AI for cyber-physical systems
Part 2 – Neural Networks & Deep Learning

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

European Commission | Horizon 2020 European Union funding for Research & Innovation

WAZIUP
IoT – from idea to reality
WAZIHUB

HUBiquitous
Paving for the next 10 years of innovation in IoT and AI

PRIMA
IN THE MEDITERRANEAN AREA

Intel-IrriS RESILINK
Advanced and disruptive IoT/AI technologies targeting the smallholder community for increased resilience

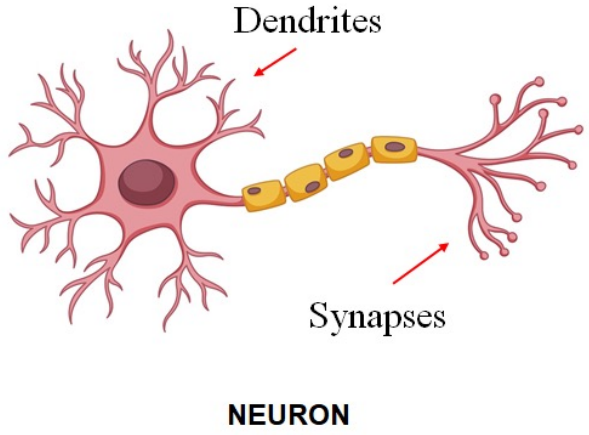
Acknowledgements

- AI is probably one of the hot topics with the most incredible amount of also incredible quality contributions & tutorials videos!
- Setting up this course would not have been possible without all these contributors!
- I've crawled the web for hours (days), trying to take knowledge from multiple sources and trying to present all these impressive resources in a simple and comprehensive way for my students
- I tried to mention the sources of materials I've borrowed from the web as much as I could
- A special "thank you" for all the incredible tutorial videos from Machine Learnia (https://machinelearnia.com/, Guillaume Saint-Cirgue) and his YouTube channel (https://www.youtube.com/@MachineLearnia/videos)
- A special "thank you" akso for the complete and very clear videos from freeCodeCamp.org, especially on DL (https://www.youtube.com/watch?v=dPWYUELwIdM)
- There are also all the articles from https://towardsdatascience.com/

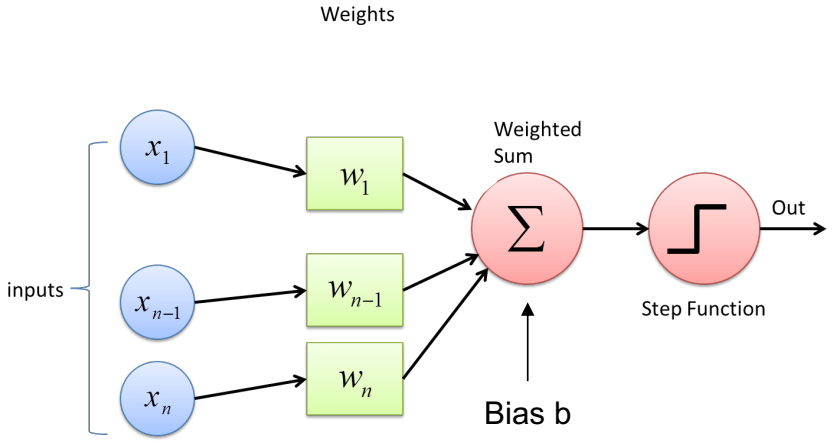
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

How does a Perceptron work?

- Single Layer Perceptron (SLP) receives the value of the attributes of an input, just as dendrites do in a neuron
- Each attribute has a **weight** w that measures its *contribution* to the final result, which is the sum of the multiplications of inputs of each attribute by its corresponding weight
- The final output depends on the activation function: here, if the sum is >0 Perceptron returns a value of 1, otherwise it yields 0

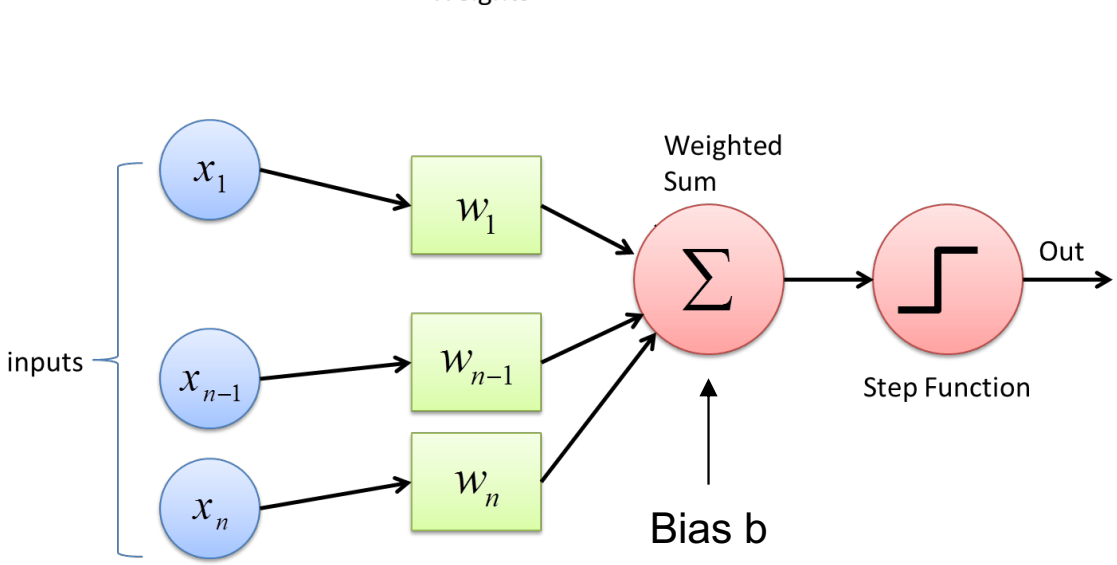


NEURON



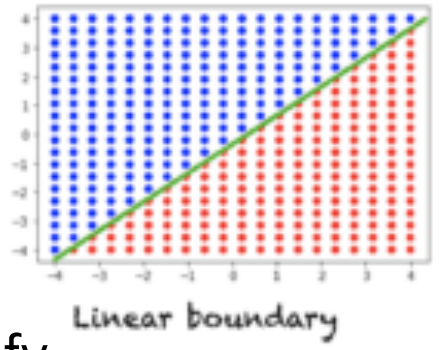
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

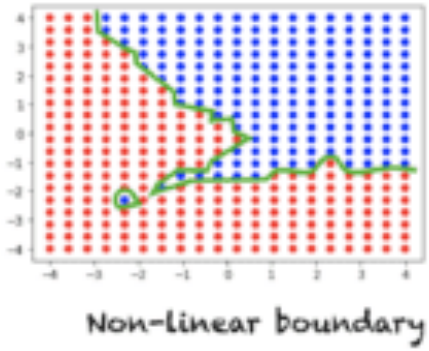
What can an SLP do?



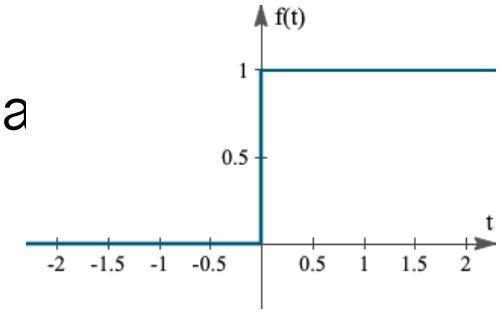
$$\hat{y} = \Theta(w_1 x_1 + w_2 x_2 + ... + w_n x_n + b)$$
$$= \Theta(\mathbf{w} \cdot \mathbf{x} + b)$$

$$\text{where} \quad \Theta(v) = \begin{cases} 1 & \text{if } v \geqslant 0 \\ 0 & \text{otherwise} \end{cases}$$





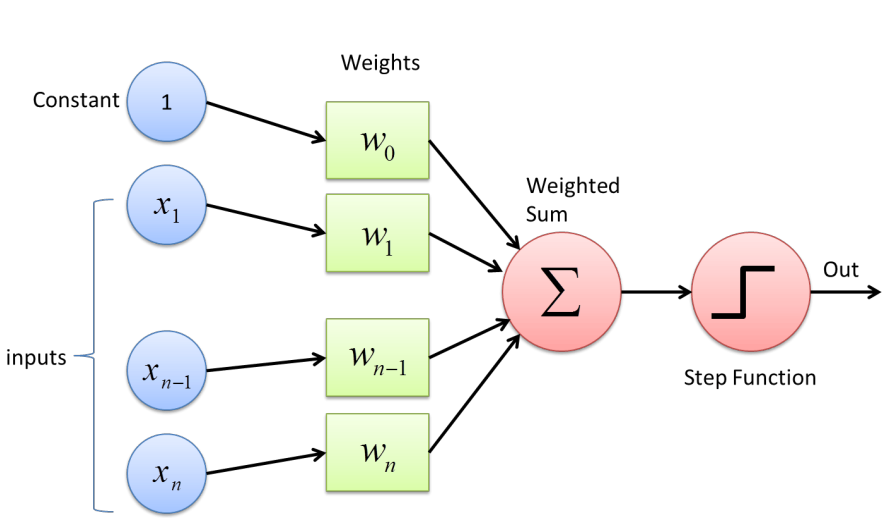
- SLP is the simplest type of artificial neural networks and can only classify linearly separable cases with a binary target
- Activation functions are mathematical equations that determine the output of a neural network
- Here, the Heaviside step function Θ

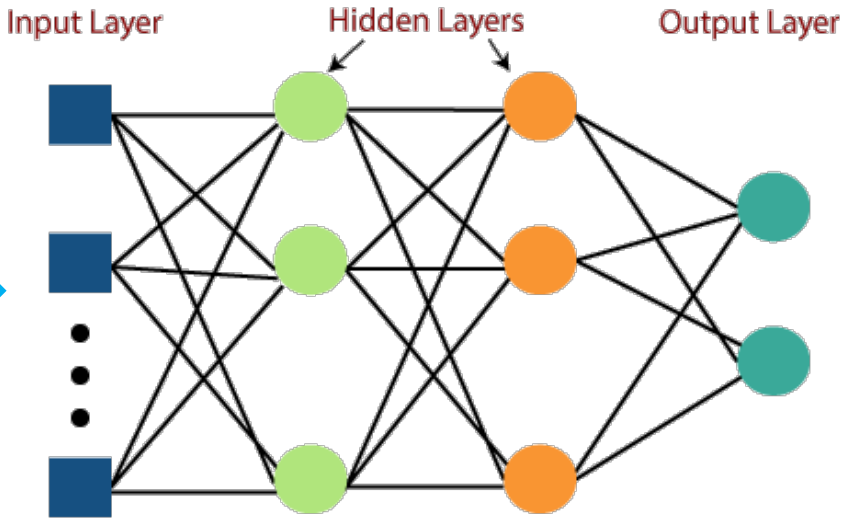


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Multi-Layer Perceptron – MLP

- Multi-Layer Perceptrons (and so-called neural networks) can be considered as more general function approximators and they are able to distinguish data that is not linearly separable
- MLP have several neuron layers with so-called hidden layers
- Hidden layers are "hidden" to the final user who only "see" inputs and outputs

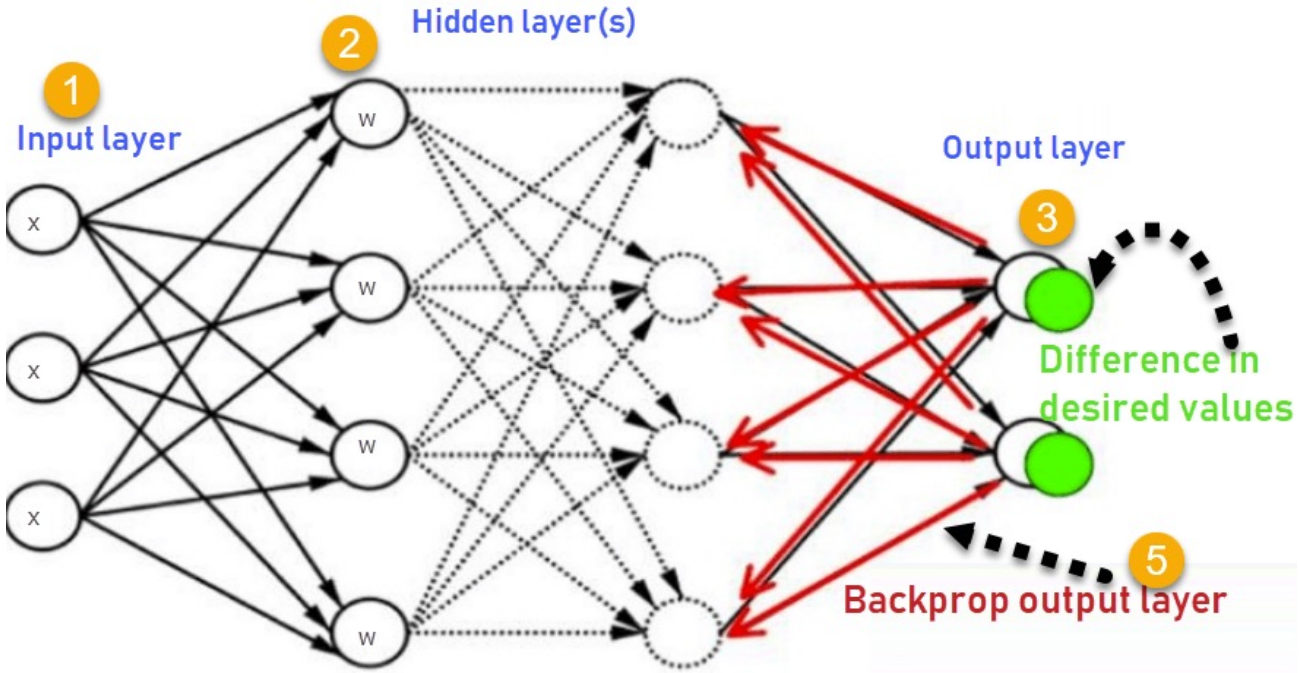




Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Learning in MLP?

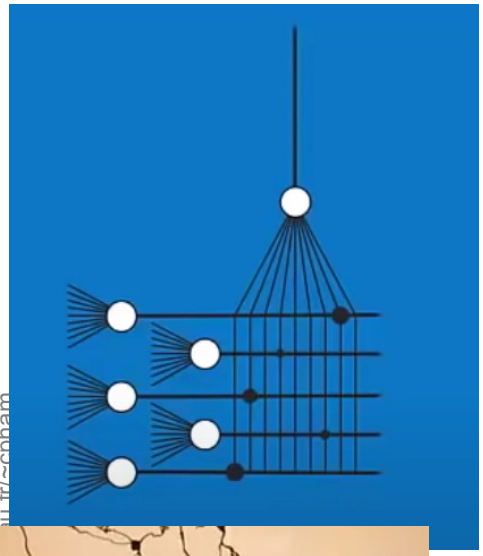
- With more layers, there are more parameters to optimize, so training an MLP is obviously more complex and will take much much more time, if not done in an intelligent way!
- A fundamental method in neural network is back-propagation!

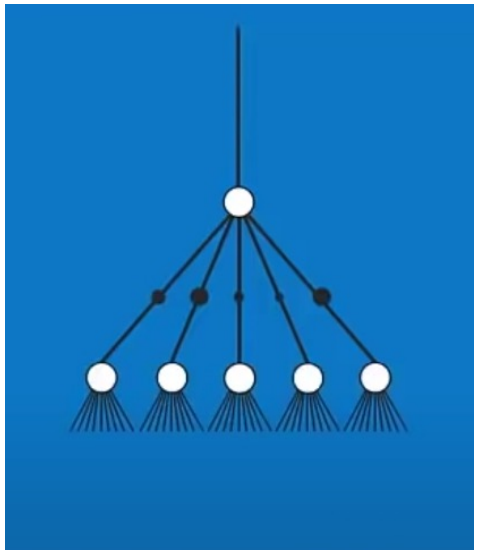


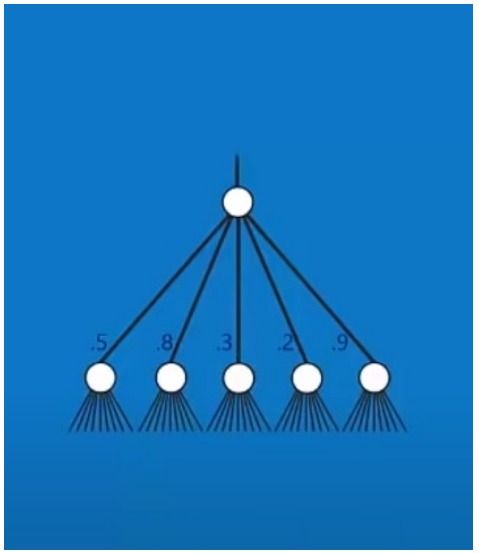
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

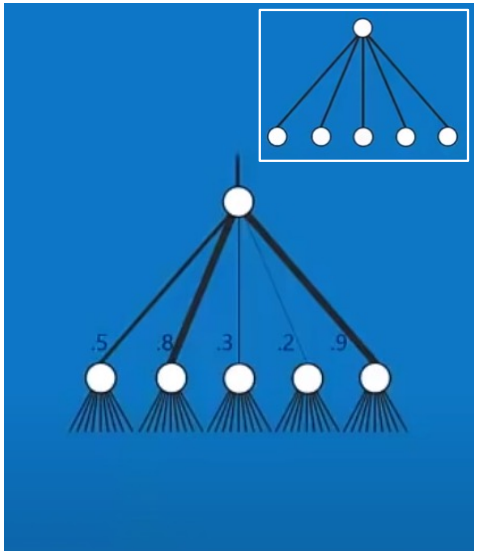
A Deep Dive in Neural Networks

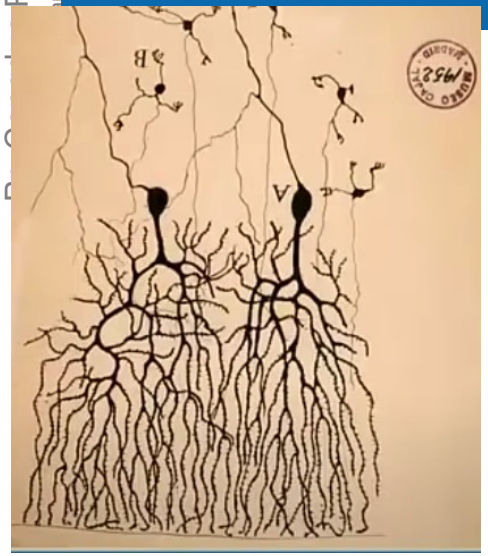
Neural Network in a nutshell

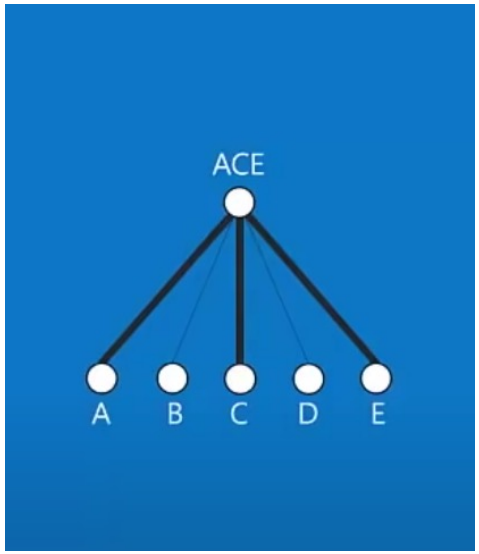


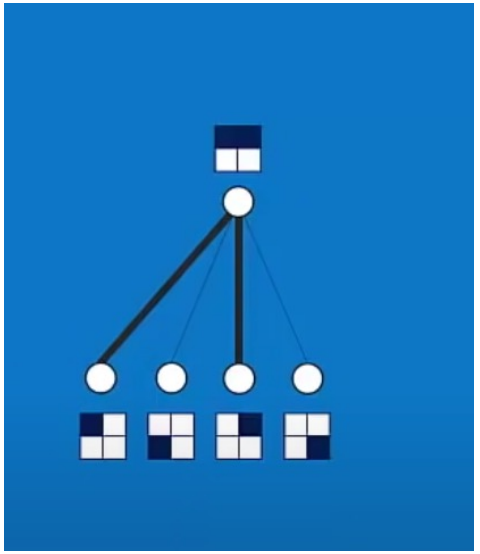


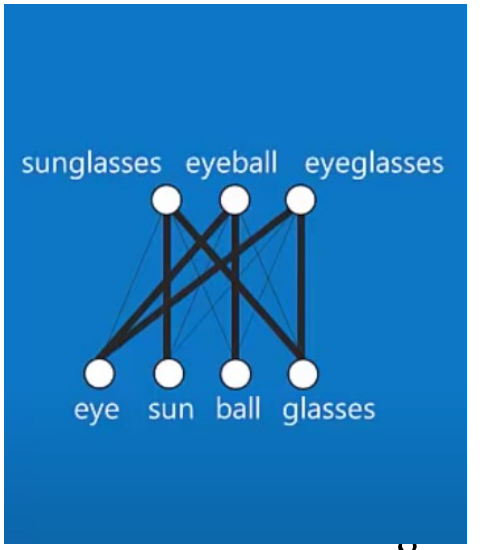








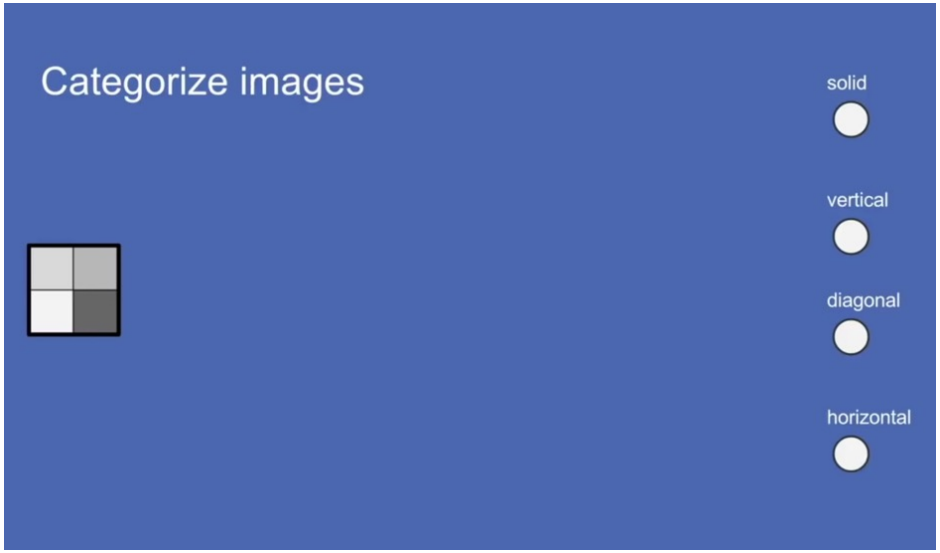


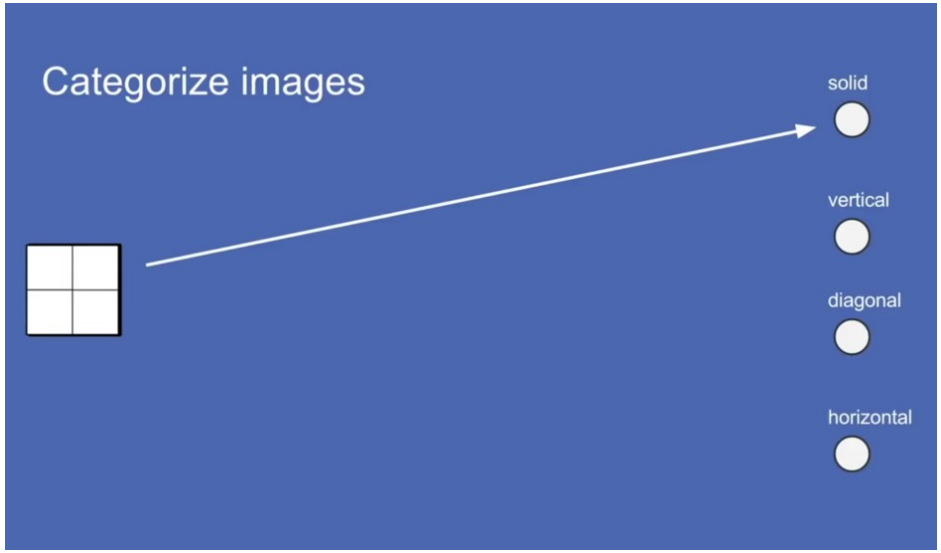


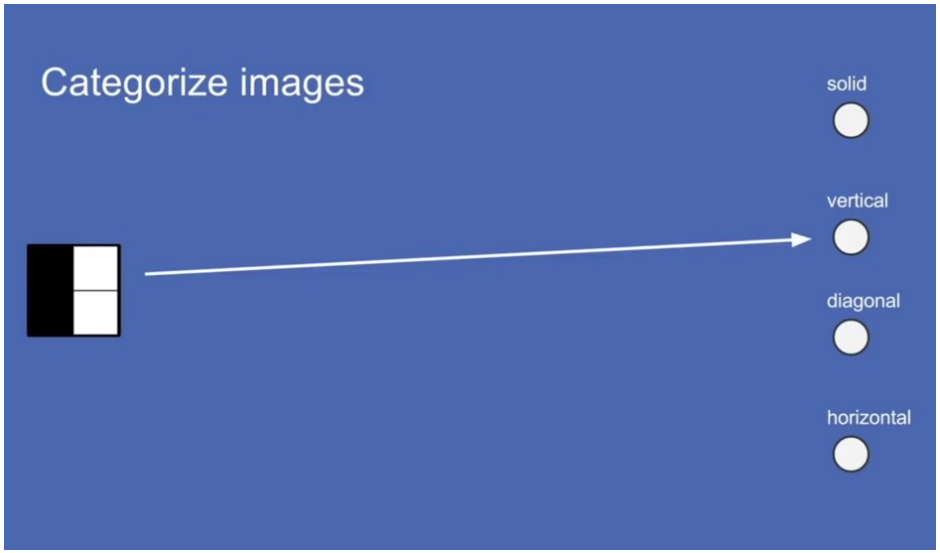
Pictures from https://www.youtube.com/watch?v=Q9Z20HCPnww (Brandon Rohrer)

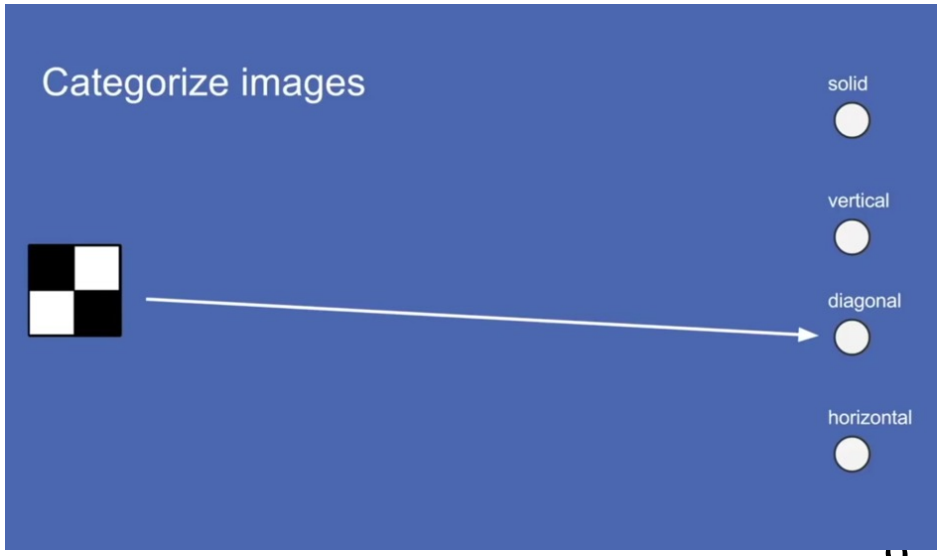
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1: 4-pixel camera





