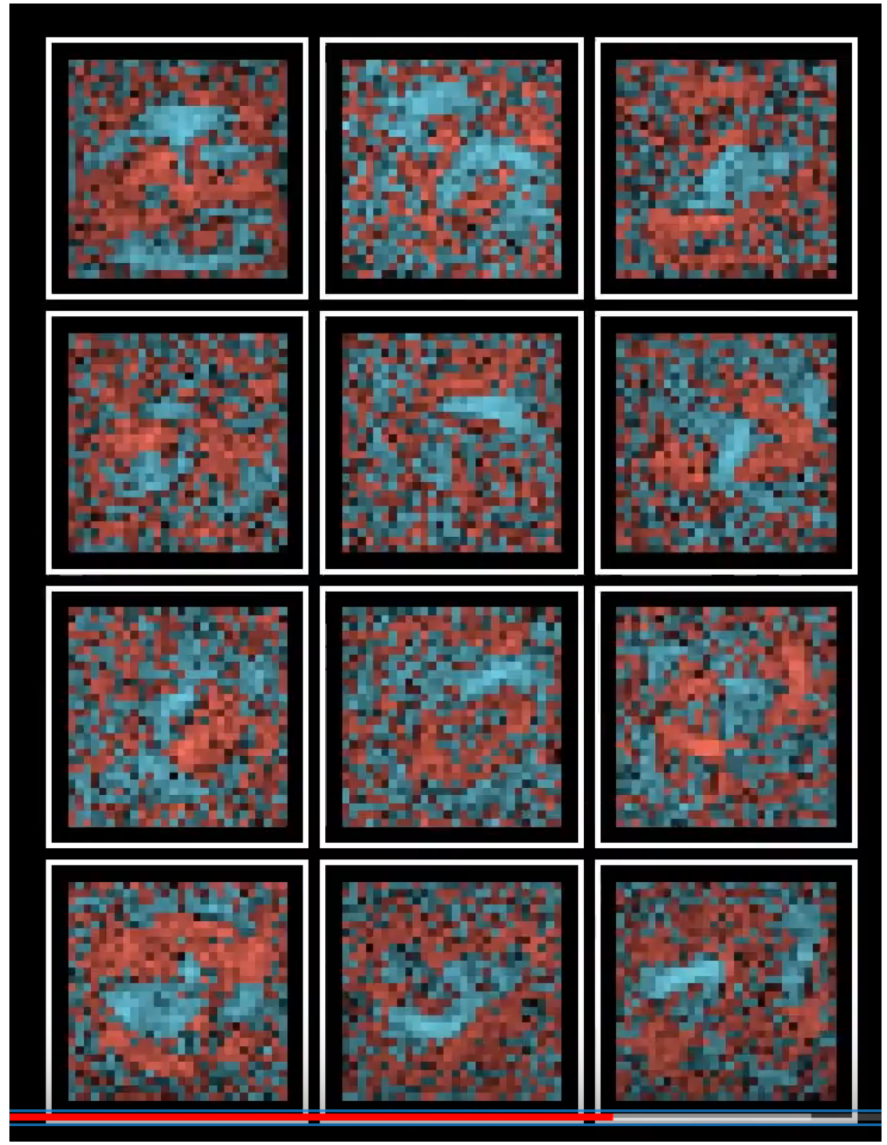


MNIST dataset



Learnt weights – what do they mean?










(b) (c) (d)

Source: Carnegie Mellon University

There is a lot more out there…

- https://youtu.be/qv6UVOQ0F44 – unsupervised learning with genetic algorithm

- https://youtu.be/b5xpXecR3LY

Further reading…

- Youtube: 3Blue1Brown – Neural networks
- http://neuralnetworksanddeeplearning.com/chap1.html
- http://karpathy.github.io/neuralnets/