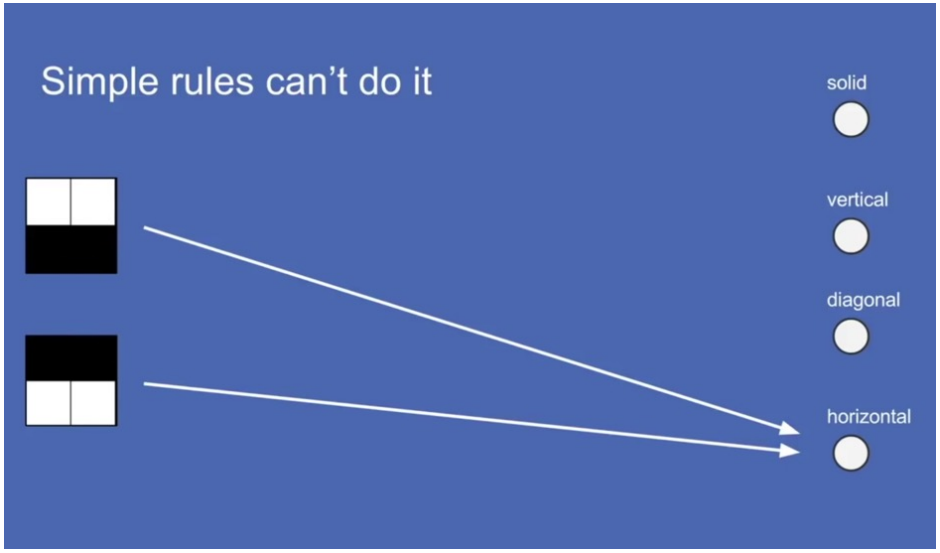


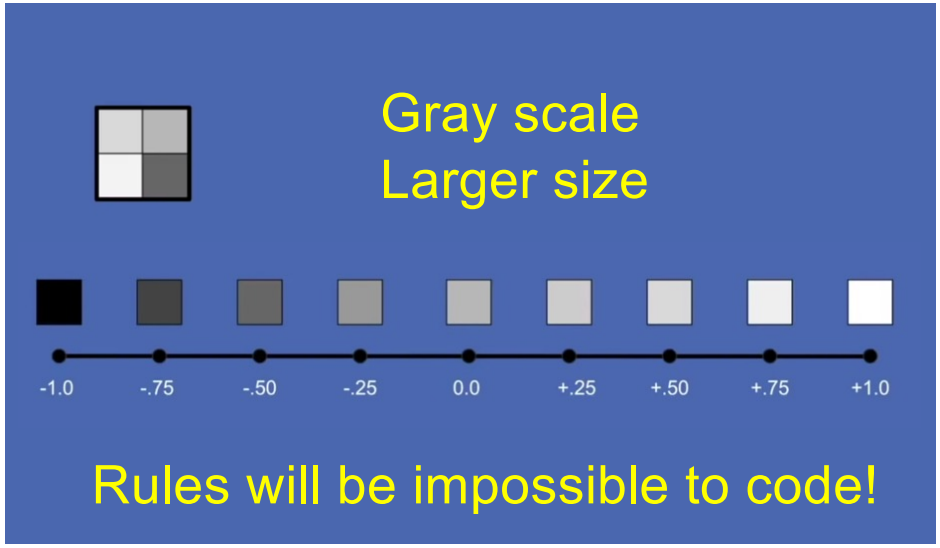


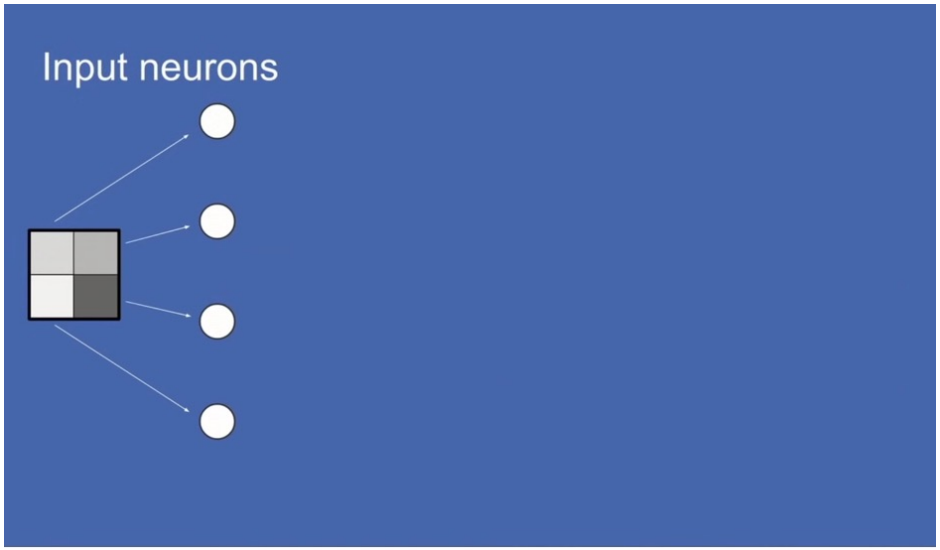
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

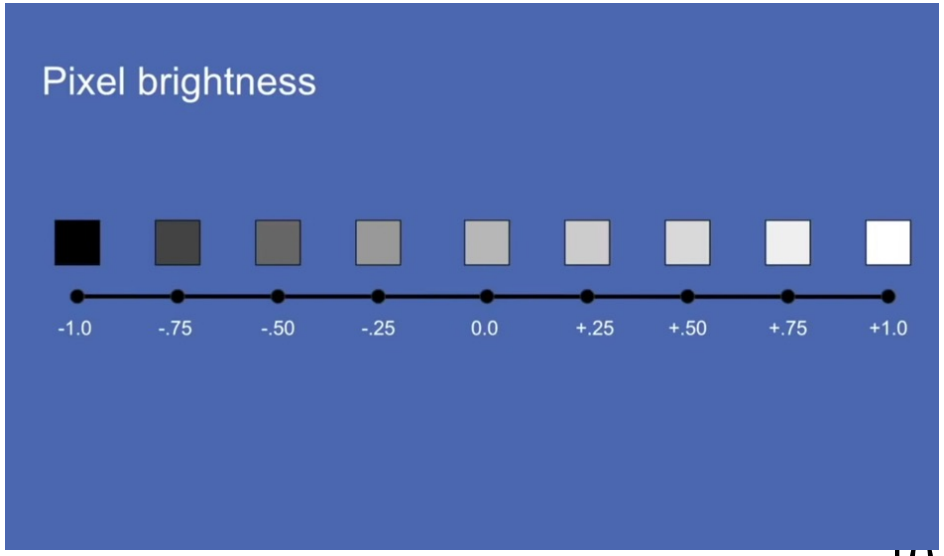
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't





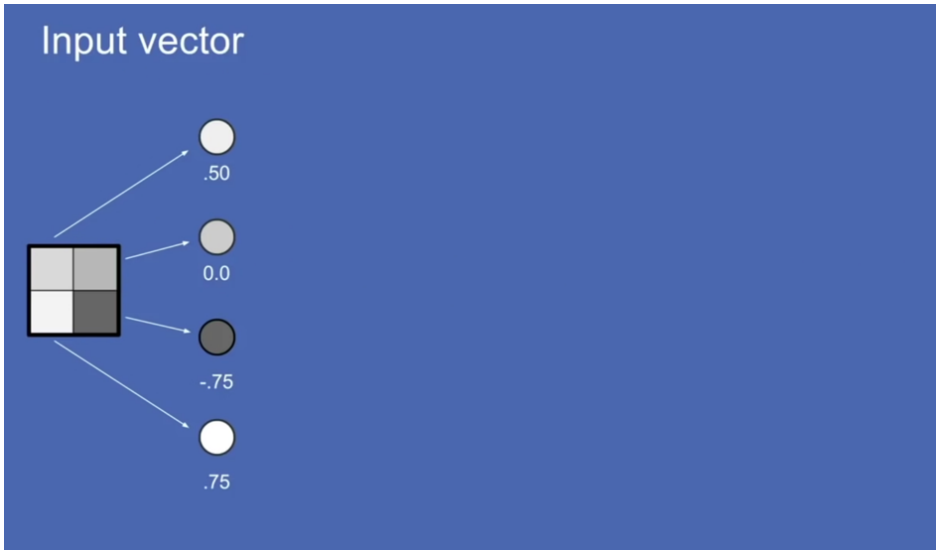


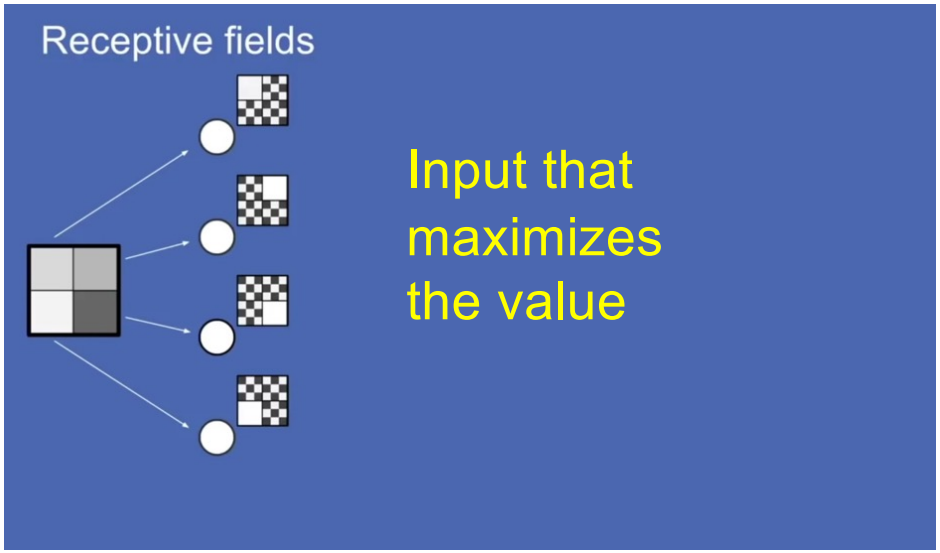


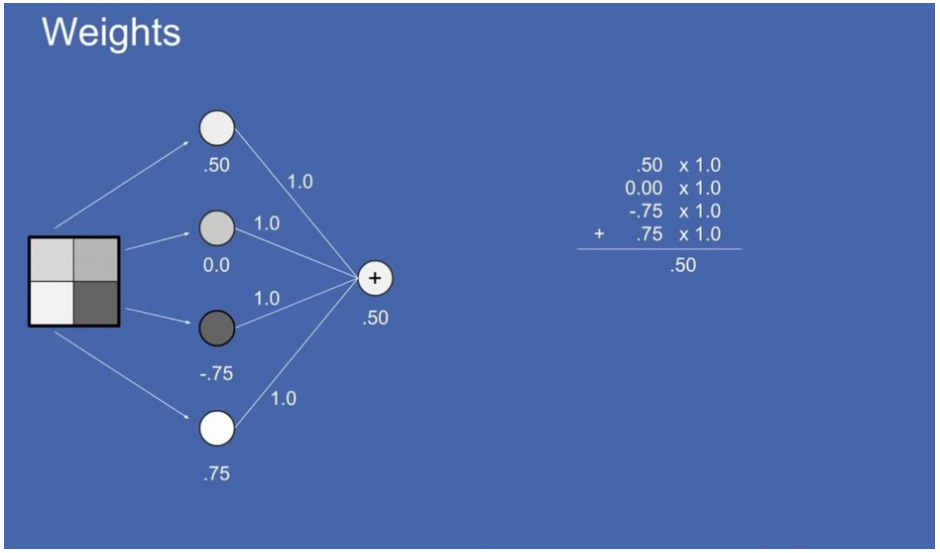
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

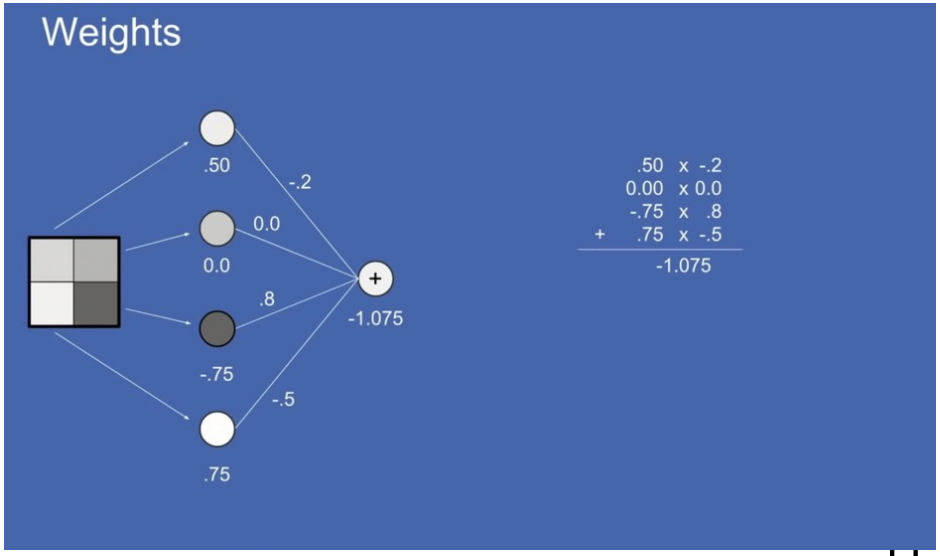
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't



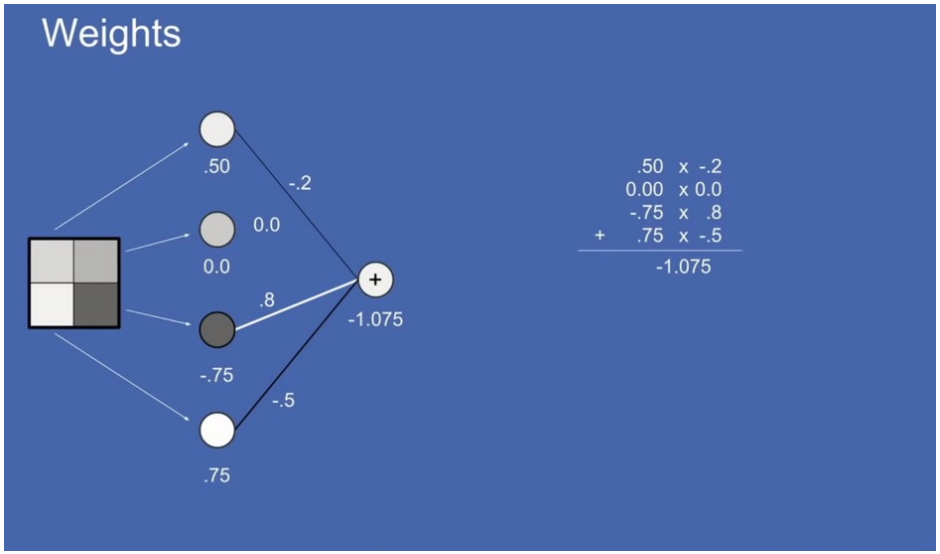


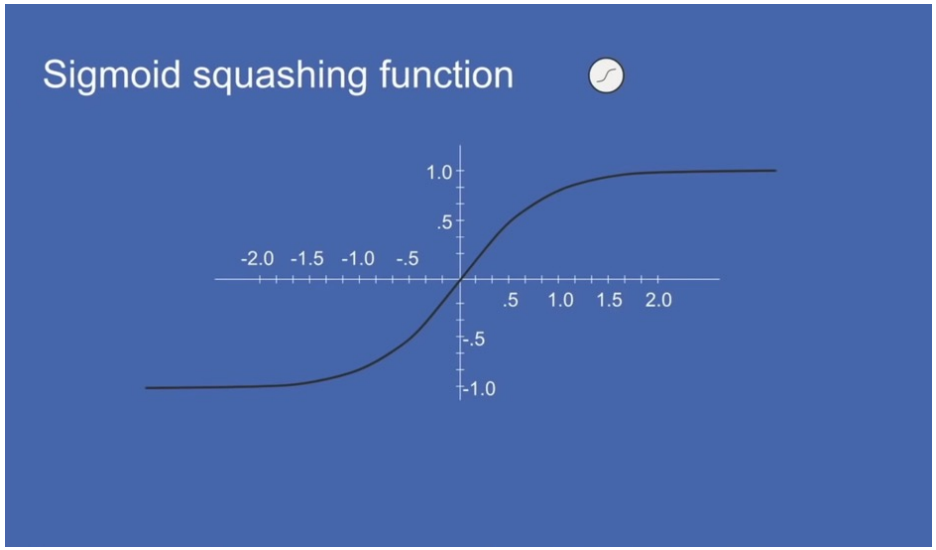


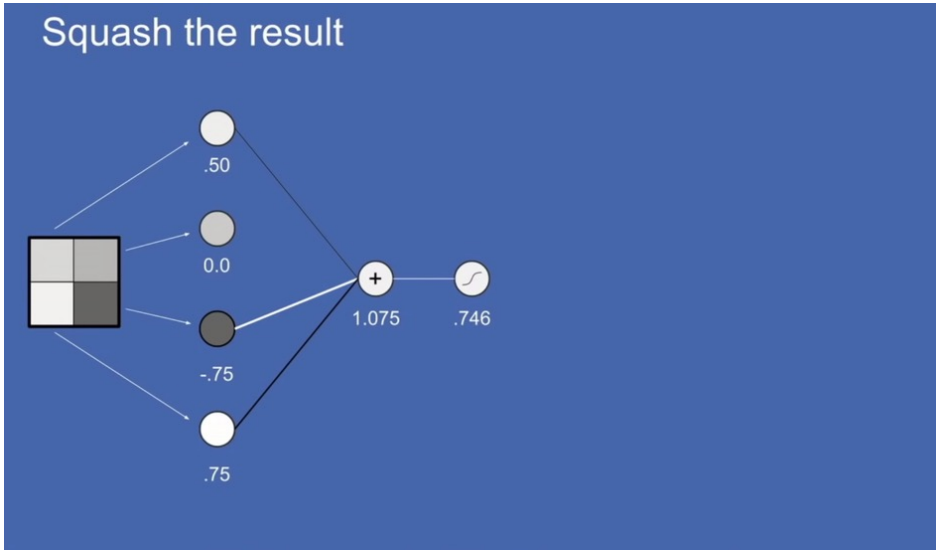


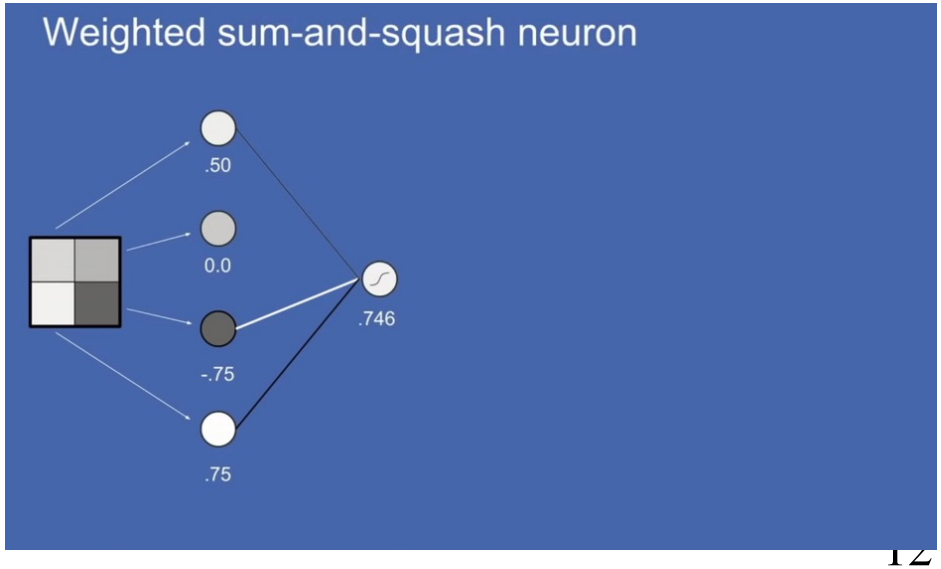
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't



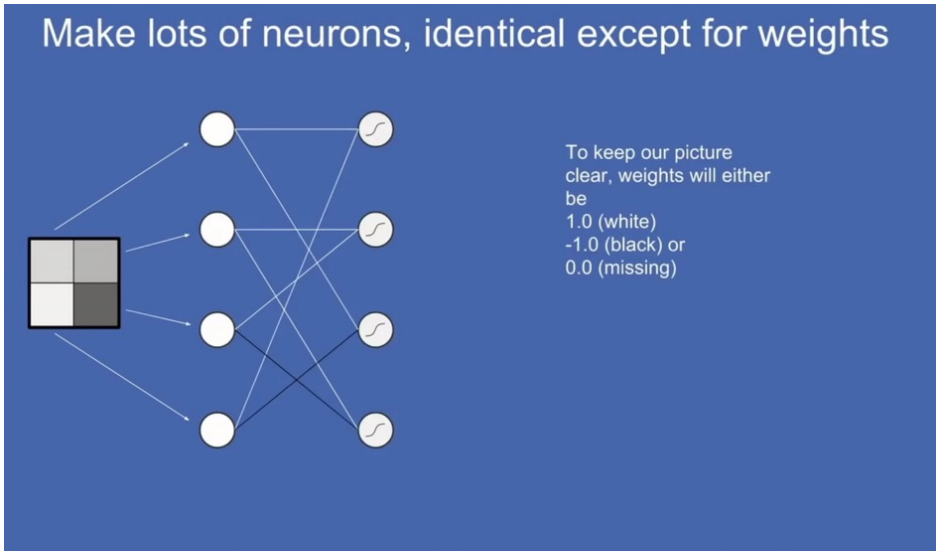


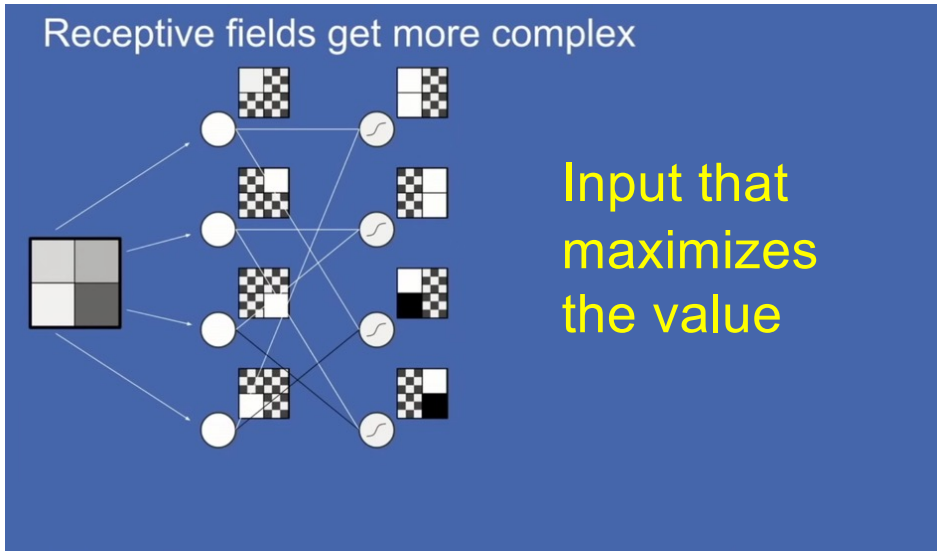


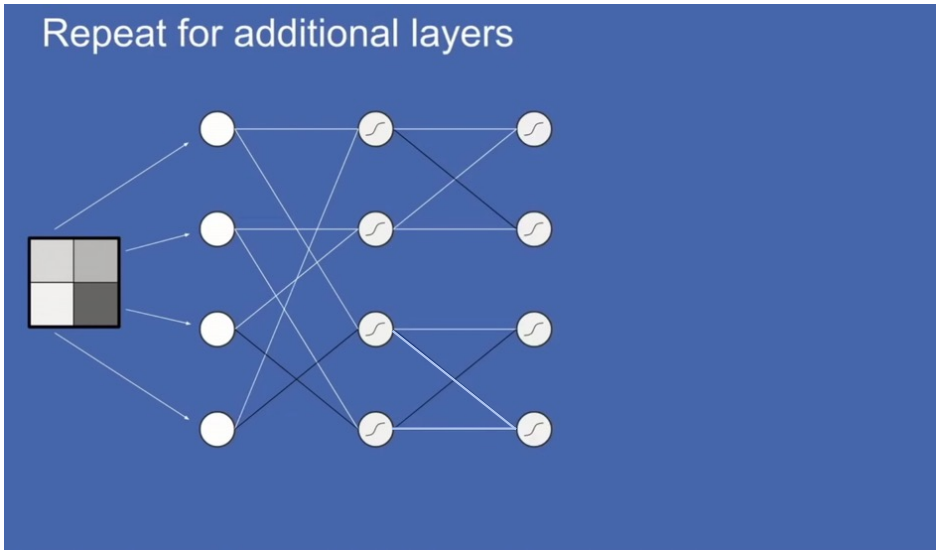


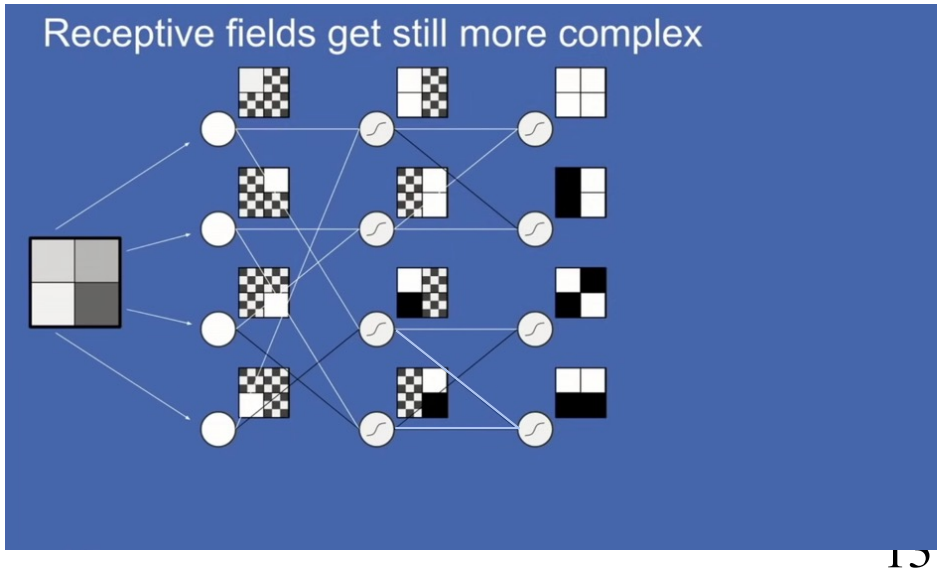
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't



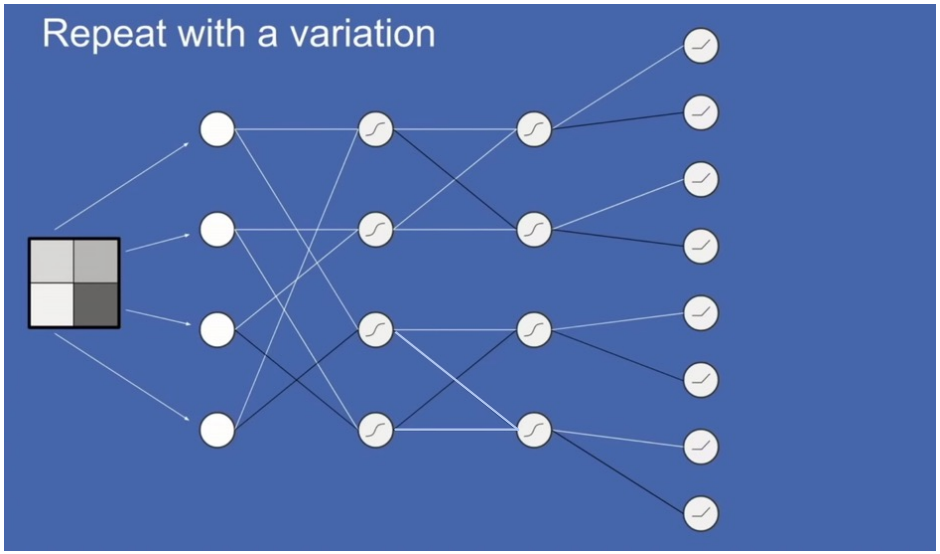


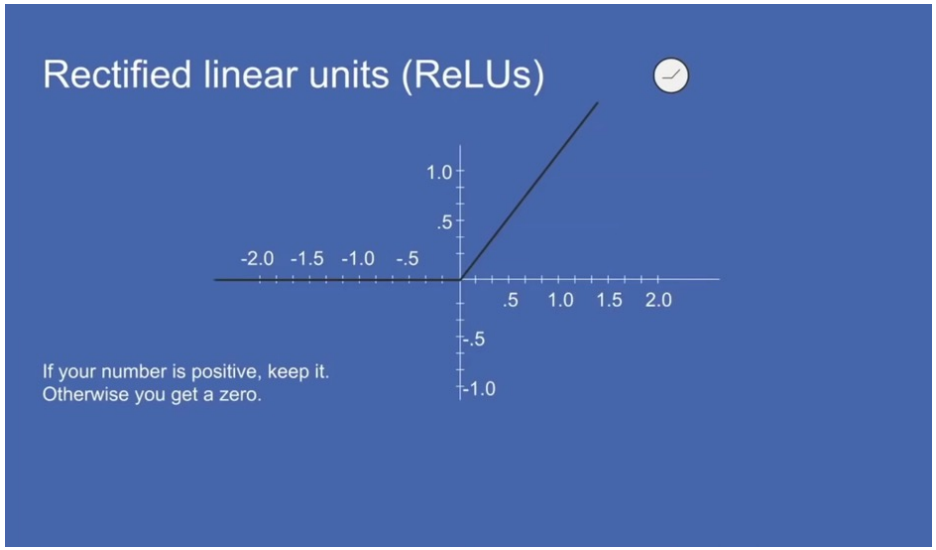


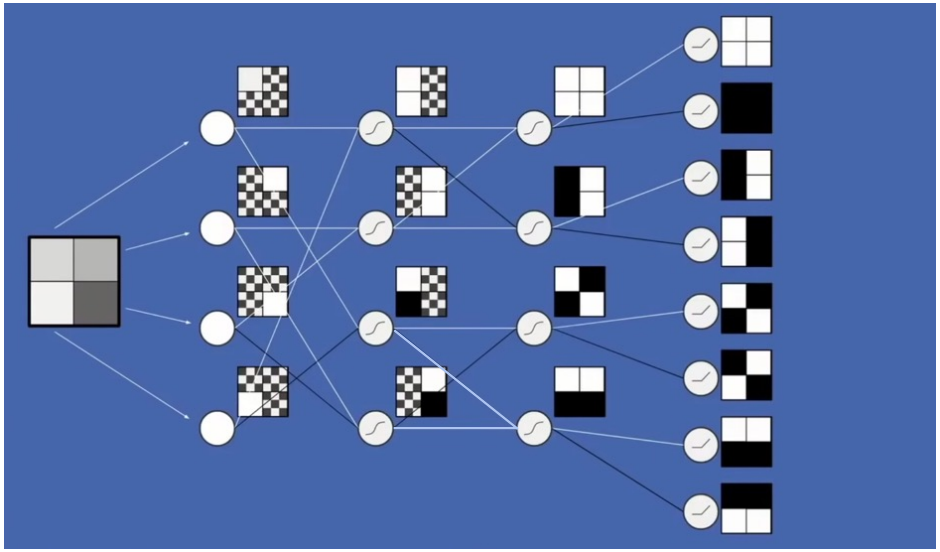


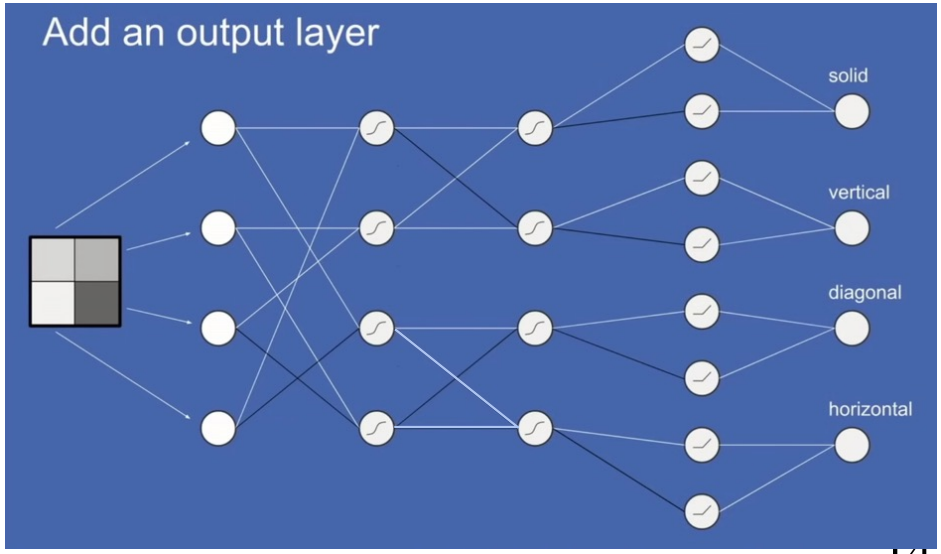
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't





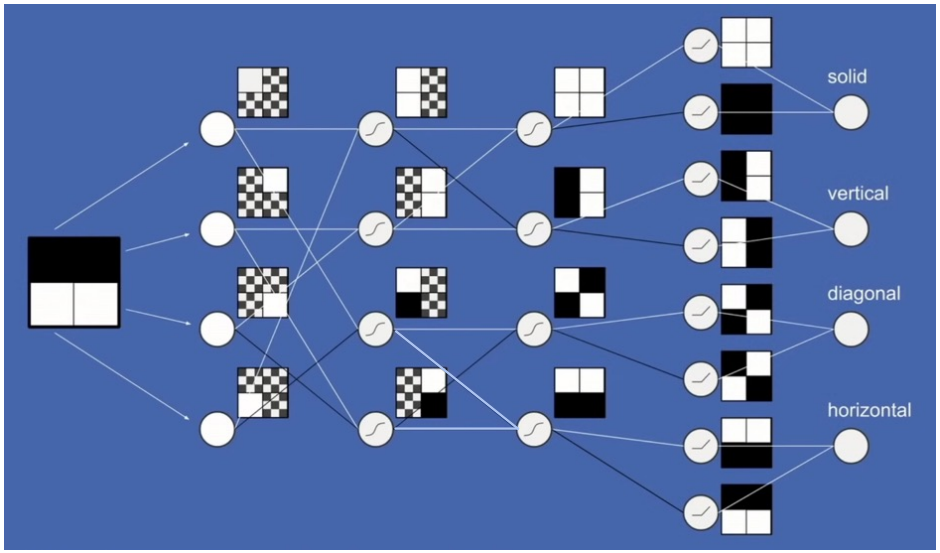


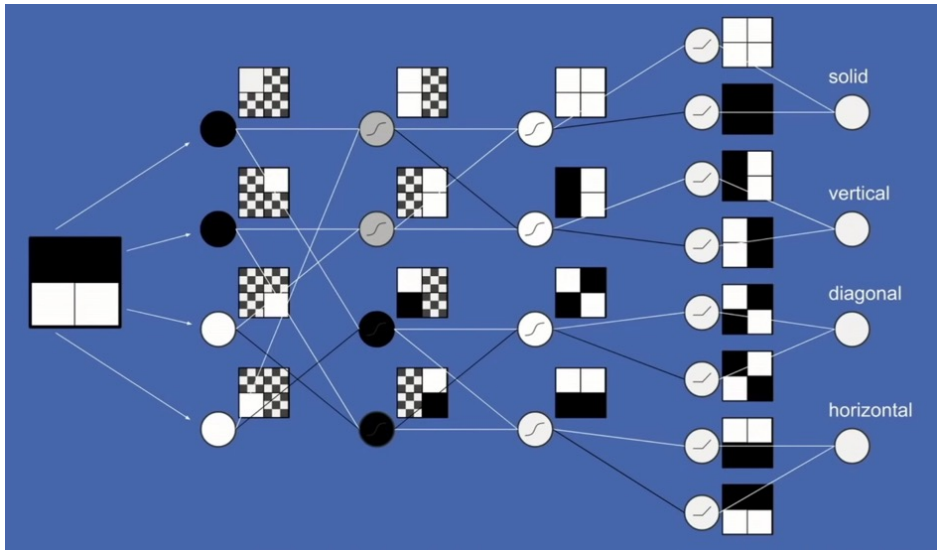


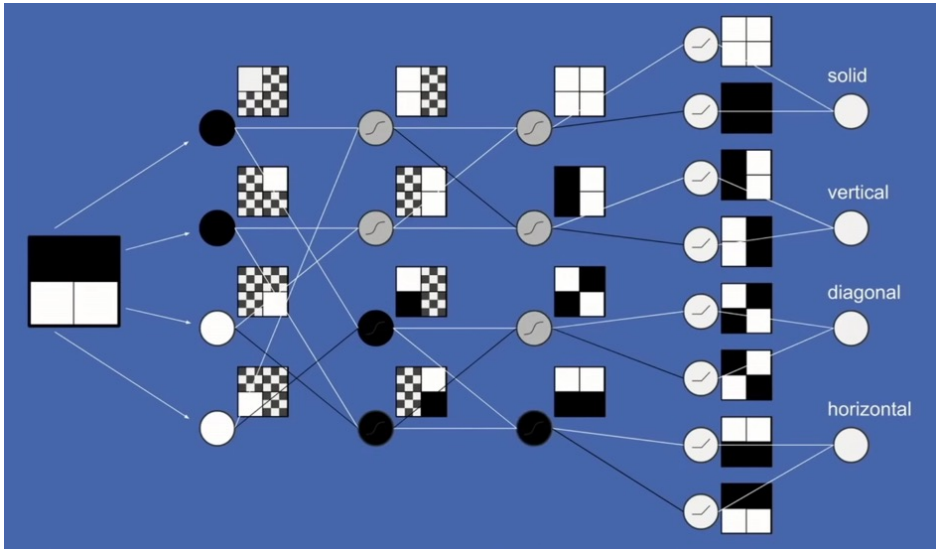
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

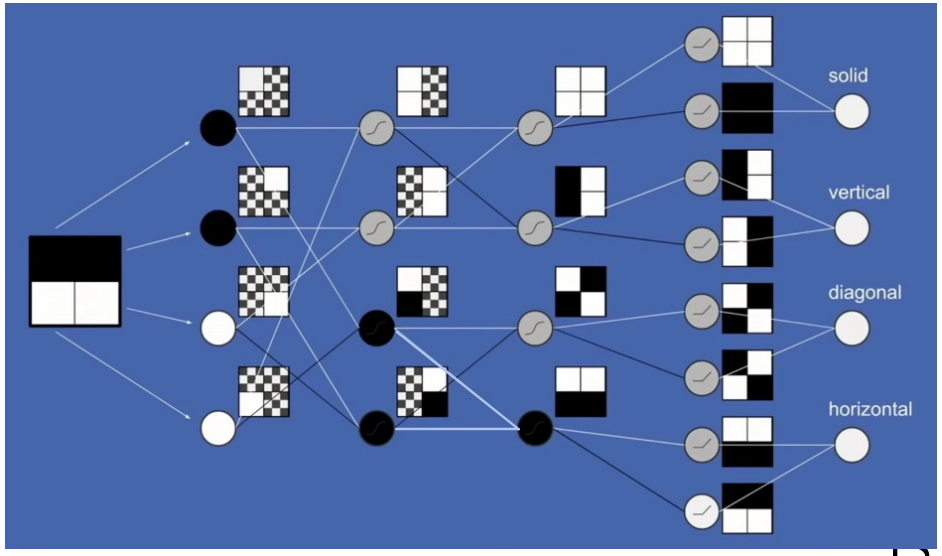
Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Example 1, con't



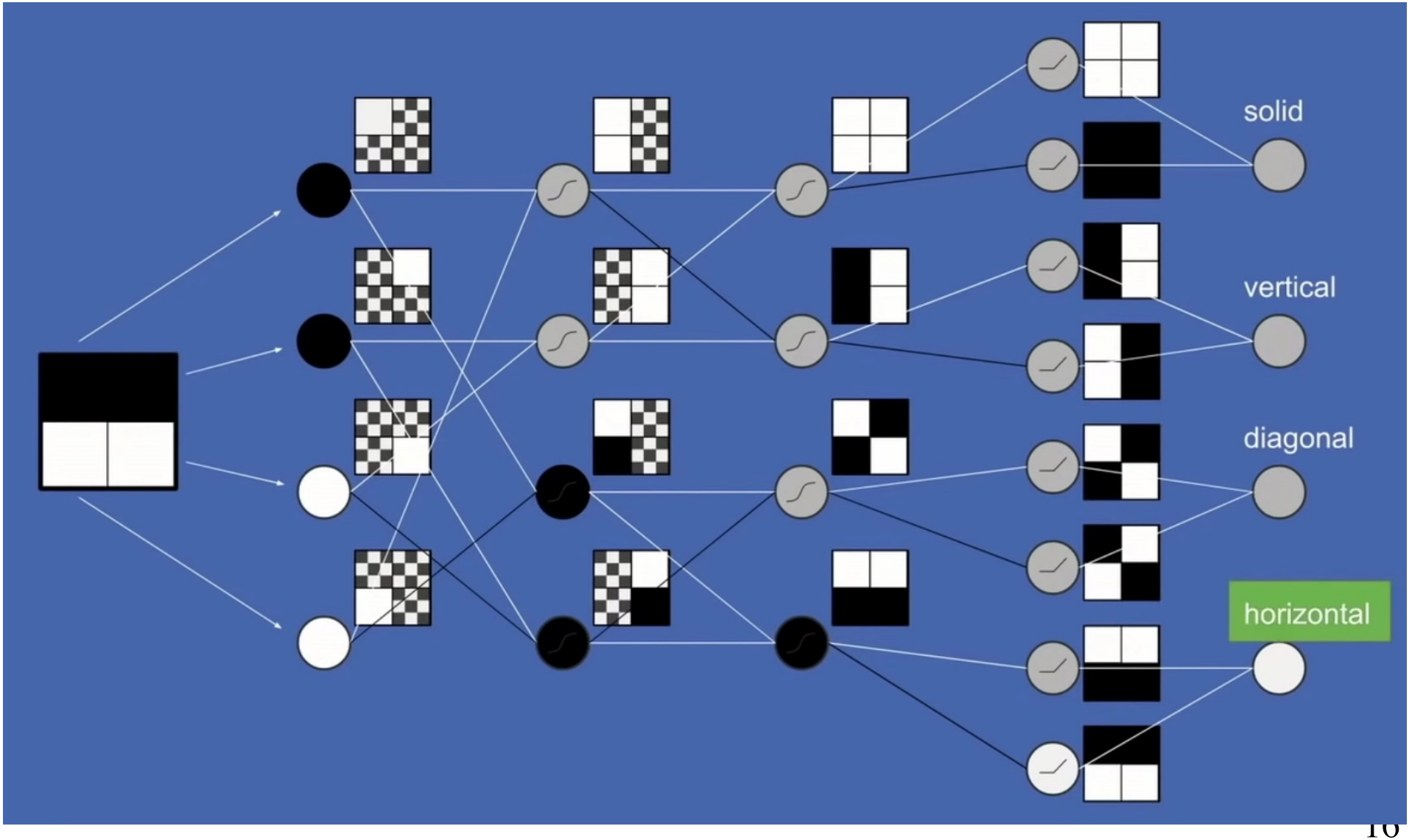






Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

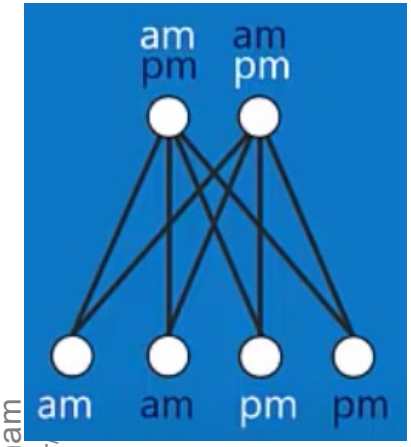
Example 1, finally!

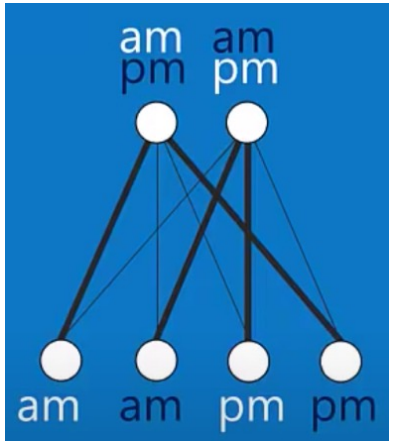


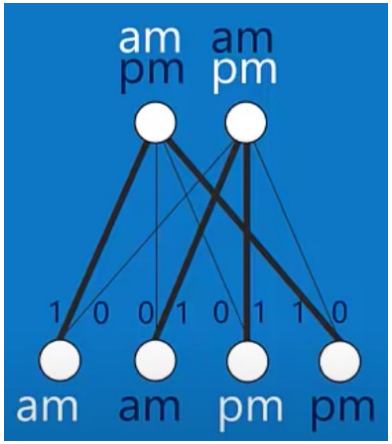
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

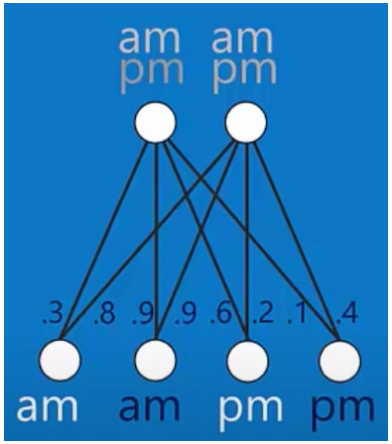
Example 2: Shawarma guy example

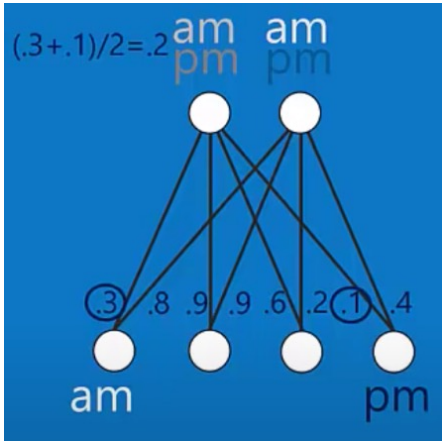
Pictures from https://www.youtube.com/watch?v=Q9Z20HCPnww (Brandon Rohrer)









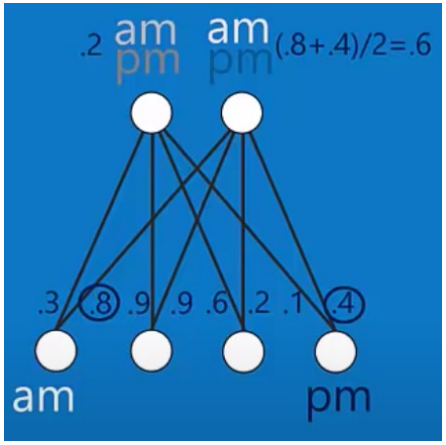


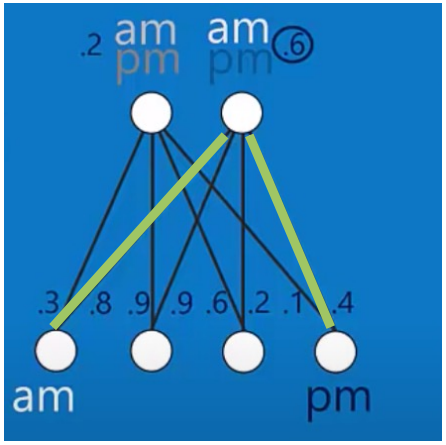
Sometimes he works mornings, sometimes evenings
light=working; dark=off

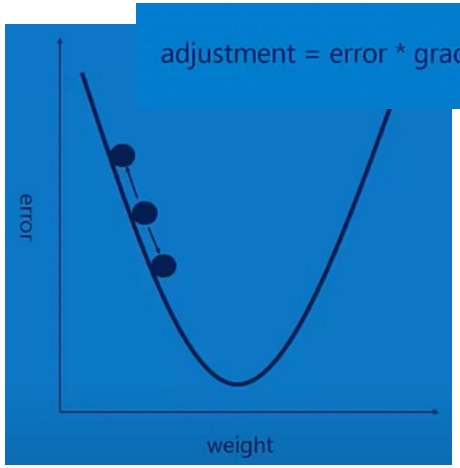
Start with random weights

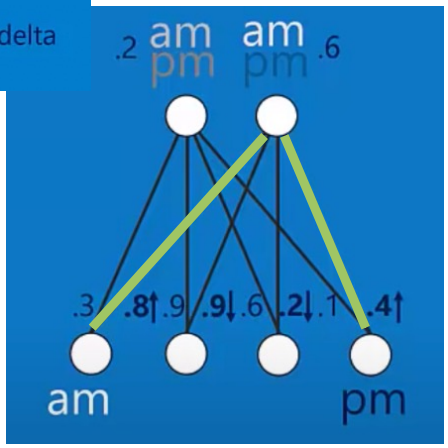
am
pm

Observed on a given day









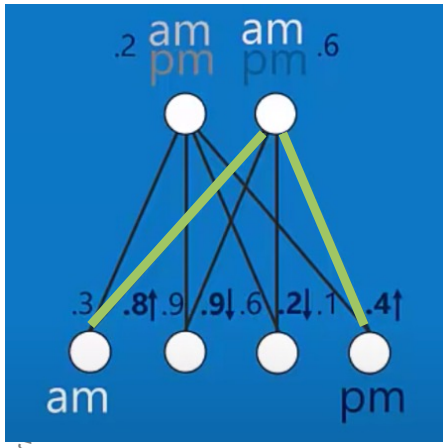
Error=1 - .2 = .8

Error=1 - .6 = .4

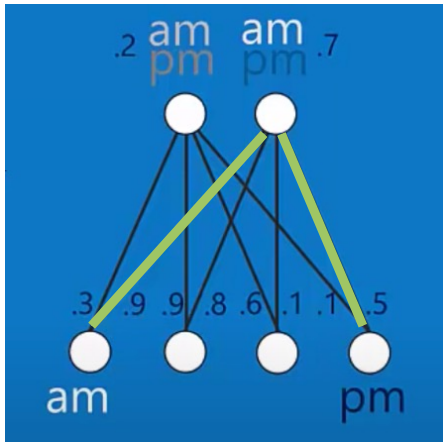
Here comes again our Gradient Descent methods!

Example 2, con't

Pictures from https://www.youtube.com/watch?v=Q9Z20HCPnww (Brandon Rohrer)

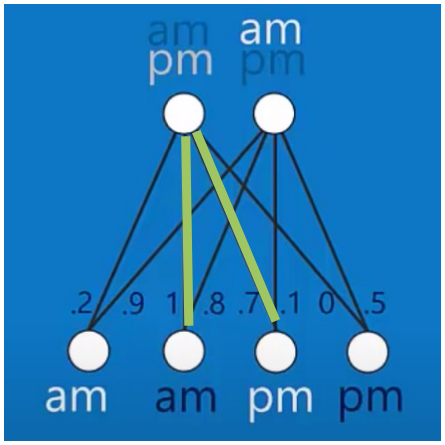


am pm Increase active links, decrease inactive links

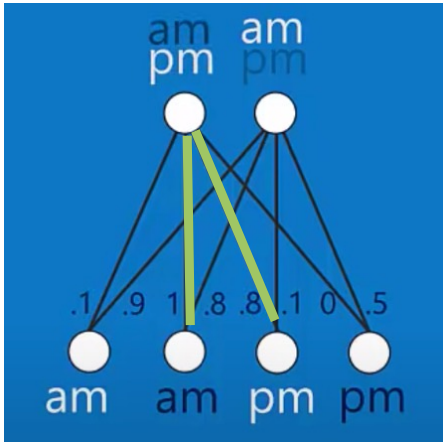


Error=1 - .7 = .3

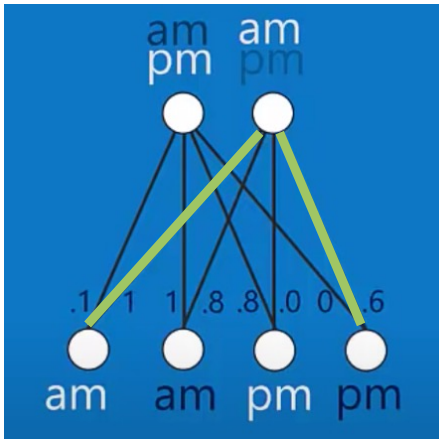
-> better!



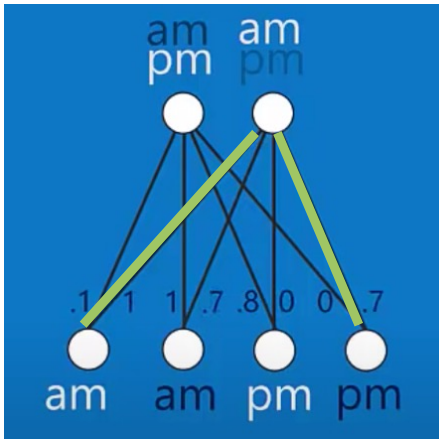
am pm The next day



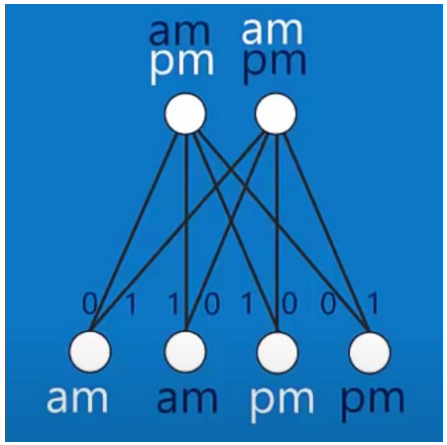
am pm The next day



am pm The next day

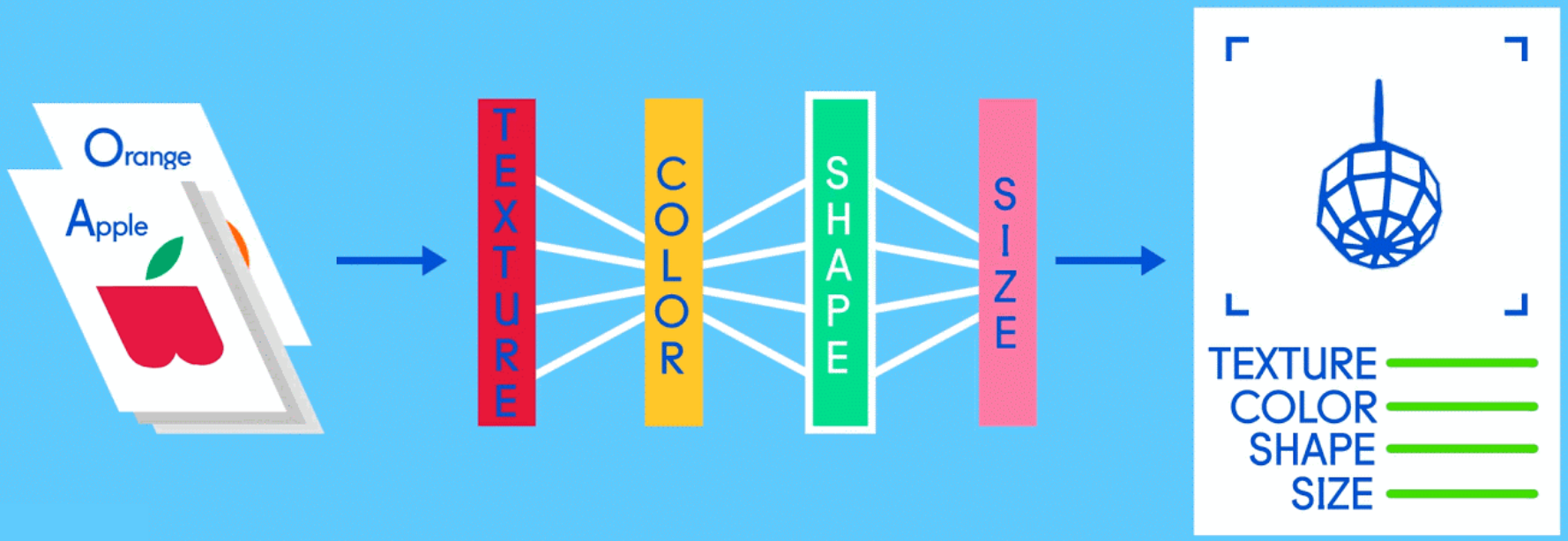


am pm The next day……



Eventually, weights don't change anymore!

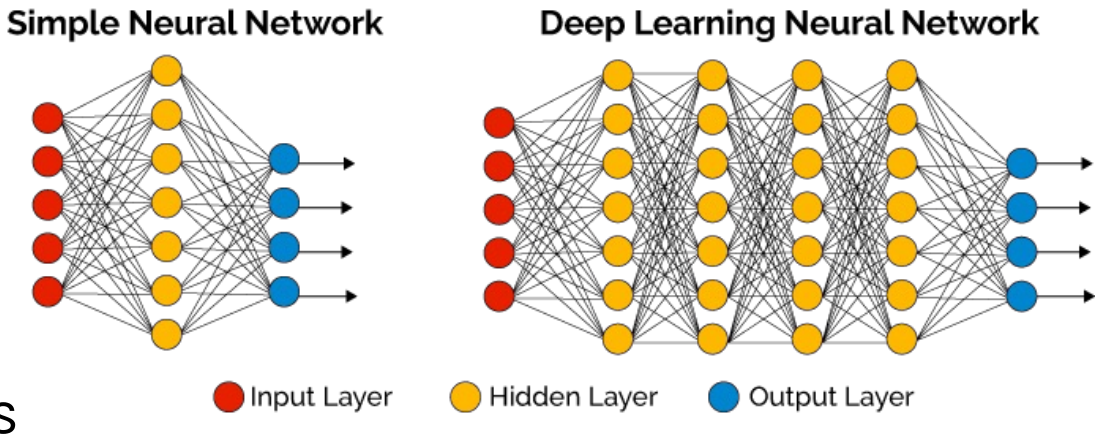
At this point you have a trained model!

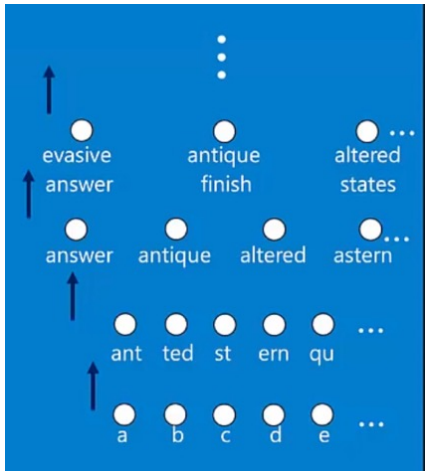


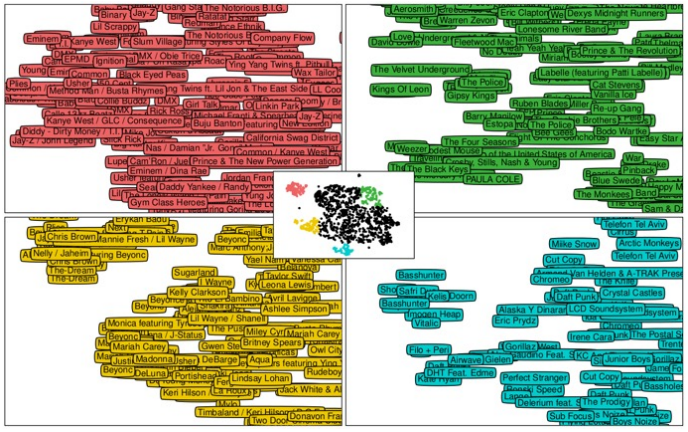
DEEP NEURAL NETWORKS

Deep Neural Networks (DNN)

- Deep Neural Networks are Neural Networks with a large number of hidden layers
- More hidden layers can create/store more internal abstract representation of the training data
- But optimization/learning is more difficult!



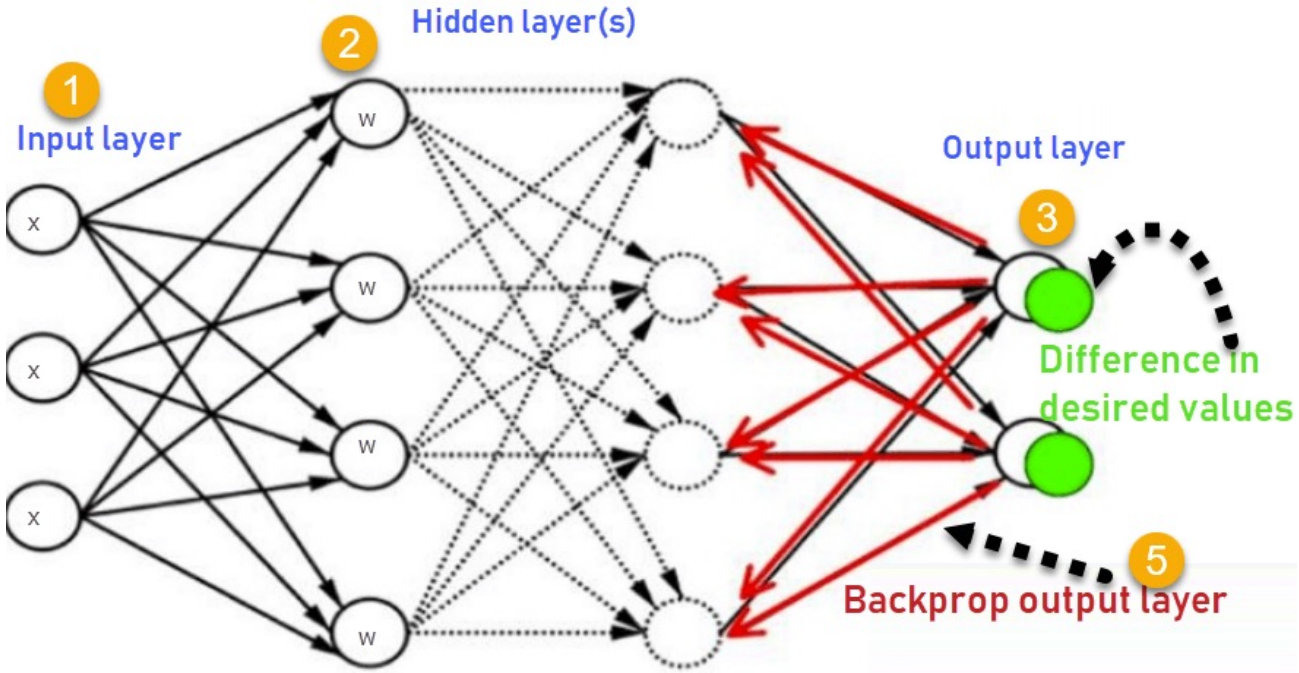




Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Review: Learning in MLP?

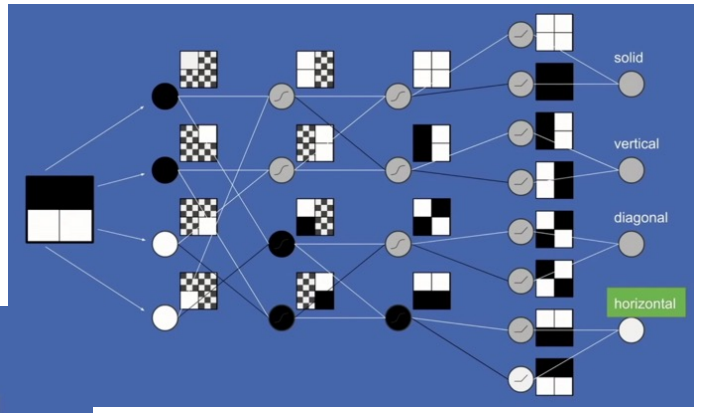
- With more layers, there are more parameters to optimize, so training an MLP is obviously more complex and will take much much more time, if not done in an intelligent way!
- A fundamental method in neural network is back-propagation!

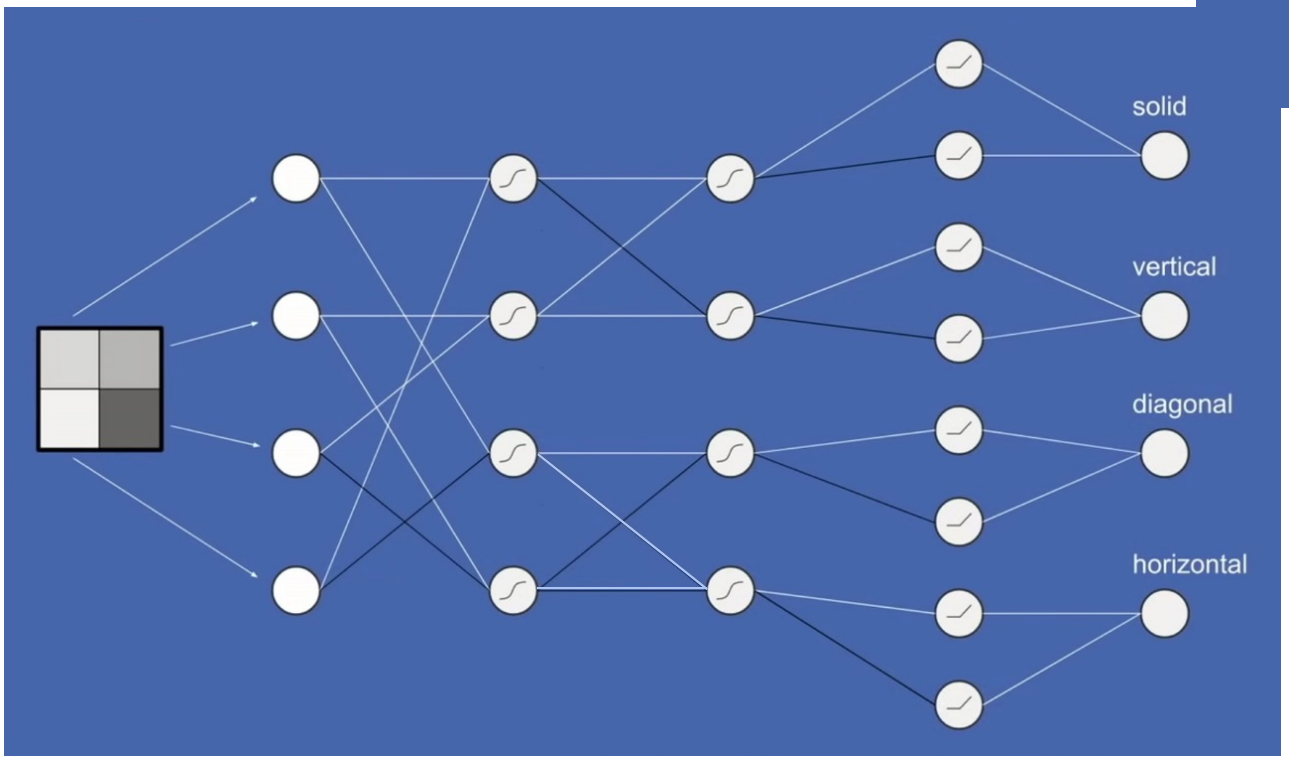


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Look back at the NN of example 1

- In example 1, the NN has already been trained so that the weights have already been assigned accordingly!





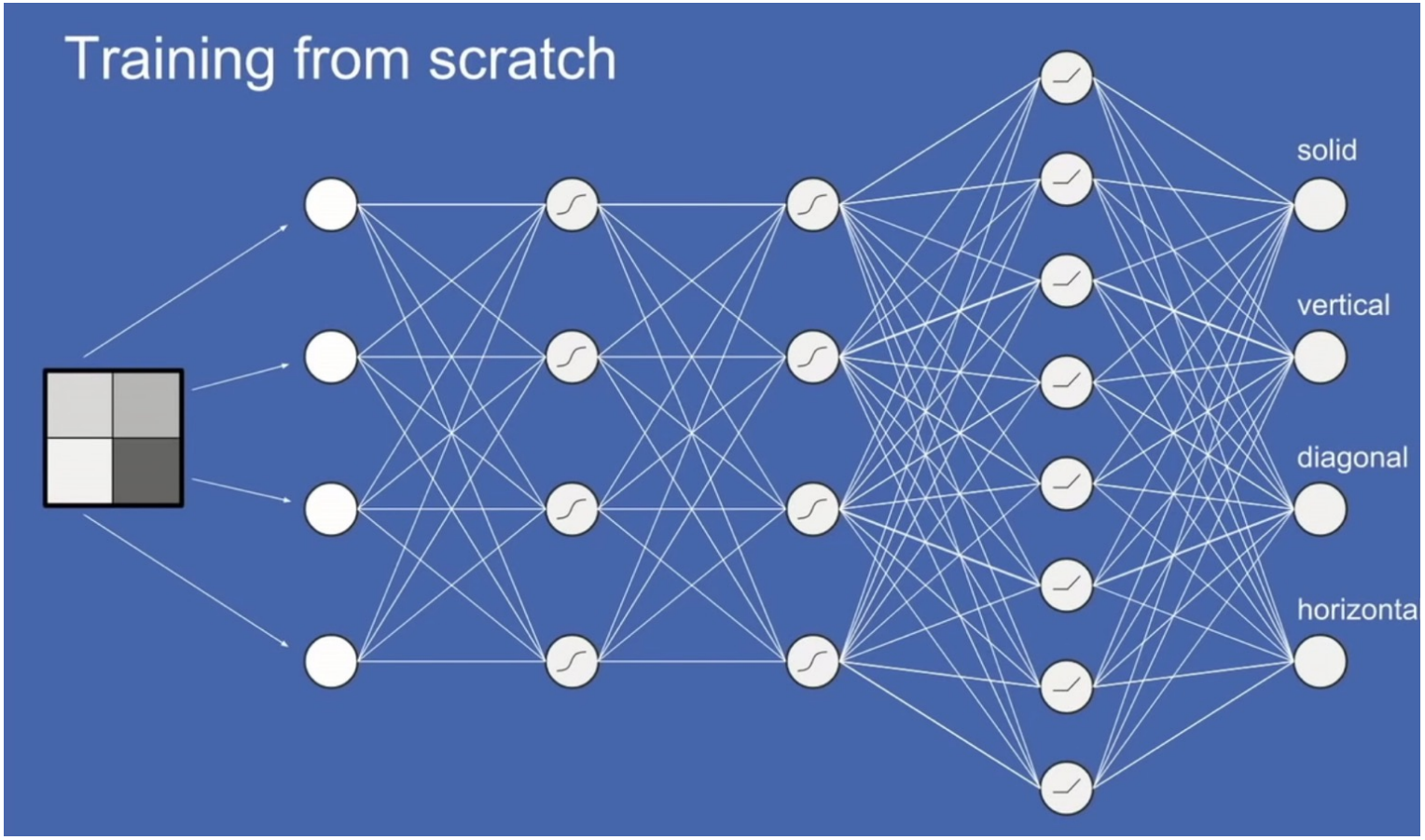
To keep our picture clear, weights will either be
1.0 (white)
-1.0 (black) or
0.0 (missing)

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

But this was the initial NN!

- Fully connected, and weights needed to be trained!

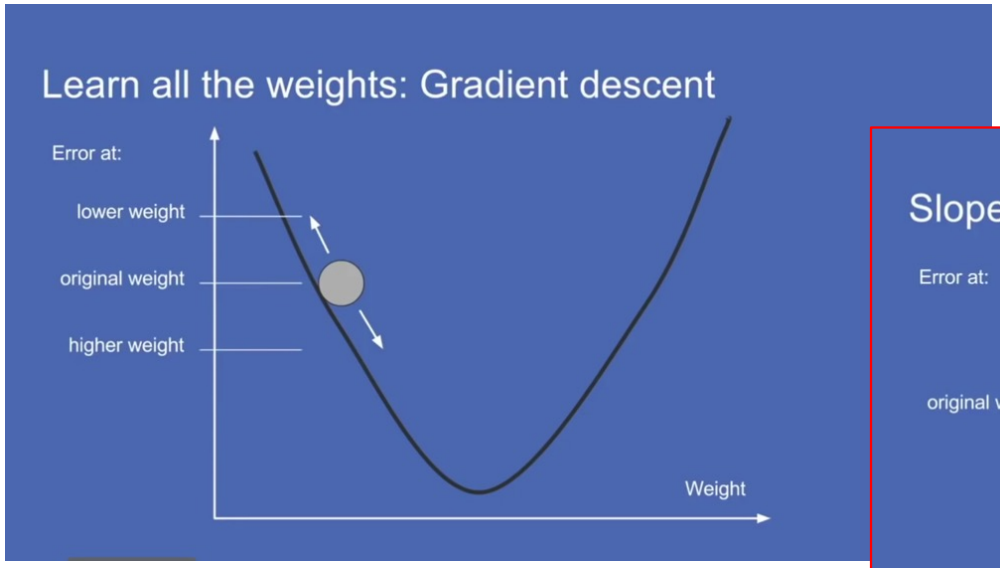


Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

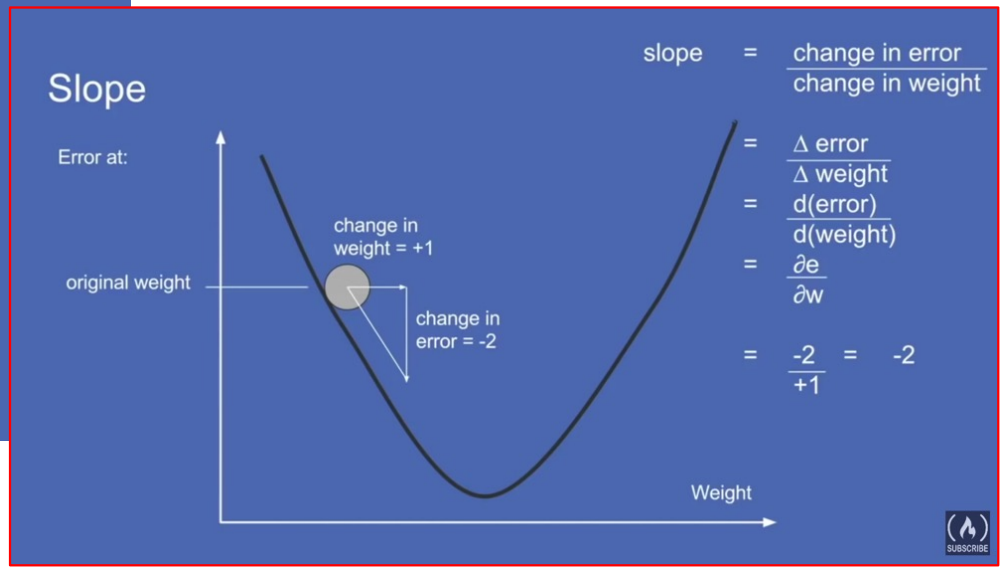
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

How can we train an DNN?

- We saw that Gradient Descent is an efficient way to "smartly" find the optimal weights when trying to minimise the error



What is the change in error with the change in the weights?



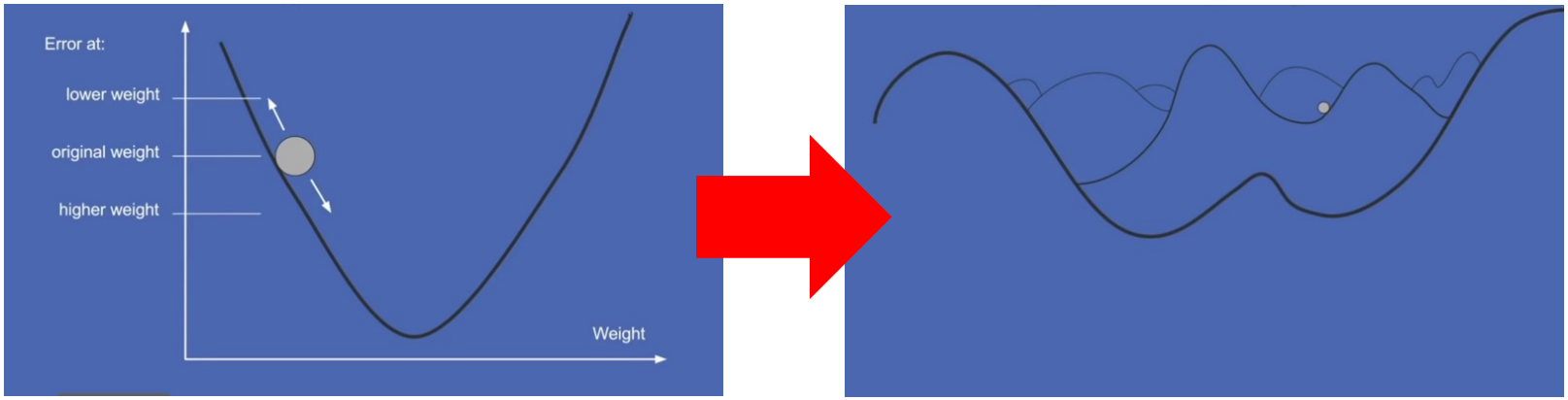
- But in a DNN, we would still need to iterate on all neurons to adjust any single weight in any single hidden layer!

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Which direction is downhill?

- In multi-dimensional problem, the loss function can be quite complex

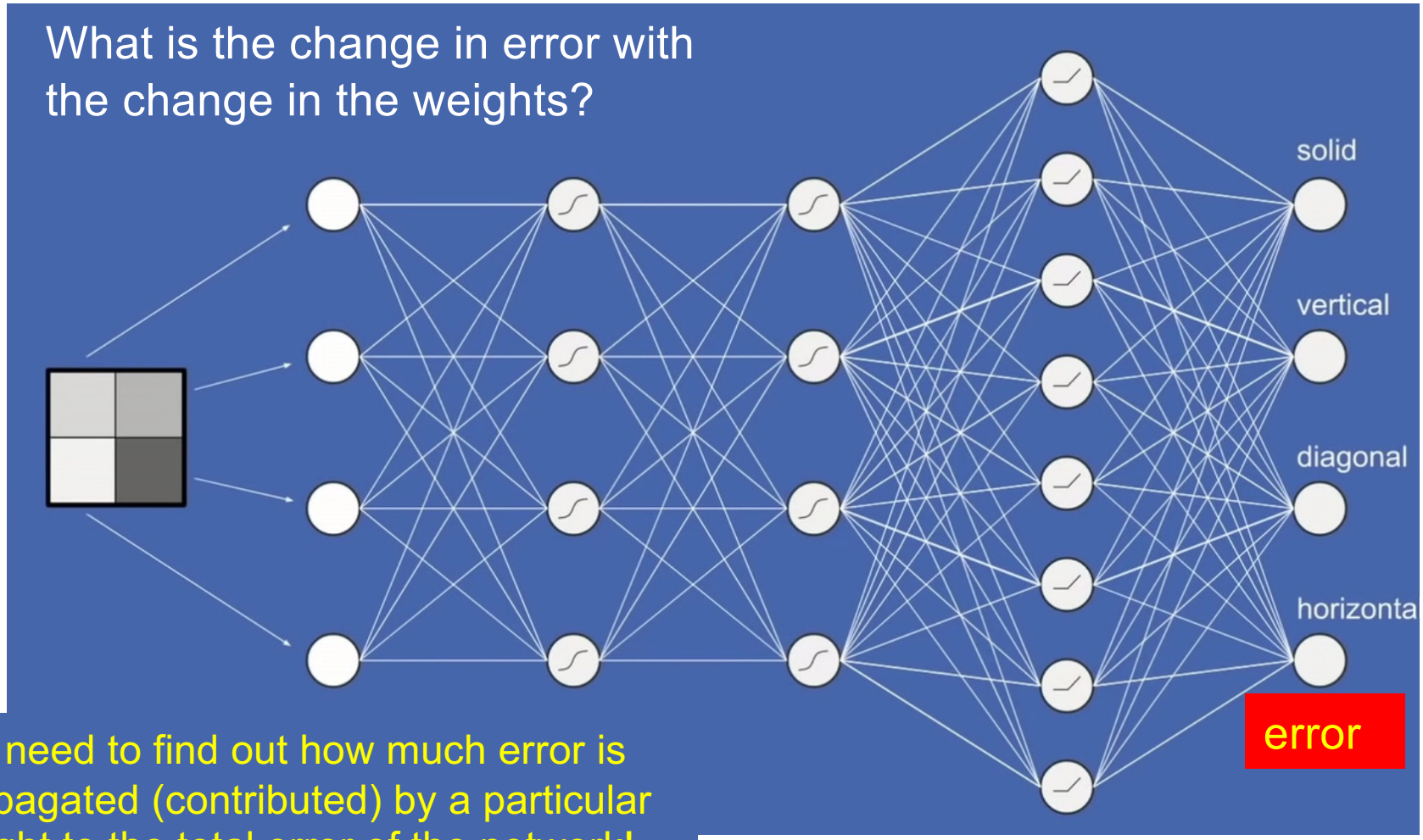


- There can be millions of weights to adjust so calculating the gradient (slope) may require millions of passes through the neural network to find out which direction is downhill!
- Therefore, we need to "help" GD to converge faster!

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

The challenge in adjusting the weights!

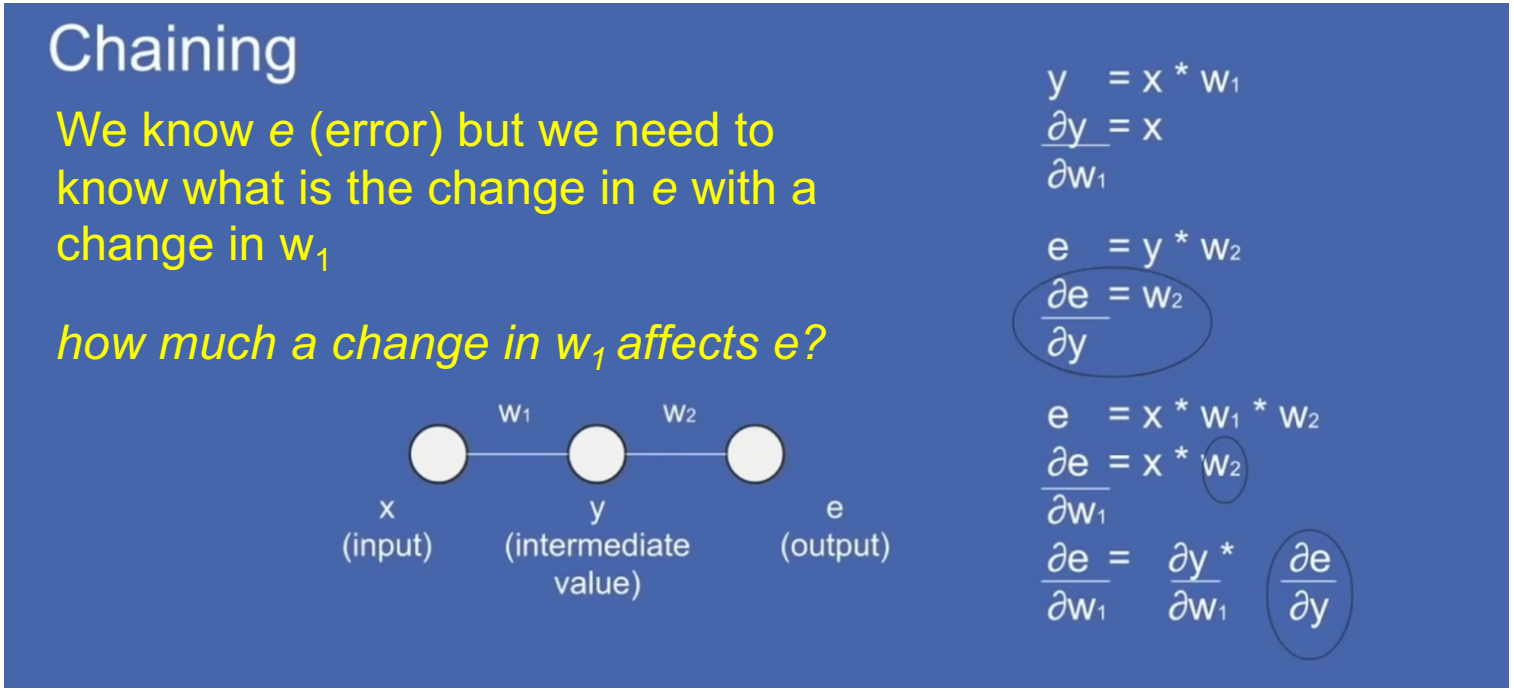


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Chaining principle

- Chaining illustrated on a simple case

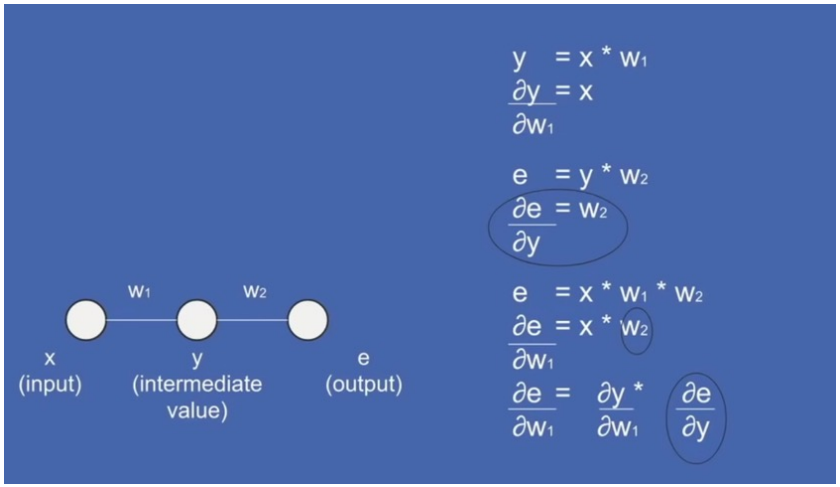


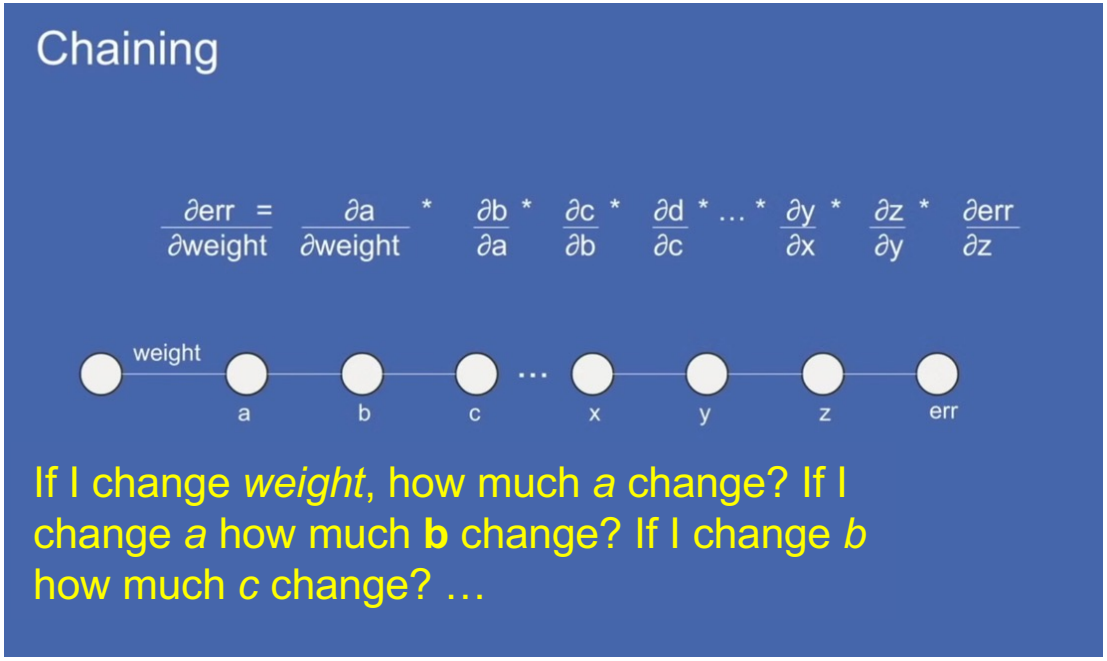
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Chaining

- If we know which way we want to change the error, chaining allows us to calculate how much we change the weights to let that happen





Here *weight* can be anywhere in the NN and we want to know how *err* will change (slope) if *weight* is changed

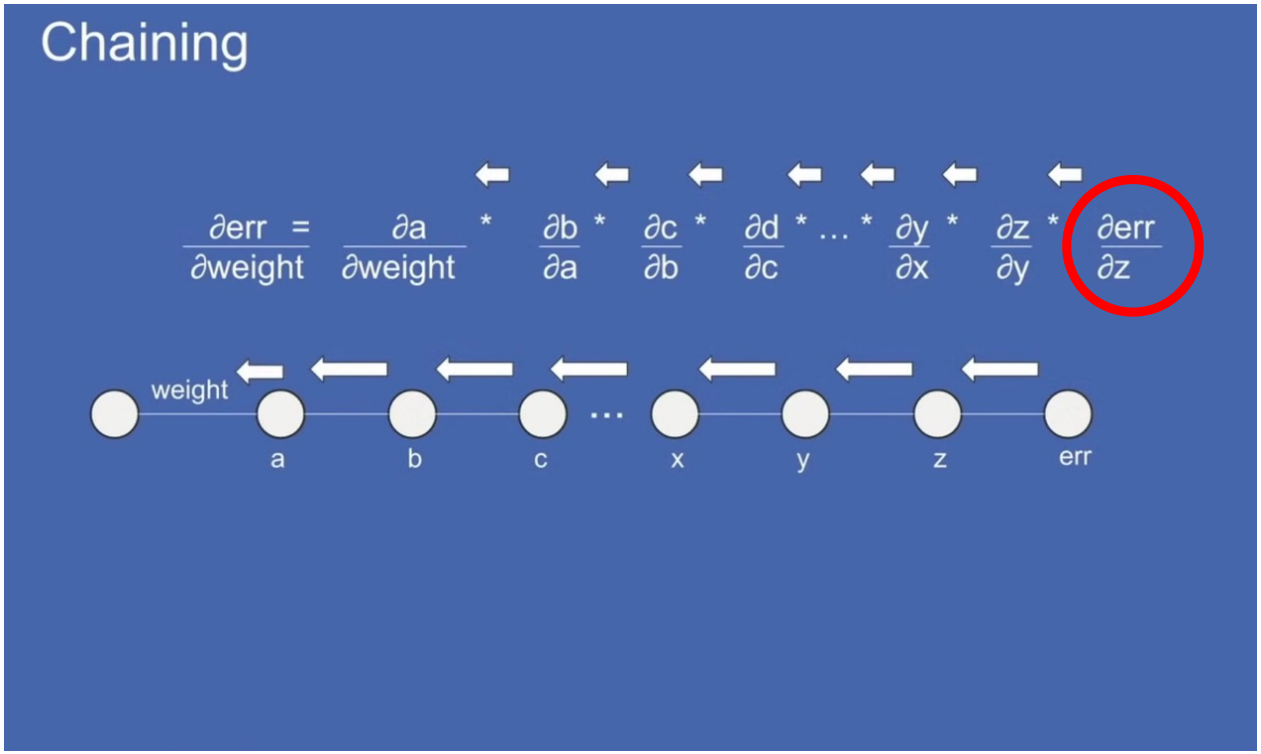
We can apply chaining again and again!

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Backpropagation

- Backpropagation involves taking the error rate of a forward propagation and feeding this loss backward through the neural network layers to fine-tune the weights



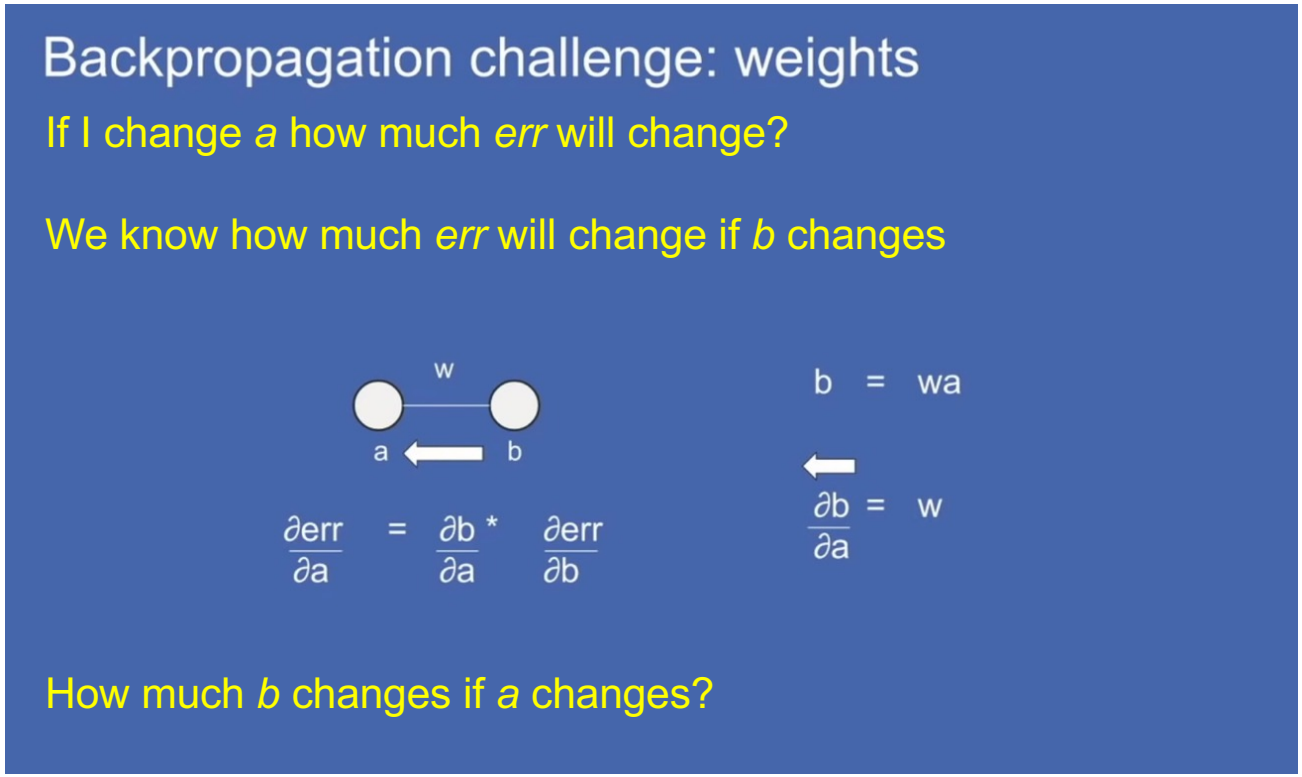
We start with the *err* value at the end

then travel upward, back in the depth of the NN

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Handling weights

- It is easy to handle weights: "How much *b* changes if *a* changes?" is just the weight, whatever it is

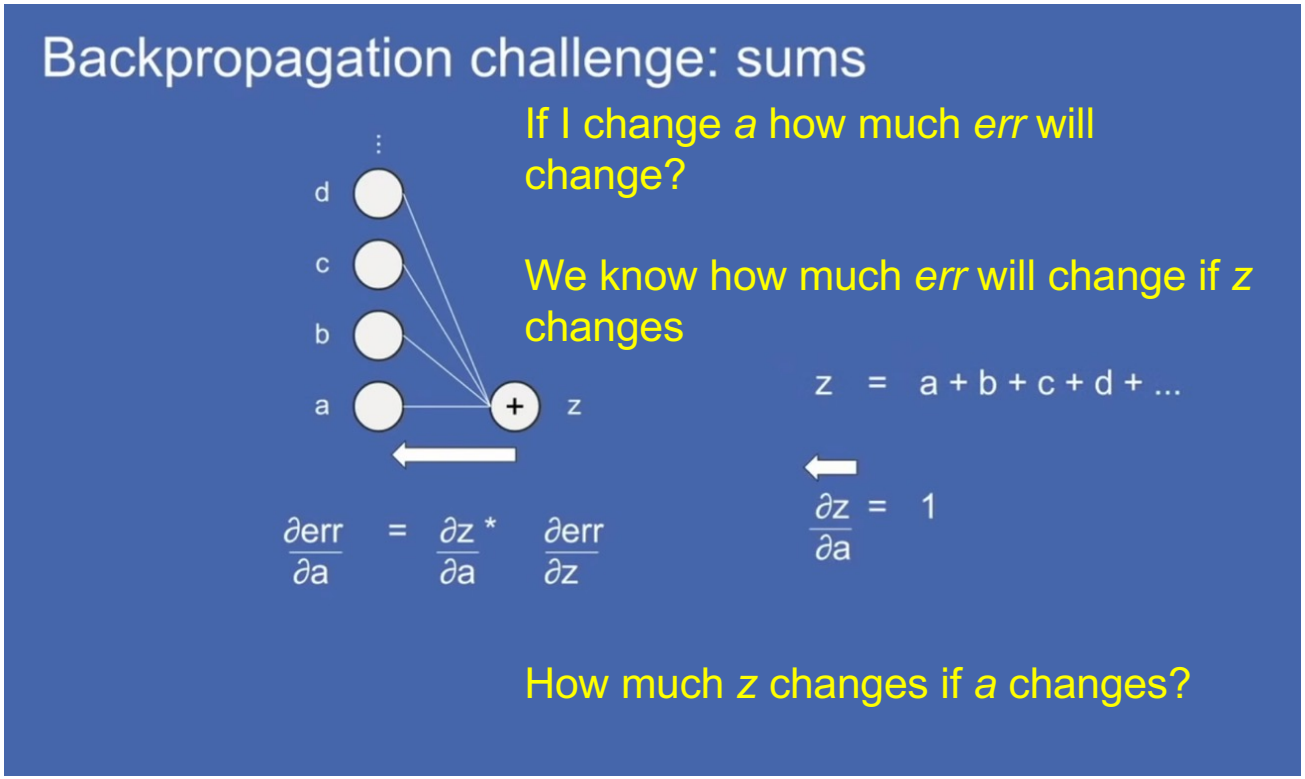


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Handling sums

- Handling sum is also trivial: "How much *z* changes if *a* changes?" is just 1

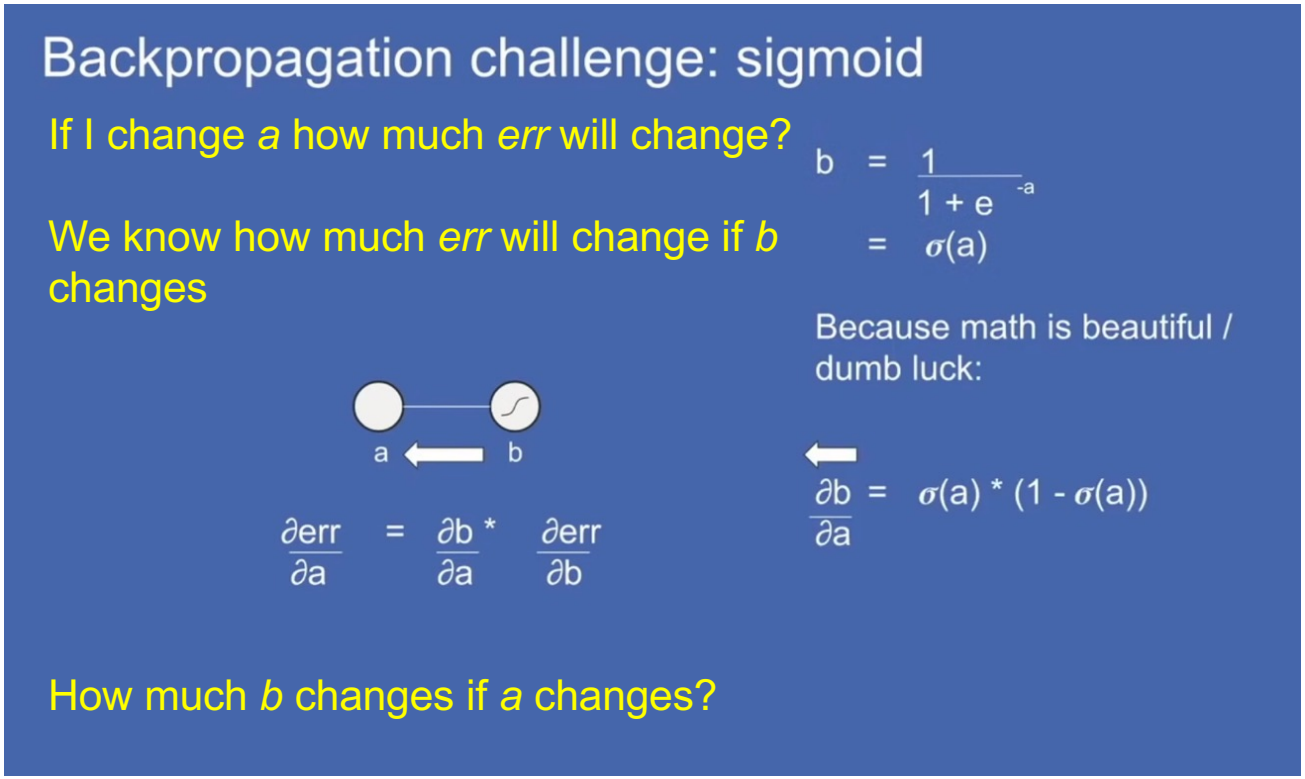


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Handling activation function

- With Logistic function (Sigmoid): "How much *b* changes if *a* changes?" can be known using the derivative of Logistic

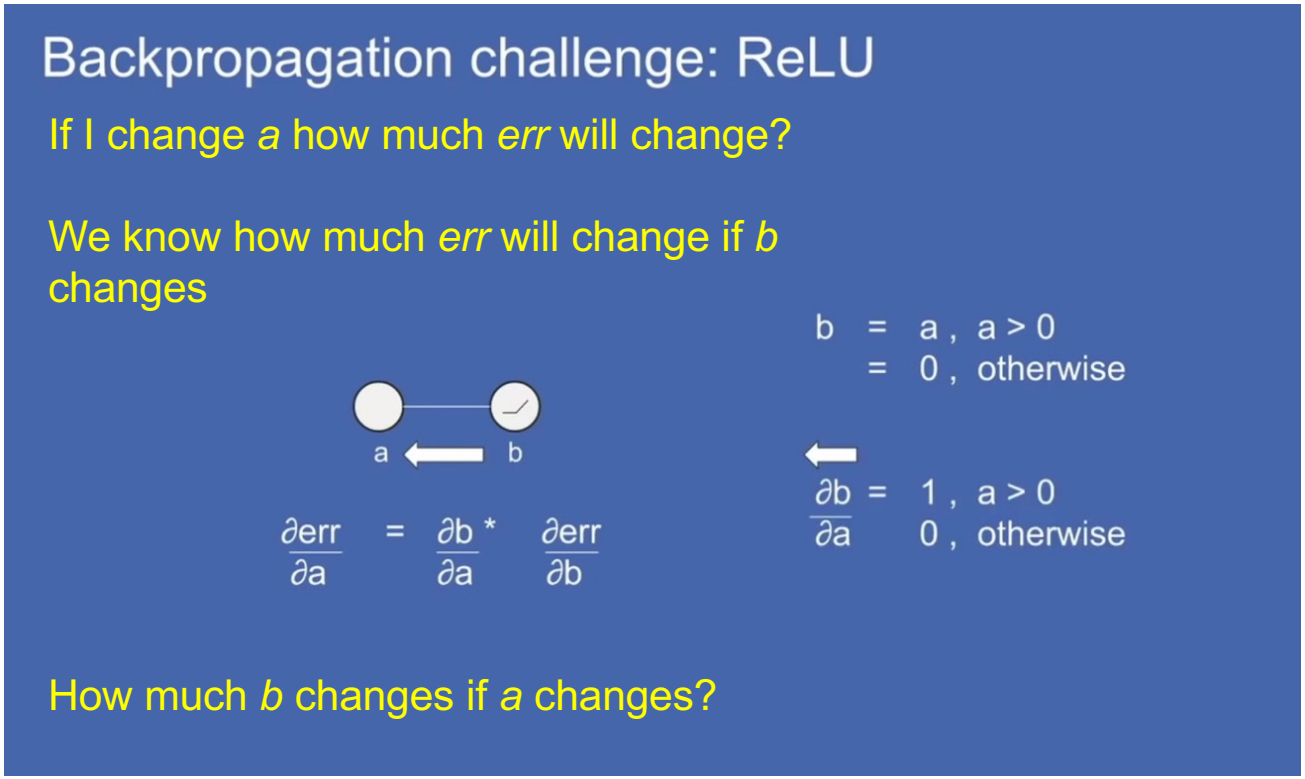


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

Handling activation function

- With ReLU function: "How much *b* changes if *a* changes?" can be known using the derivative of ReLU

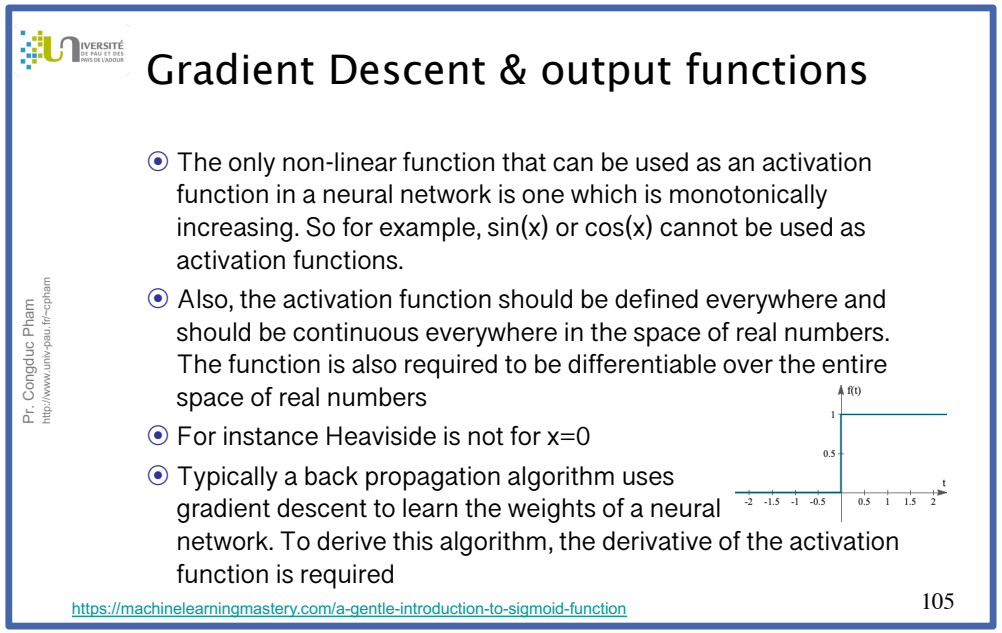


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Pictures from https://www.youtube.com/watch?v=dPWYUELwIdM

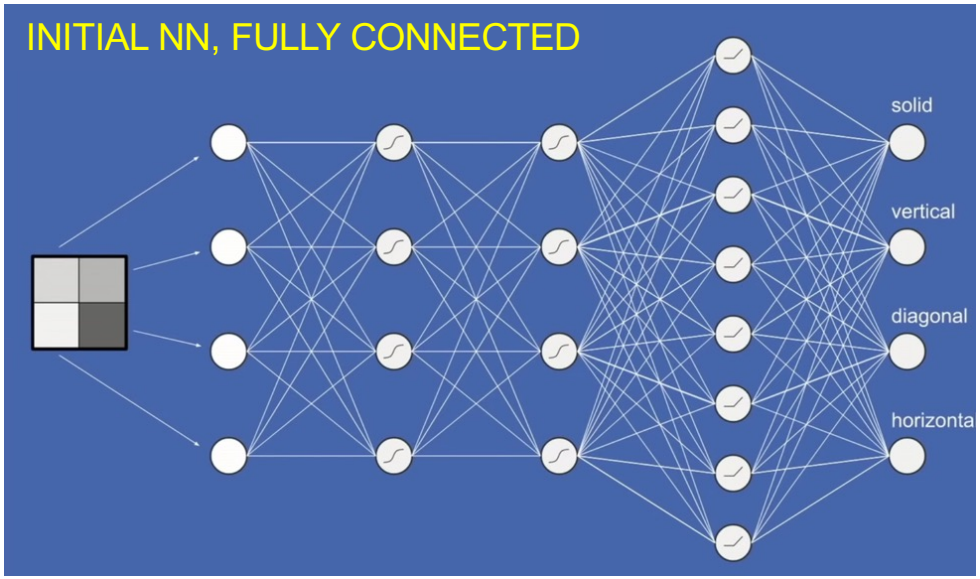
Back (again) to activation function

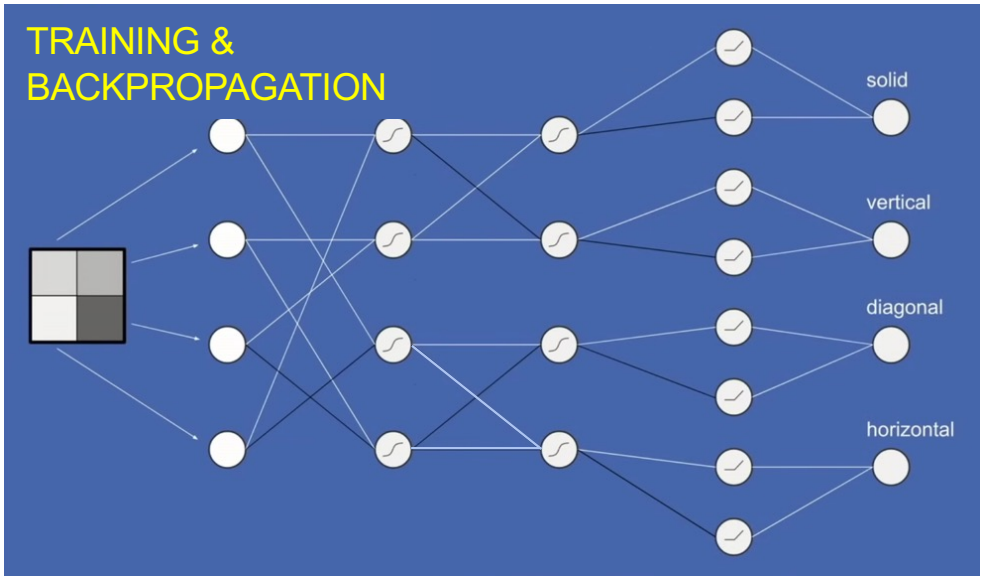
- Remember this slides from part 1?
- "The function is also required to be differentiable over the entire space of real numbers"
- This is needed to make back-propagation works when we are doing the chain rule on the derivatives throughout all the layers of the Neural Network!

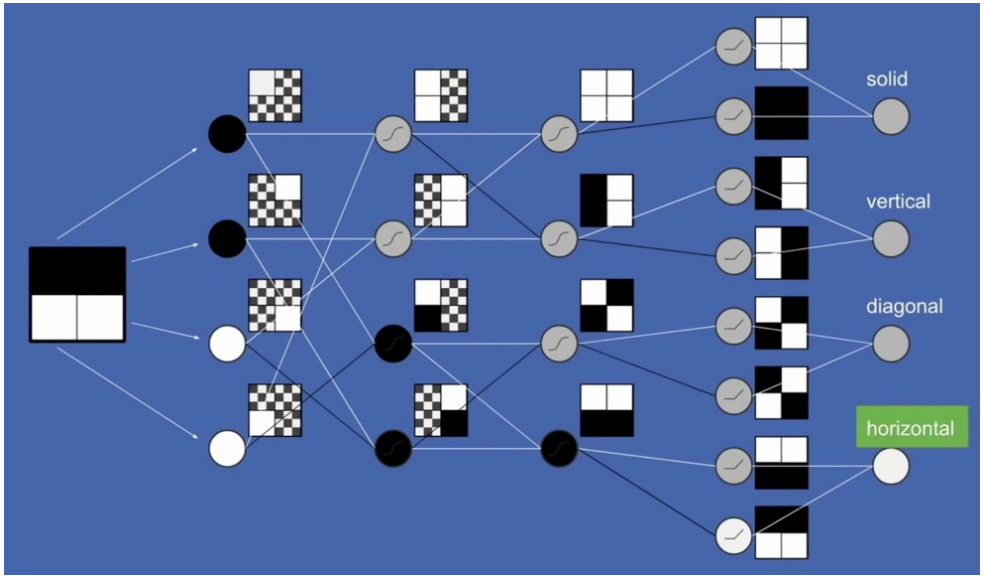


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Putting it altogether



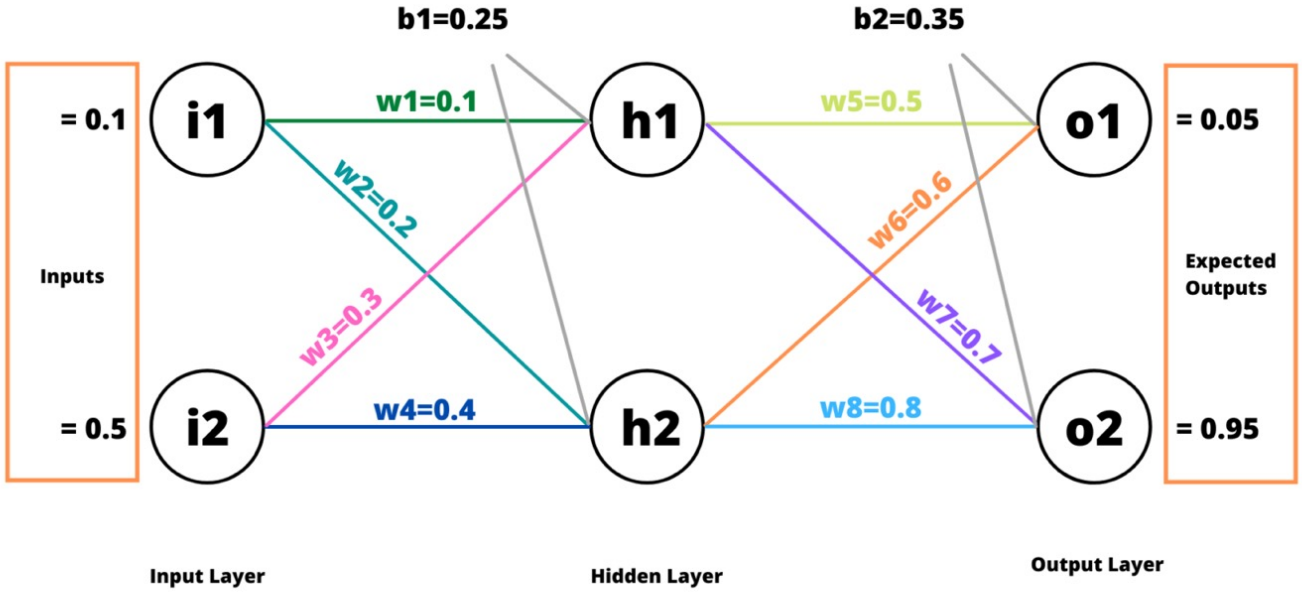




Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

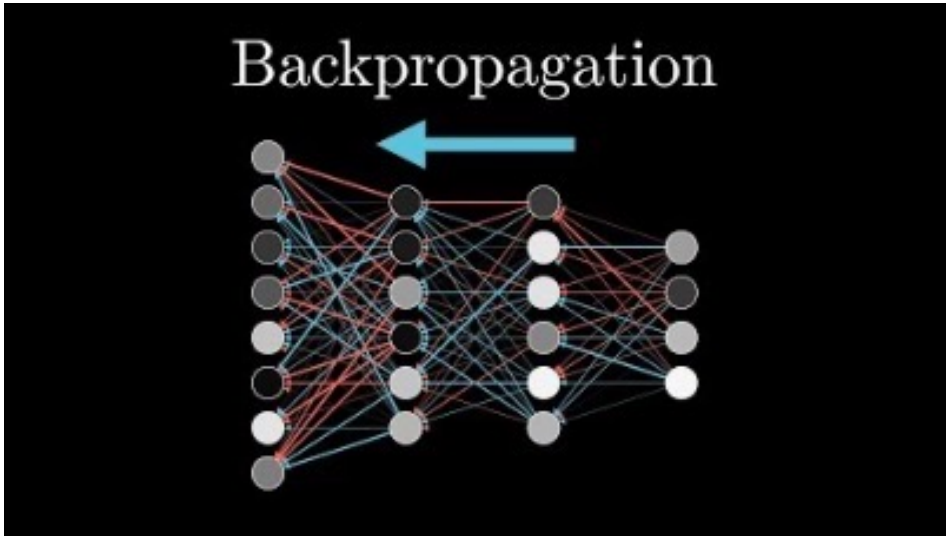
Step-by-step example

- https://theneuralblog.com/forward-pass-backpropagation-example/

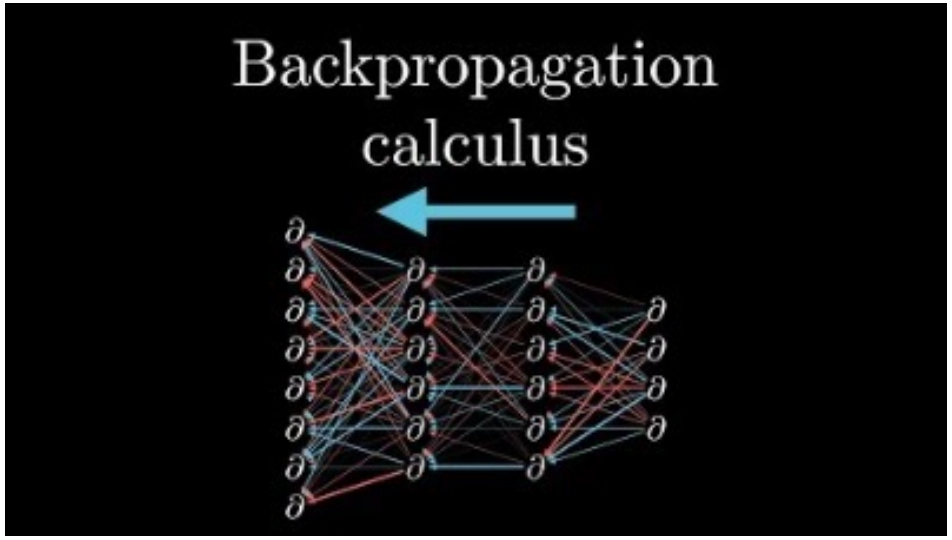


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Watch additional (great) videos



https://www.youtube.com/watch?v=Ilg3gGewQ5U
(3Blue1Brown) What is backpropagation really doing?



https://www.youtube.com/watch?v=tIeHLnjs5U8
(3Blue1Brown) Backpropagation calculus

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

More details on backpropagation

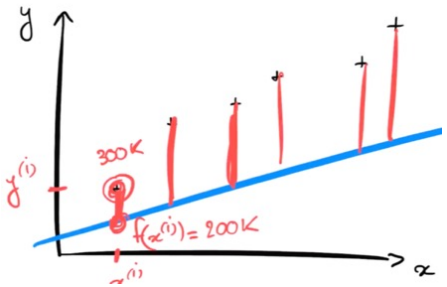
For those who are curious!

- How Does Back-Propagation Work in Neural Networks?
  - https://towardsdatascience.com/how-does-back-propagation-work-in-neural-networks-with-worked-example-bc59dfb97f48
- How to Code a Neural Network with Backpropagation In Python (from scratch) – without any library, just to understand what's under the hood!
  - https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/
- How Back-Propagation Works — A Python Implementation
  - https://towardsdatascience.com/how-back-propagation-works-a-python-implementation-21004d3b47c6

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Back to Loss/Error functions

- Mean Squared Error (MSE) and Mean Absolute Error (MAE) are appropriate loss functions for regression tasks



$$J(a,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(x^{(i)}) - y^{(i)} \right)^2$$

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \hat{y} \right)}^{2}$$

The square of the difference between actual and predicted

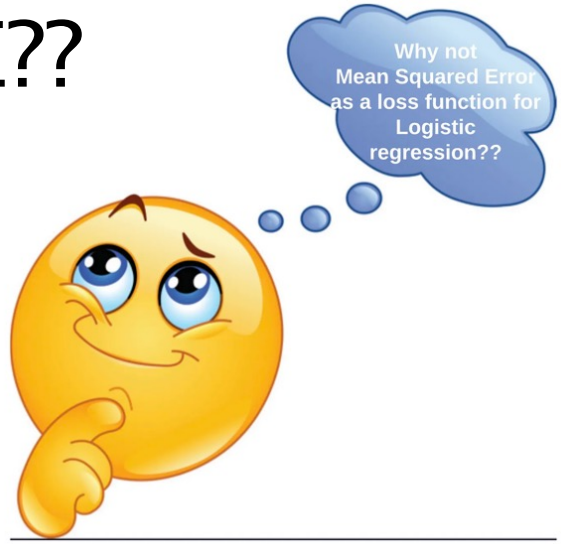
- For classification problems, log loss function (also known as cross-entropy) is more suitable with Sigmoid activation function

$$-(y \log(p) + (1 - y) \log(1 - p))$$

Equation of Log loss function. **y**-Actual output, **p**-probability predicted by the logistic regression

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Why Log Loss instead of MSE??

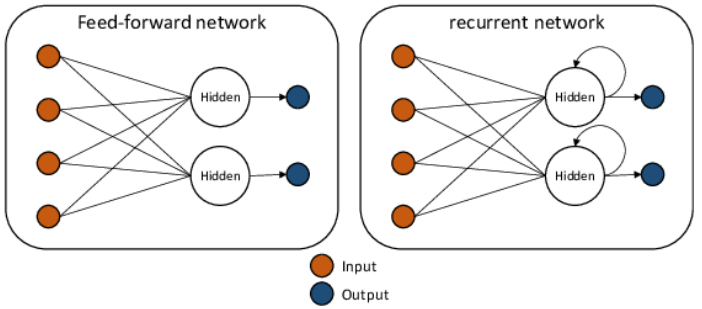


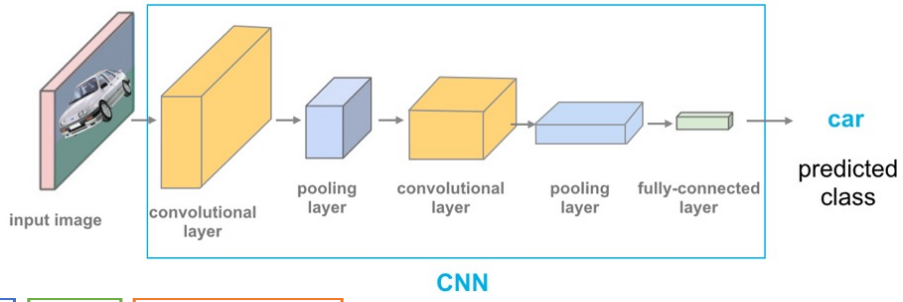
- https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c
- First, the target value is either 0/1 in classification problems, so $\left(y - \hat{y}\right)$ will always be between 0–1, making it very difficult to track the progress of error value
- Second, and the main reason actually, MSE doesn't work well with logistic regression because when the MSE loss function is plotted with respect to weights of the logistic regression model, the curve obtained is not a convex curve which makes it very difficult to find the global minimum

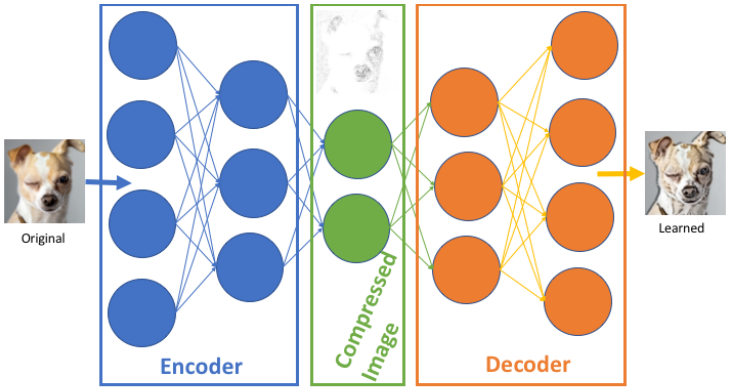
From https://medium.com/analytics-vidhya/understanding-the-loss-function-of-logistic-regression-ac1eec2838ce

Types of Deep Neural Networks

- Feedforward Neural Networks (**FFNs, ANNs** or **NNs**)
- Recurrent Neural Networks (**RNNs**)
- Convolutional Neural Networks (**CNNs**)
- Autoencoder Neural Networks (**AEs**)



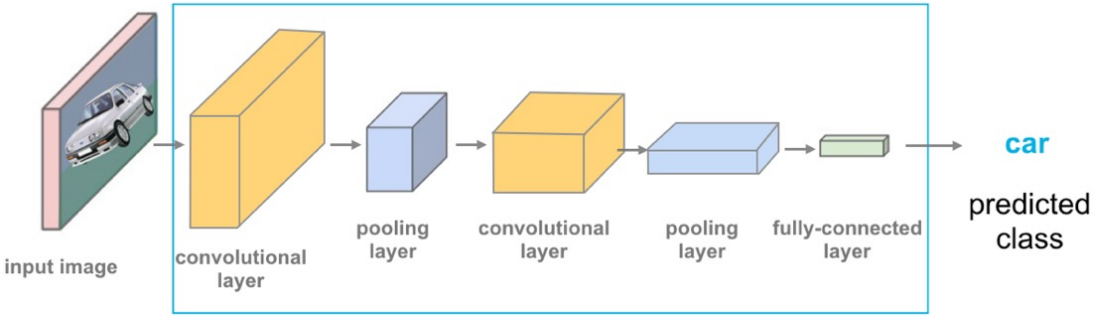


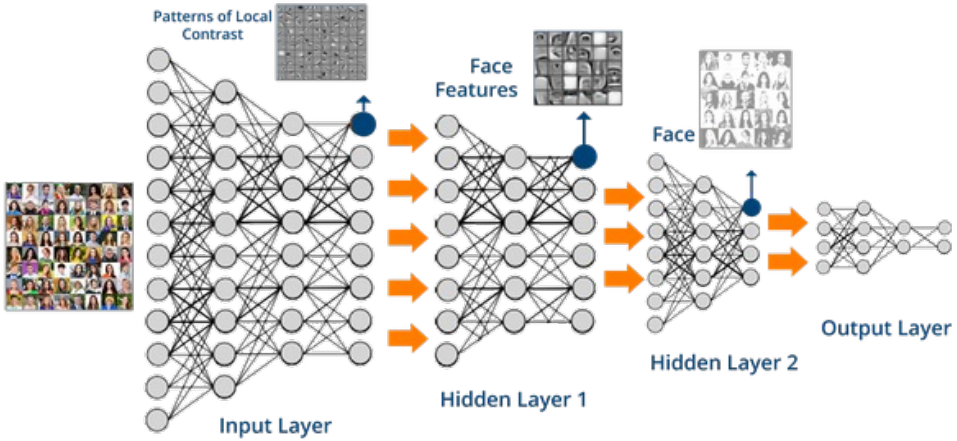


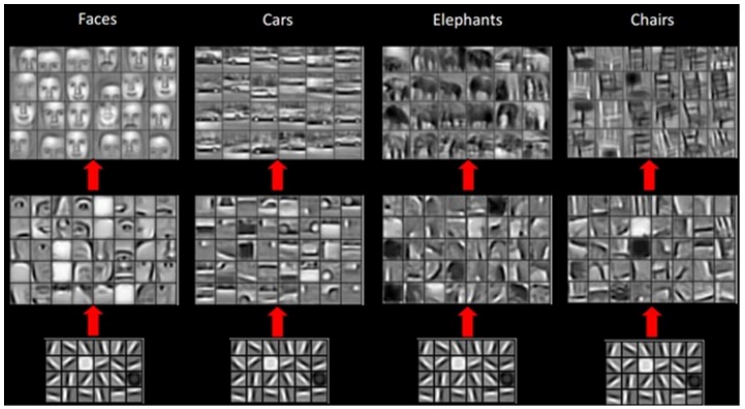
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

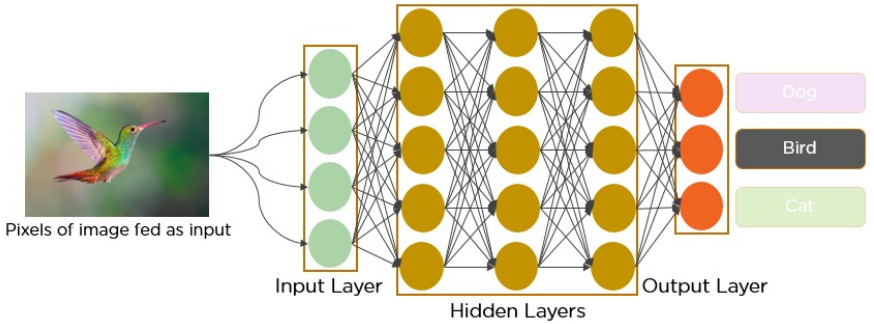
Convolutional Neural Networks

- Contain five types of layers
- Each layer has a specific purpose, like summarizing, connecting or activating
- CNN are good at image classification and object detection





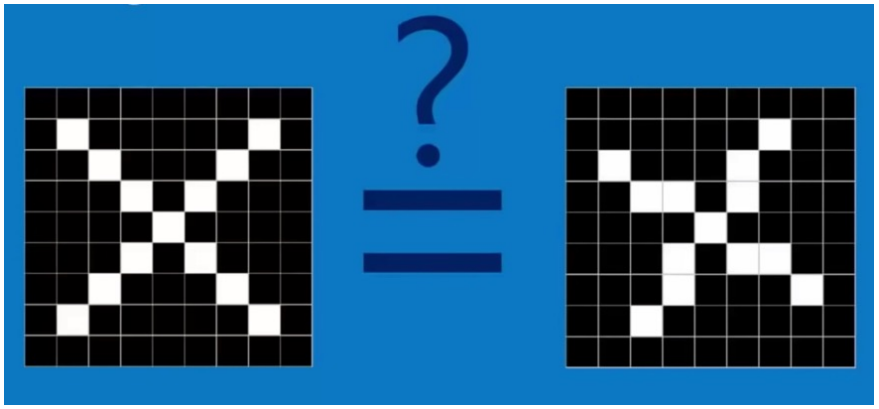


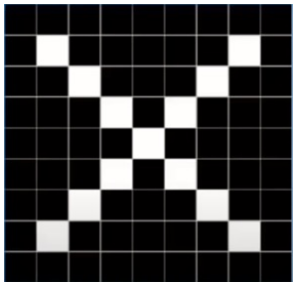


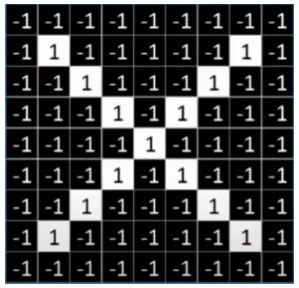
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

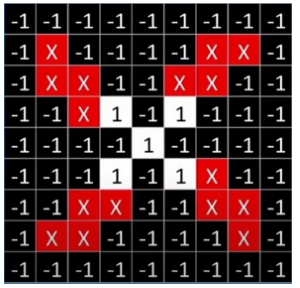
https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvI8E7ogmIBnFNBUY70&index=5&t=3097s

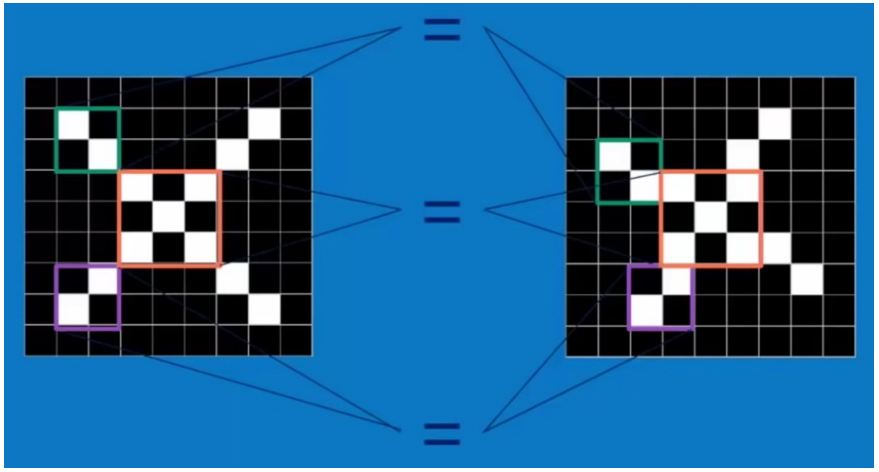
What is a convolution?

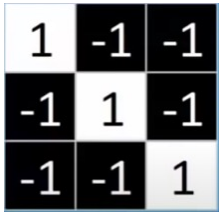


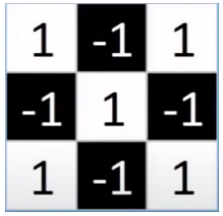


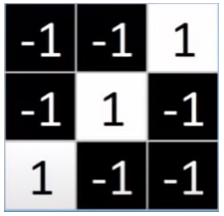


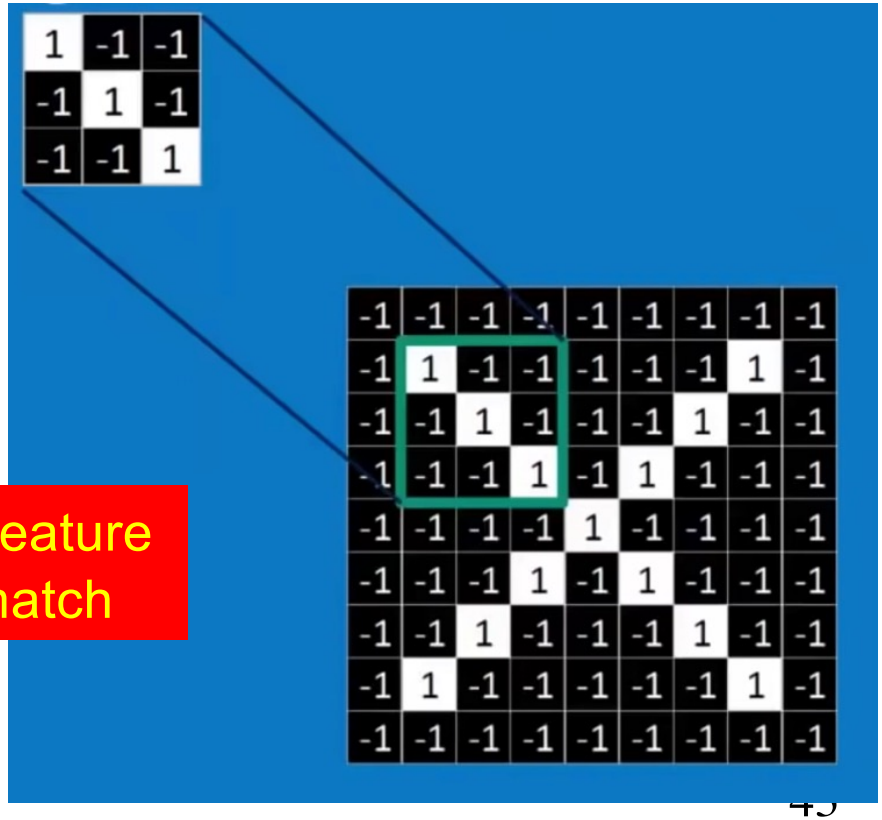








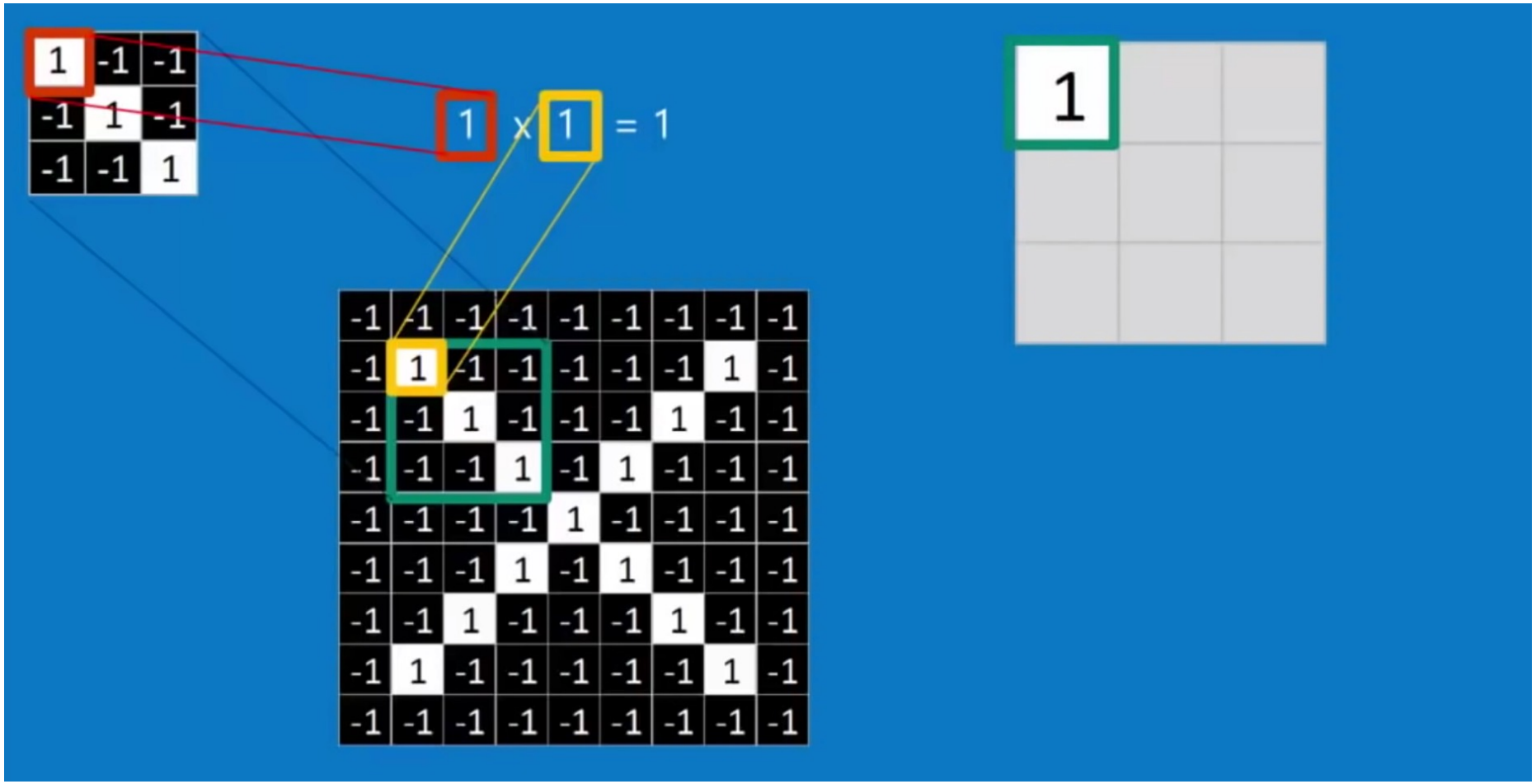




Feature match

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

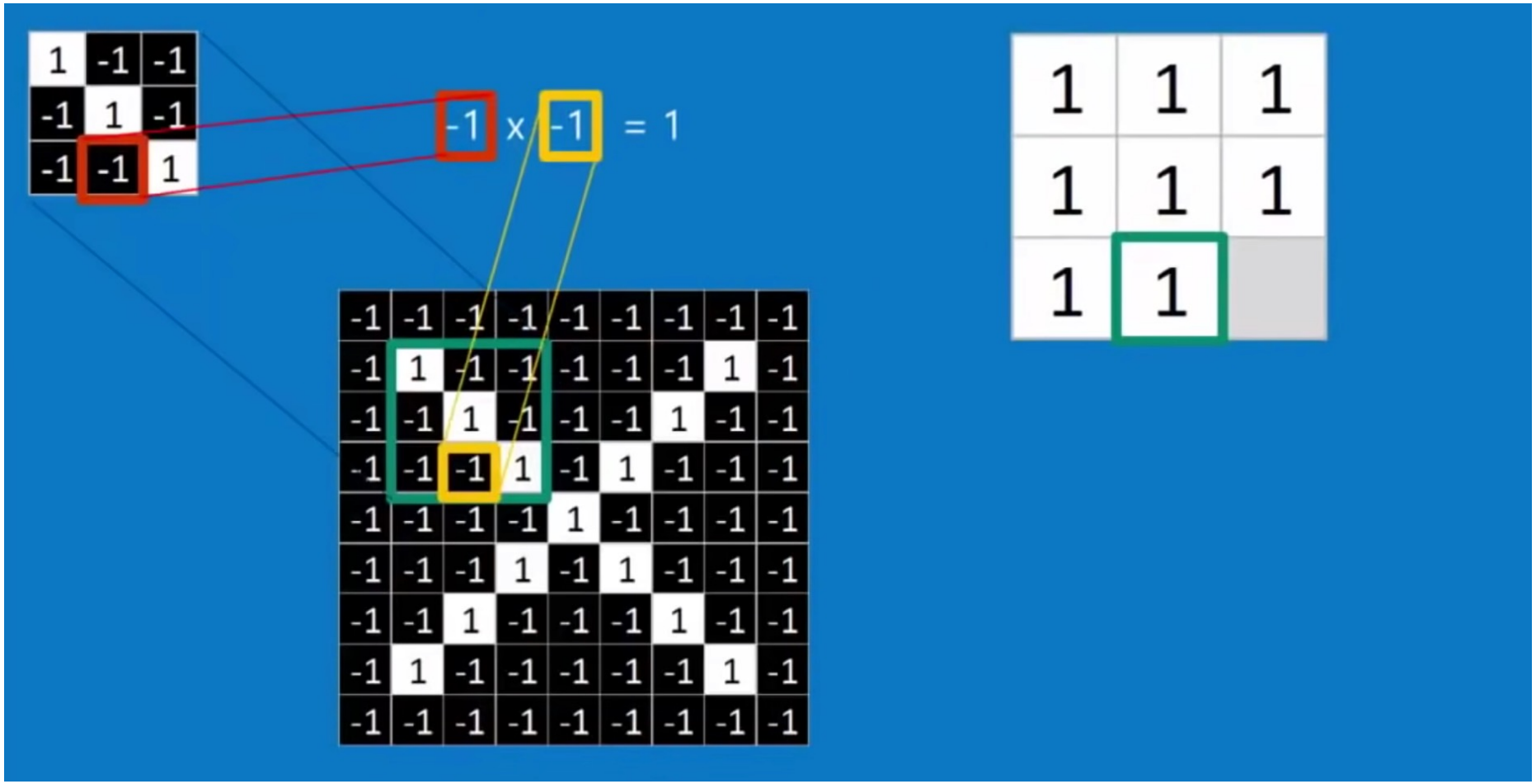
Running the convolution: filtering (1)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

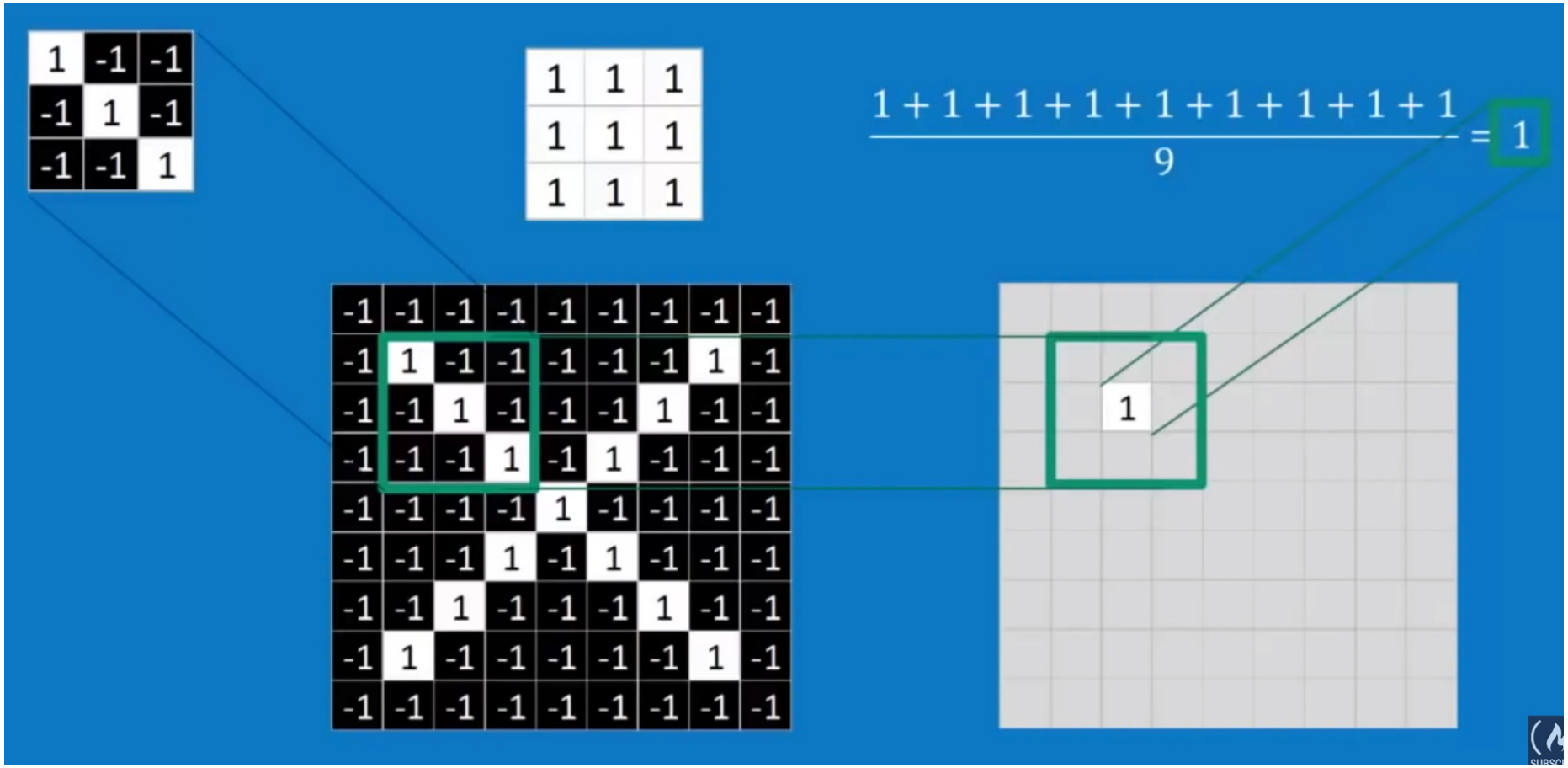
https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Running the convolution: filtering (2)



https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

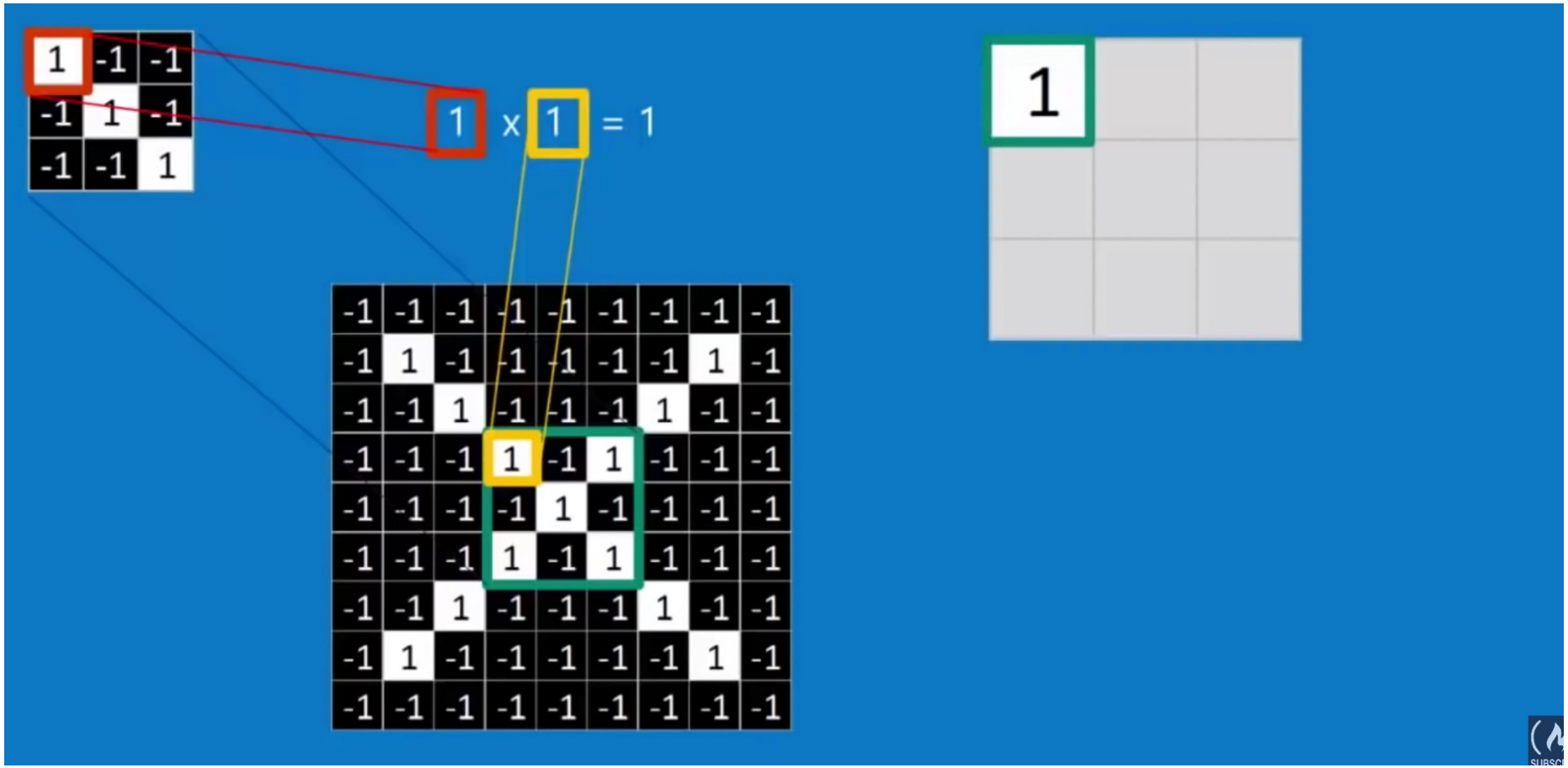
Running the convolution (3)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

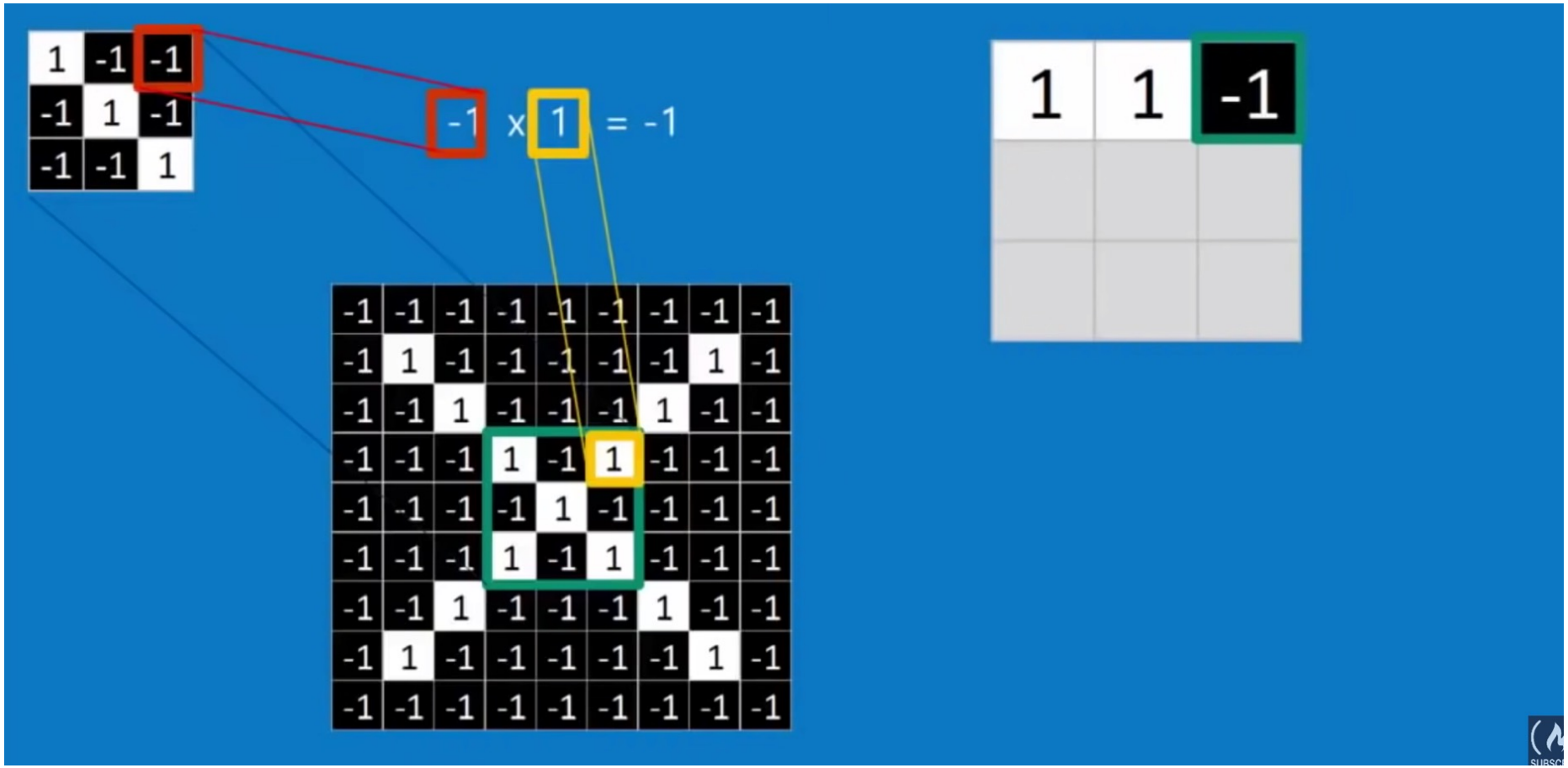
Running the convolution: filtering (4)



https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

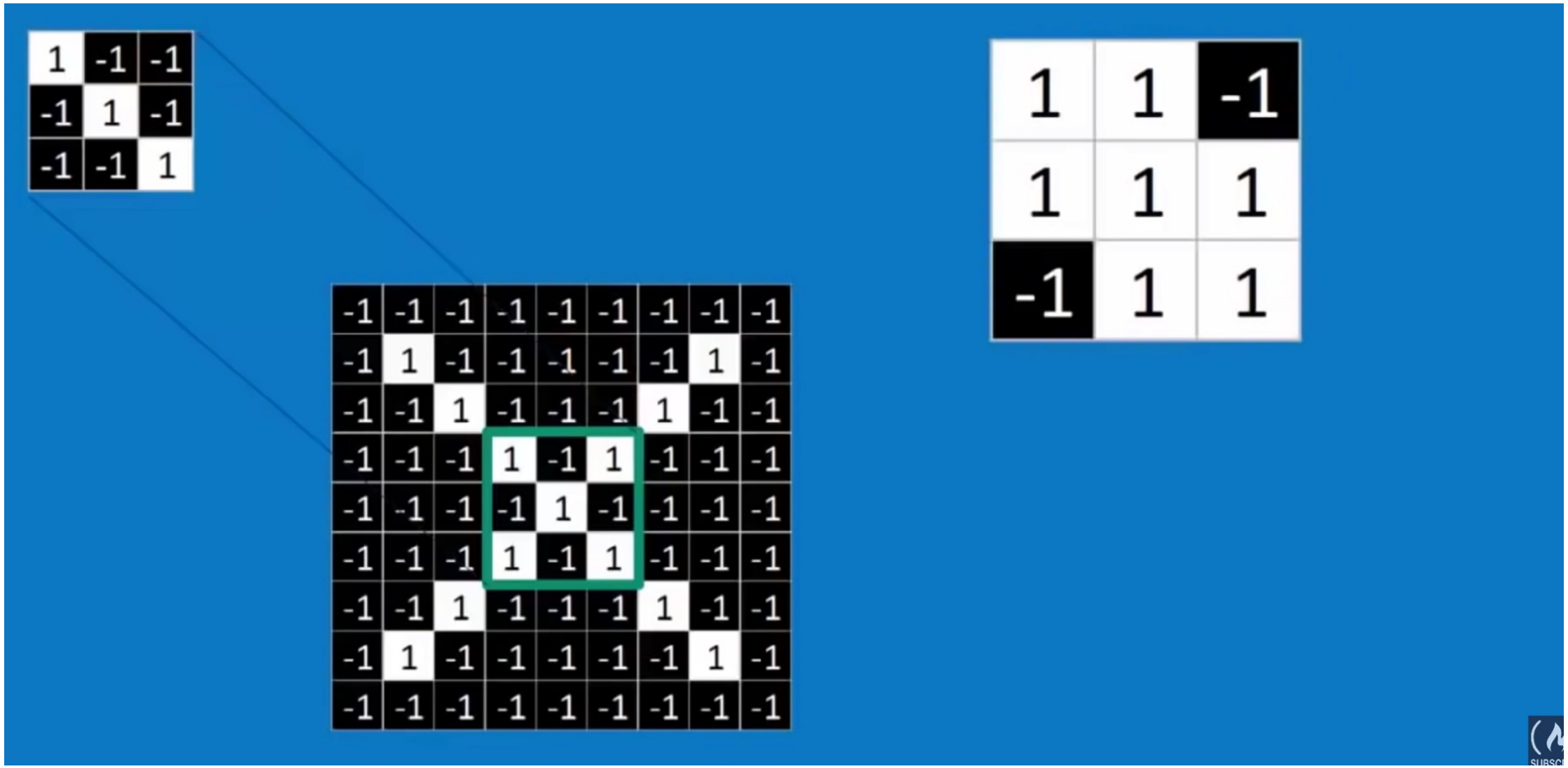
Running the convolution: filtering (5)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

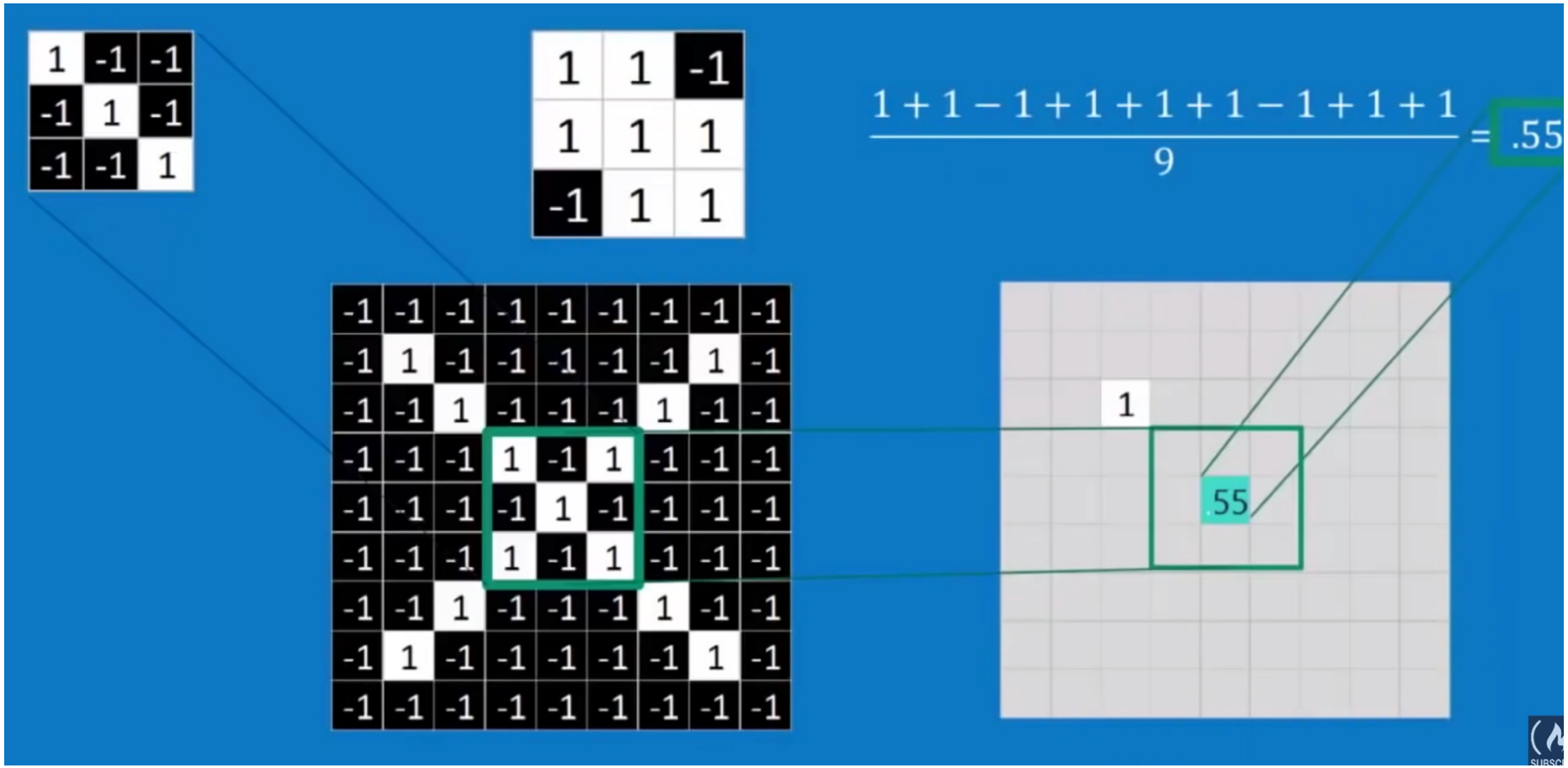
Running the convolution: filtering (6)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

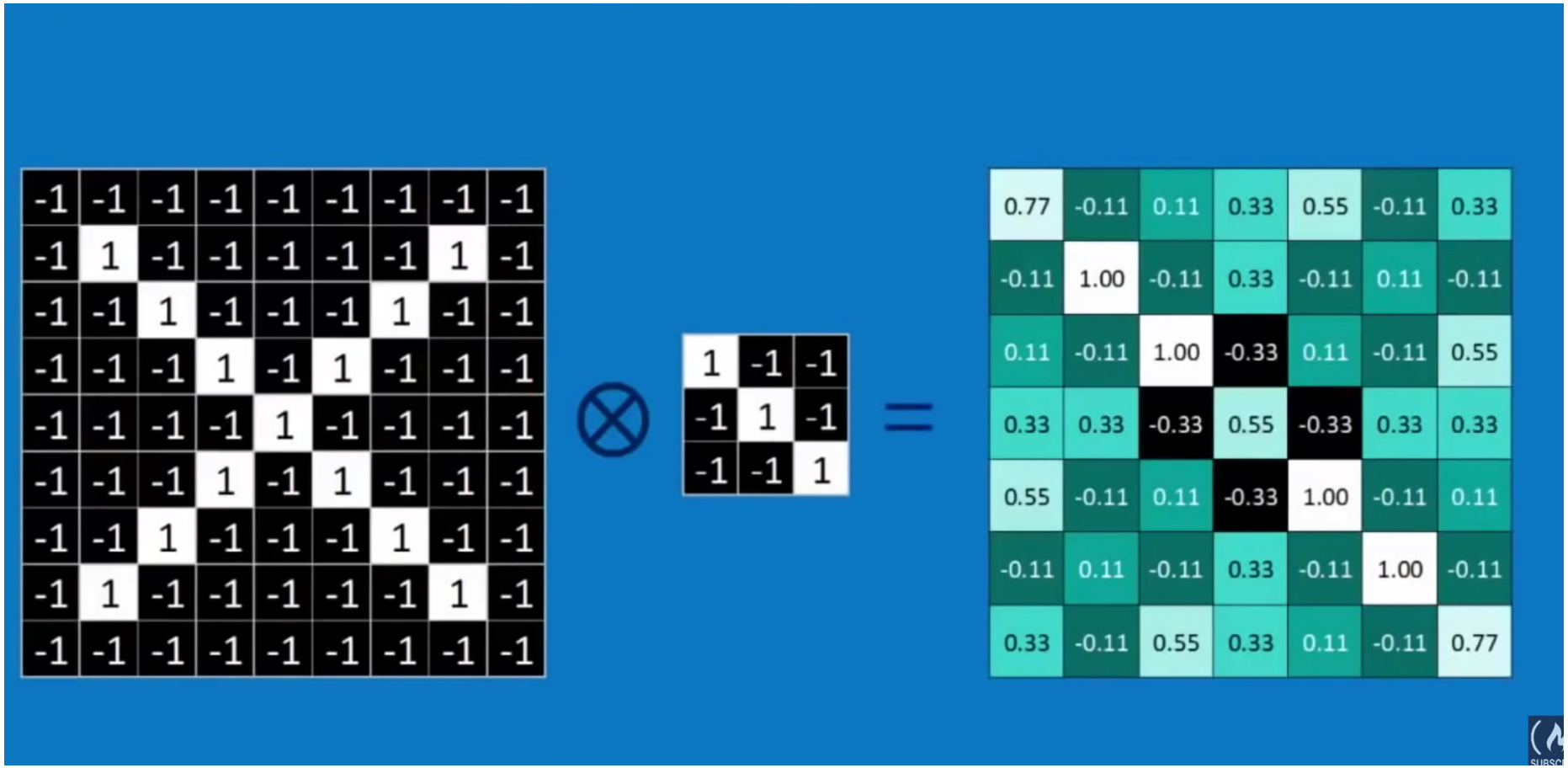
Running the convolution: filtering (7)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

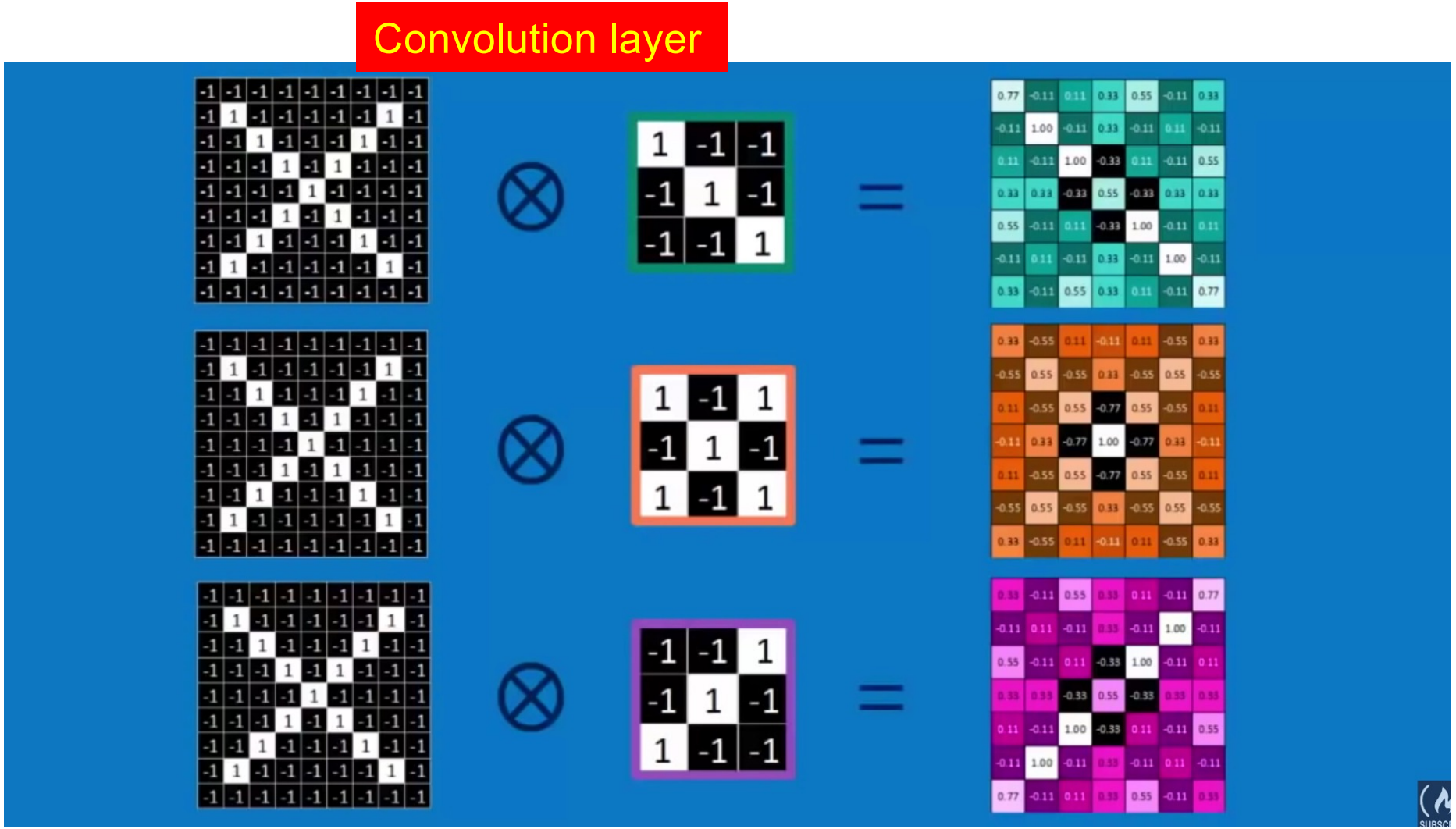
Running the convolution: filtering (8)



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

1 image = stack of filtered images

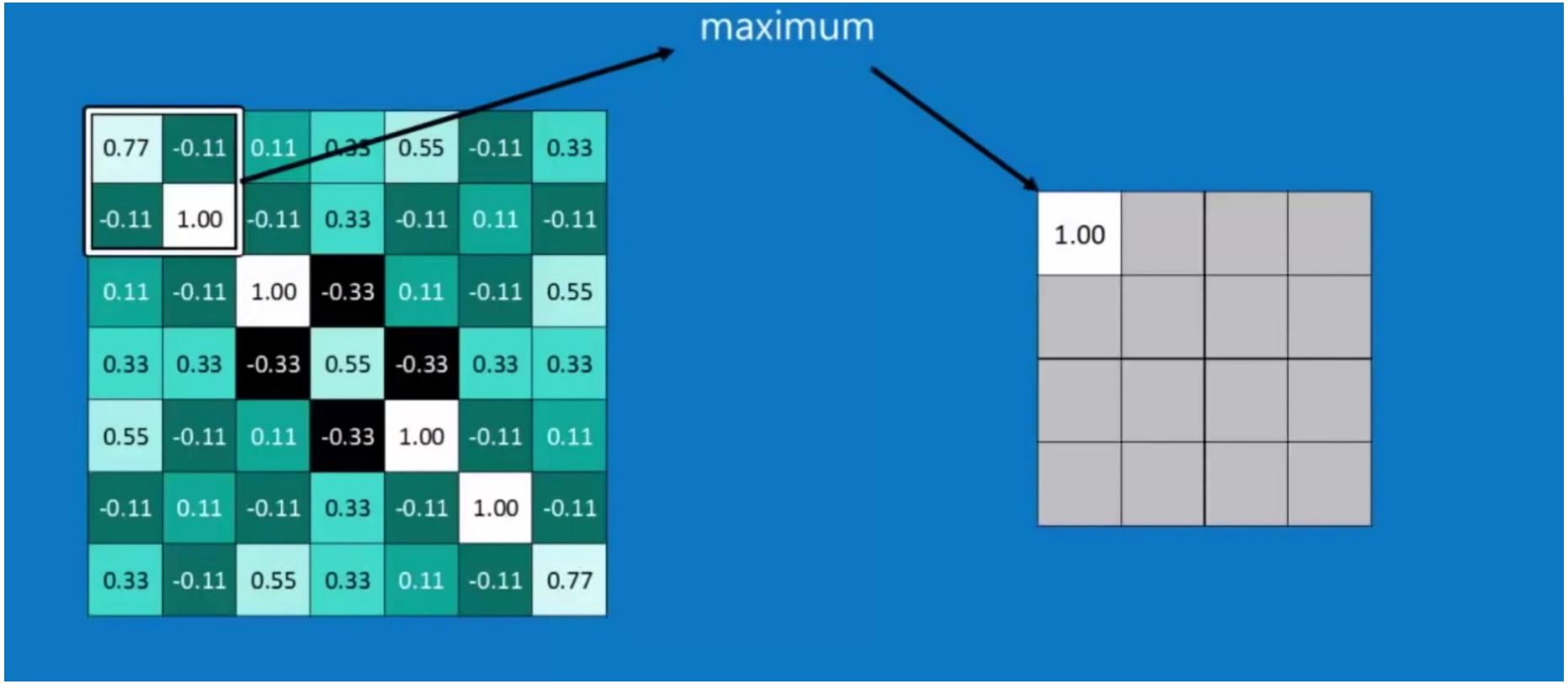


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

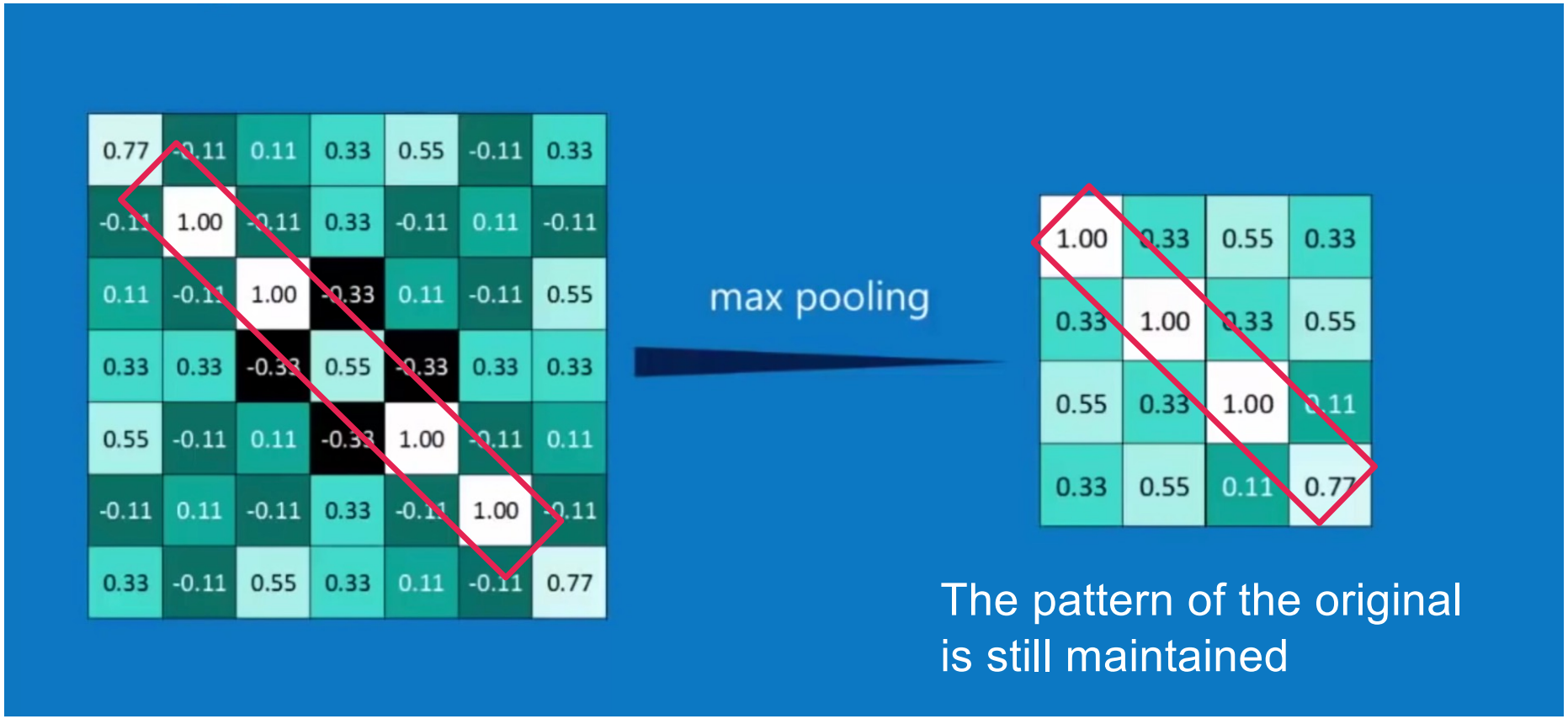
Next, Pooling = reduce the size

- Define window size (usually 2 or 3)
- Define a stride (usually 2)



https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

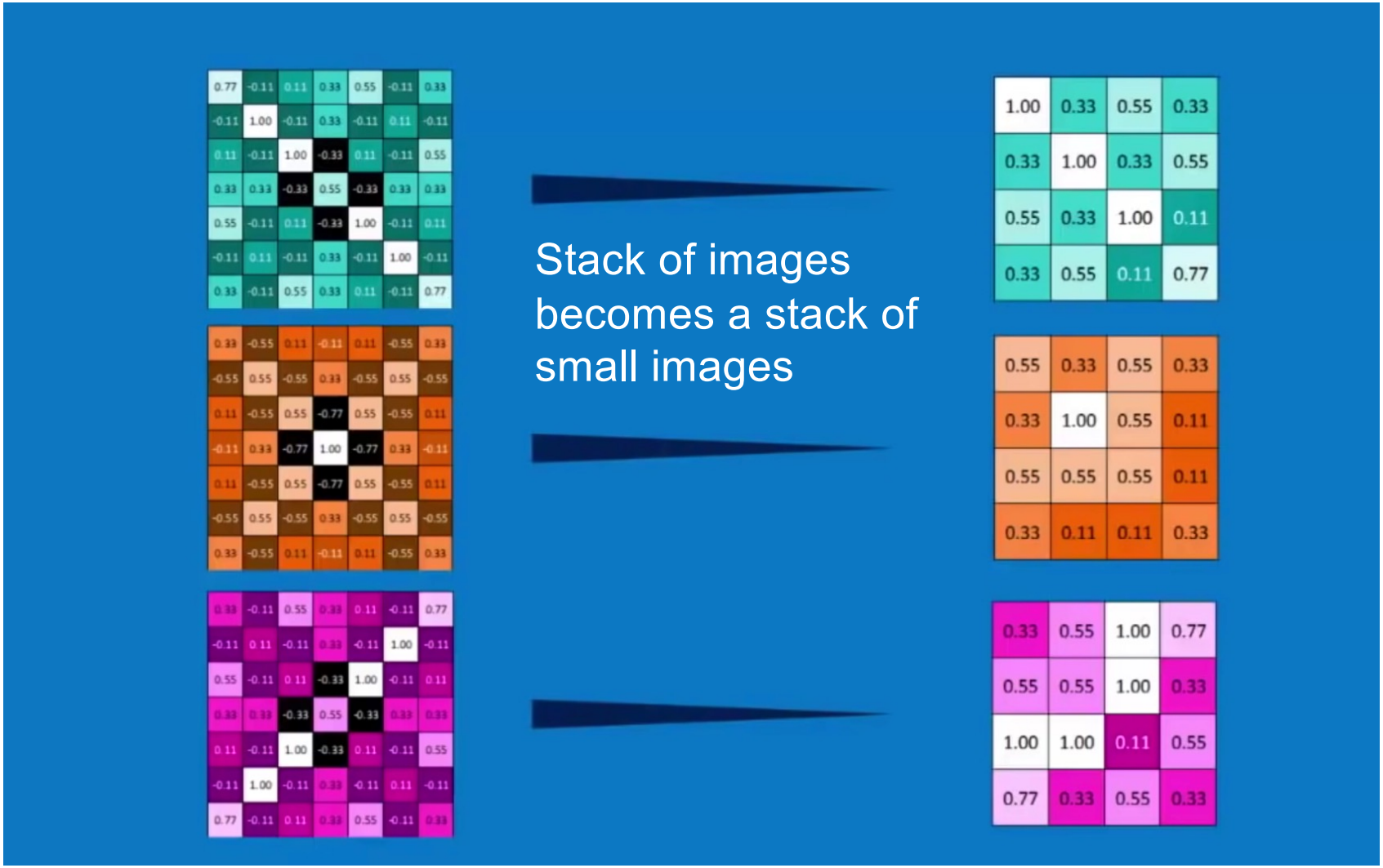
At the end of pooling



Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Do the same for stack of images

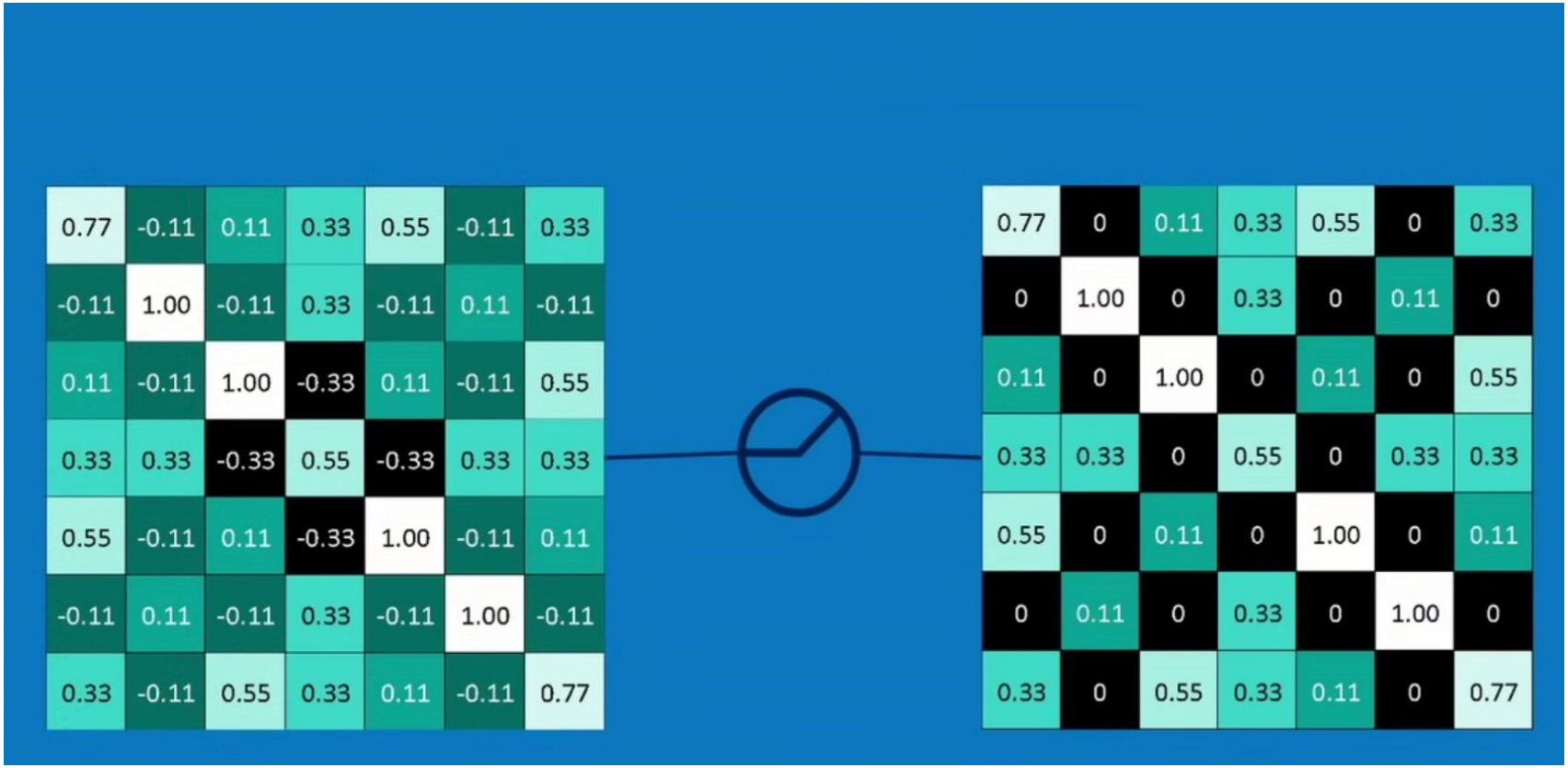


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Normalization: ReLU function

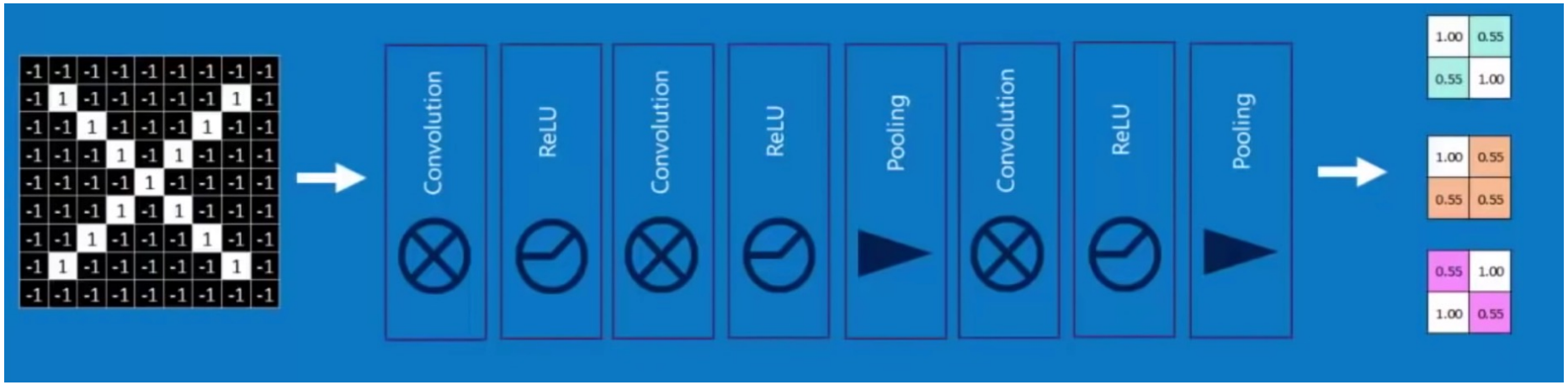
- Keep things from becoming unmanageably large as we are progressing through all the layers

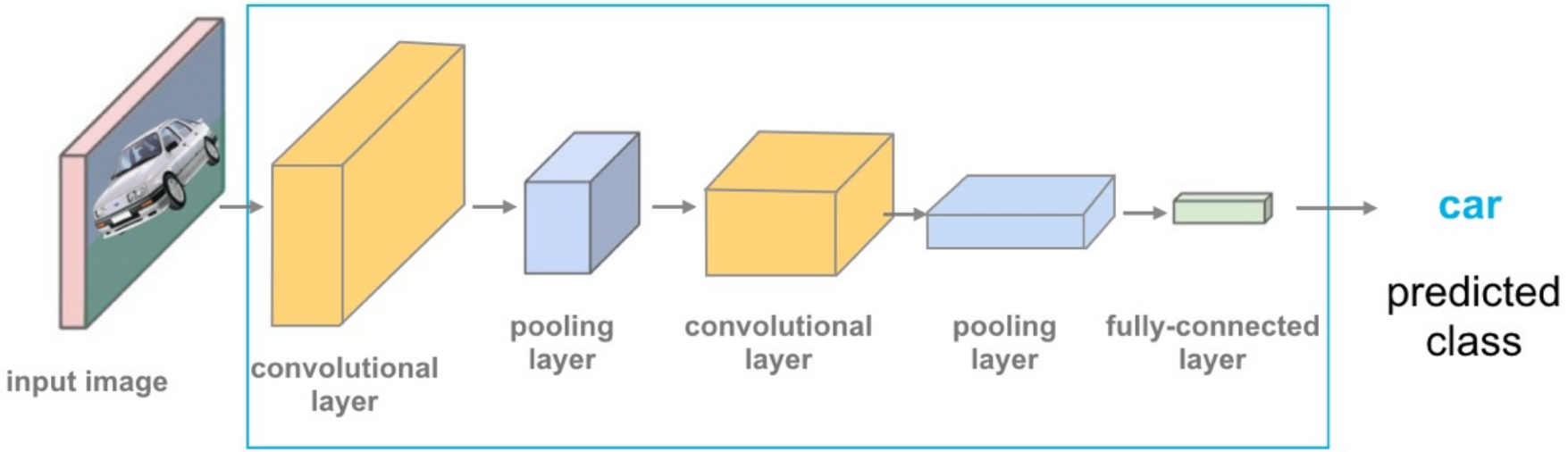


https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Layers can be repeated!

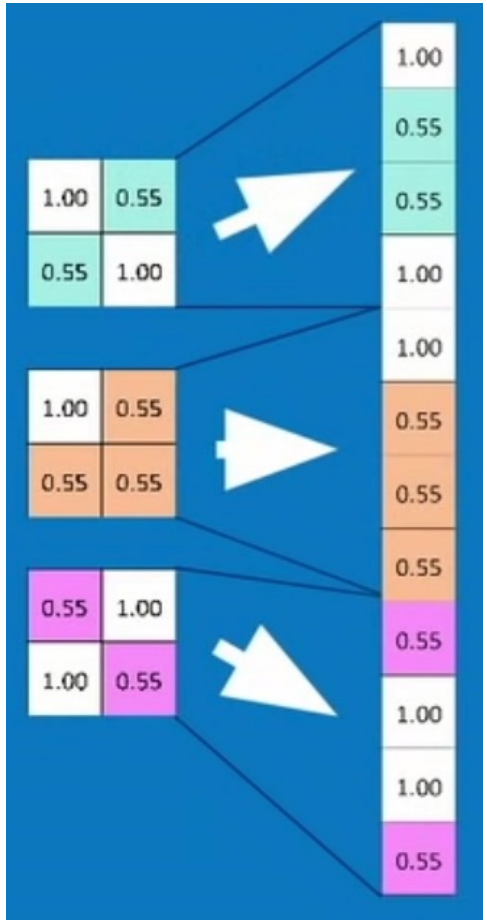
Watch the video on YouTube https://www.youtube.com/watch?v=FmpDIaiMIeA (Brandon Rohrer)

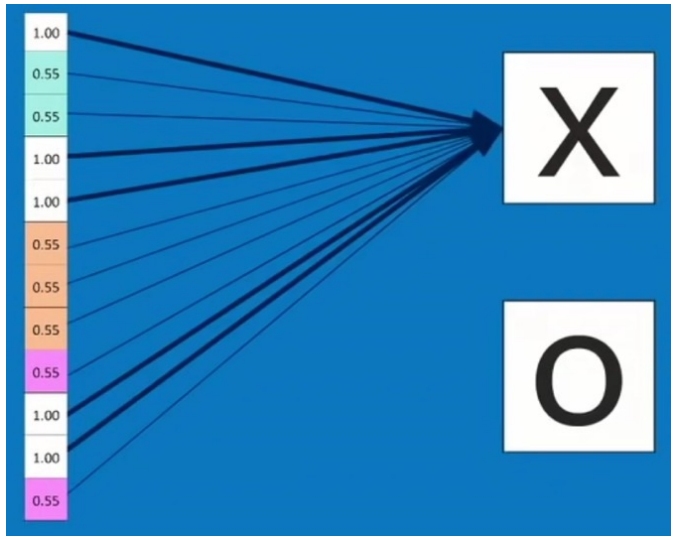


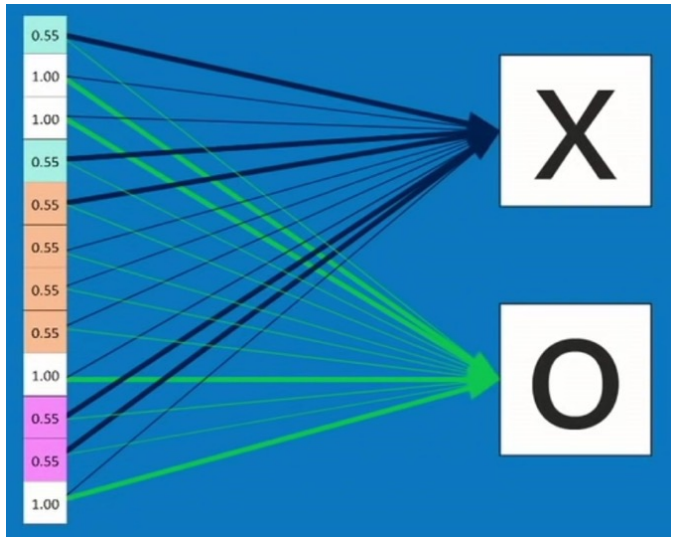


https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

Last step! Let's vote



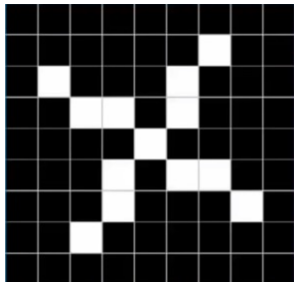


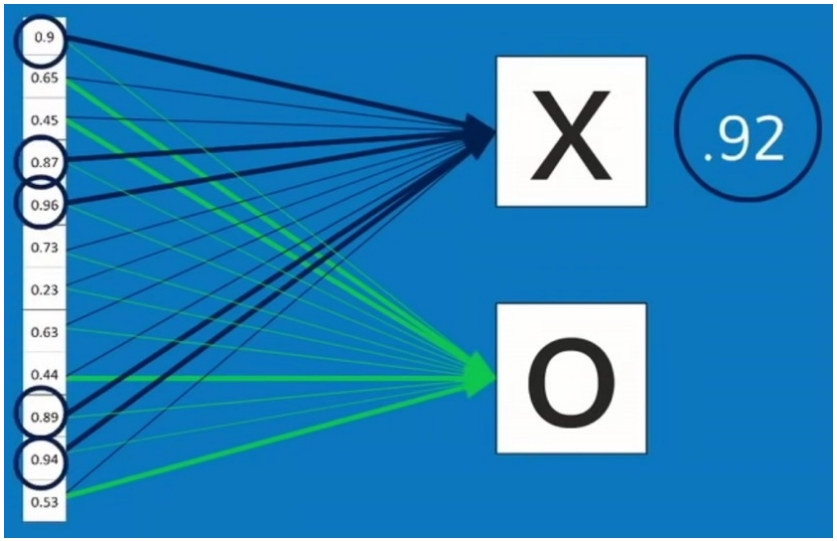


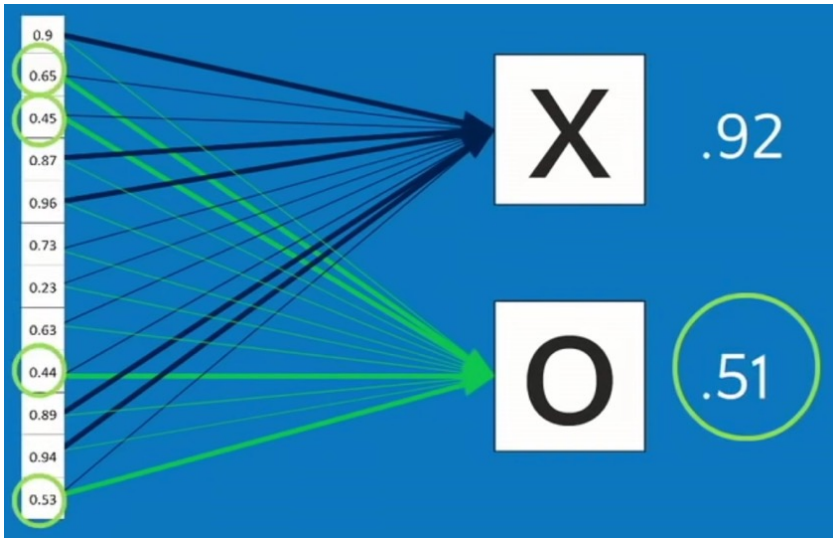
https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

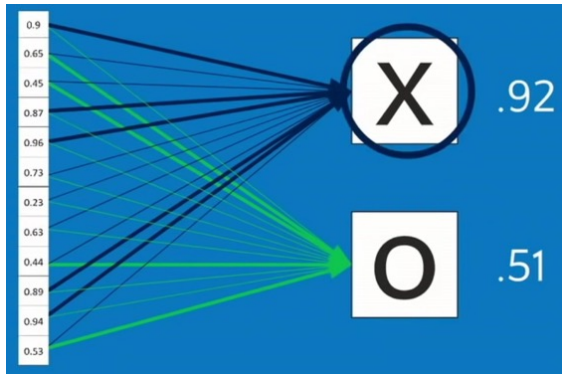
Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

We found the winner!





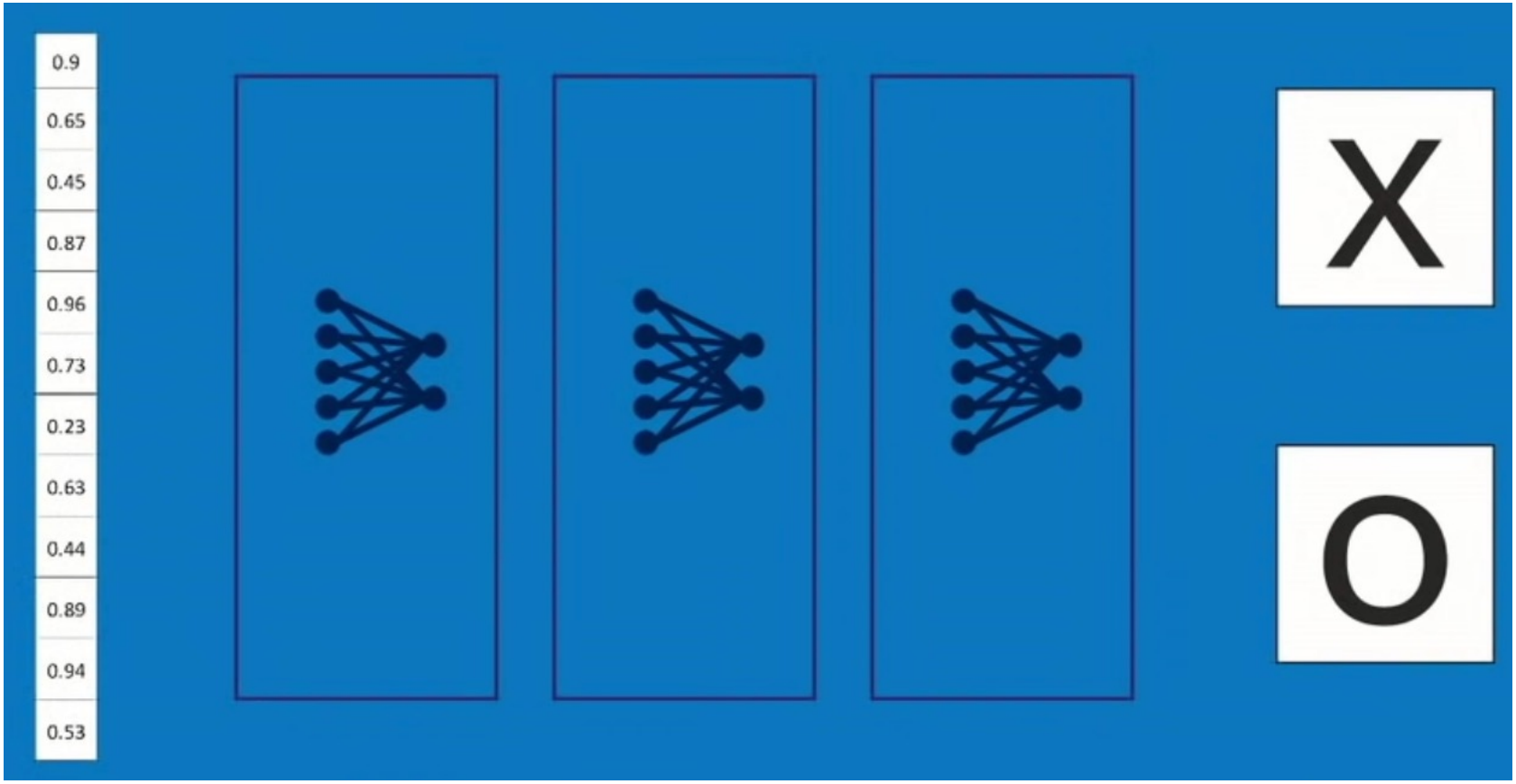




Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

We can always add more stacks!

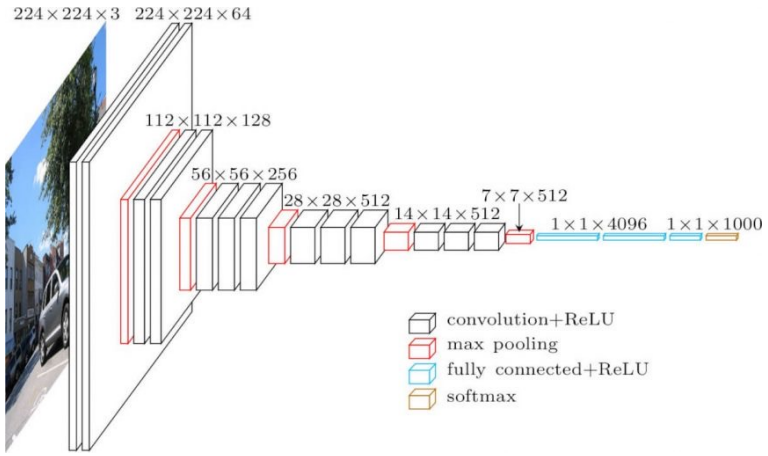


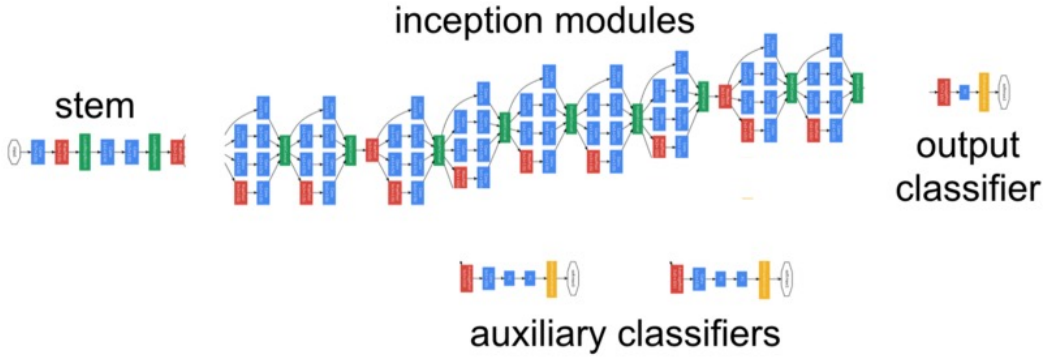
In addition to hidden layers, the NN can have hidden "secret" categories before concluding in is an "X"

https://www.youtube.com/watch?v=dPWYUELwIdM&list=PLhnok3M54IAVKHLvl8E7ogmlBnFNBUY70&index=5&t=3097s

DNN for image classification

- Top 4 Pre-Trained Models for Image Classification with Python Code
- https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/

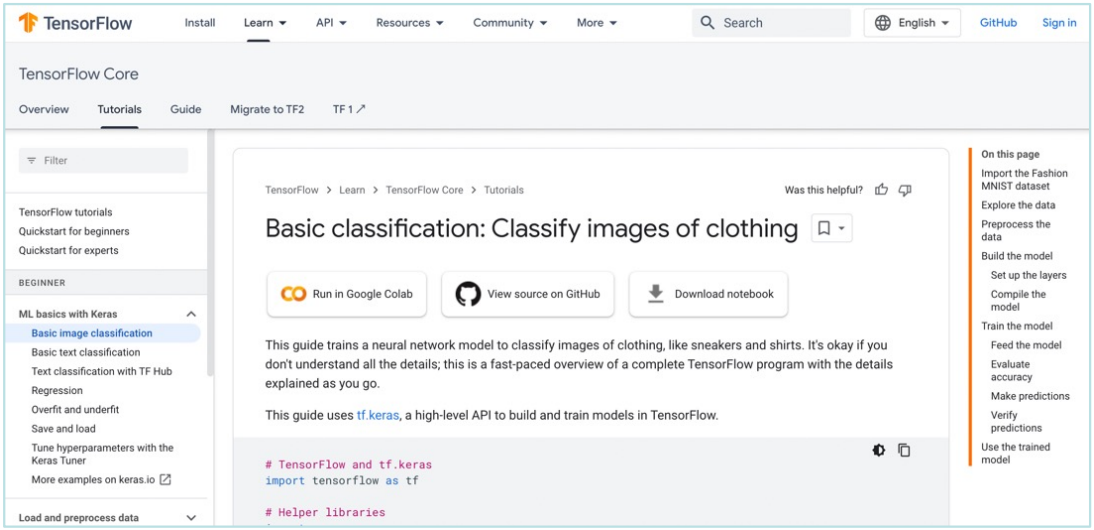




Deep Learning Examples

TensorFlow tutorial 1

- ML basics with Keras
  - Basic image classification
  - https://www.tensorflow.org/tutorials/keras/classification

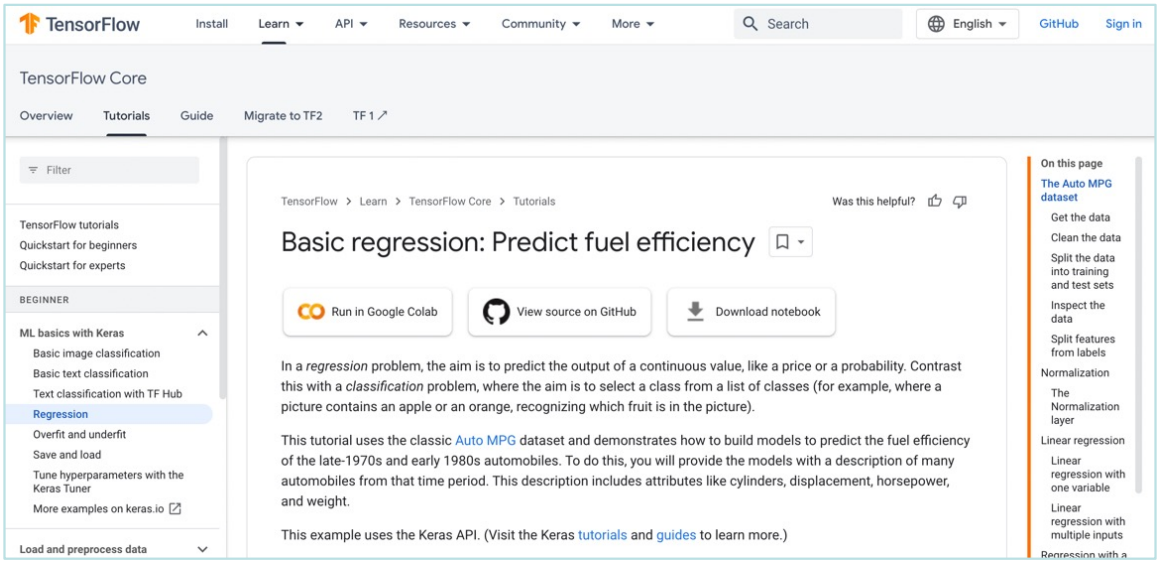


- You can click on "Run in Google Colab"
- Then use "Copy to Drive" to get your own copy for further testing and editing

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

TensorFlow tutorial 2

- ML basics with Keras
  - Basic regression
  - https://www.tensorflow.org/tutorials/keras/regression

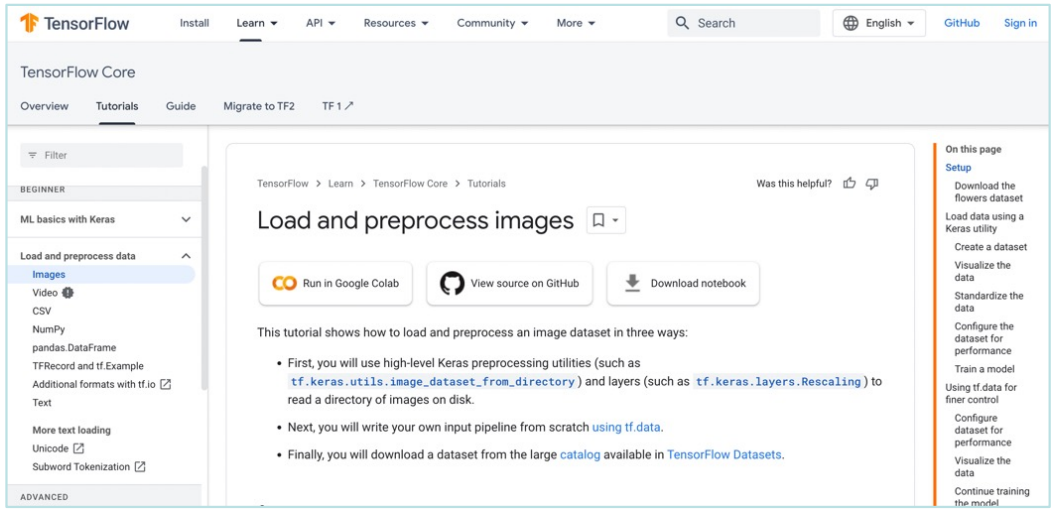


- You can click on "Run in Google Colab"
- Then use "Copy to Drive" to get your own copy for further testing and editing

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

TensorFlow tutorial 3

- Load and preprocess data
  - Images
  - https://www.tensorflow.org/tutorials/load_data/images



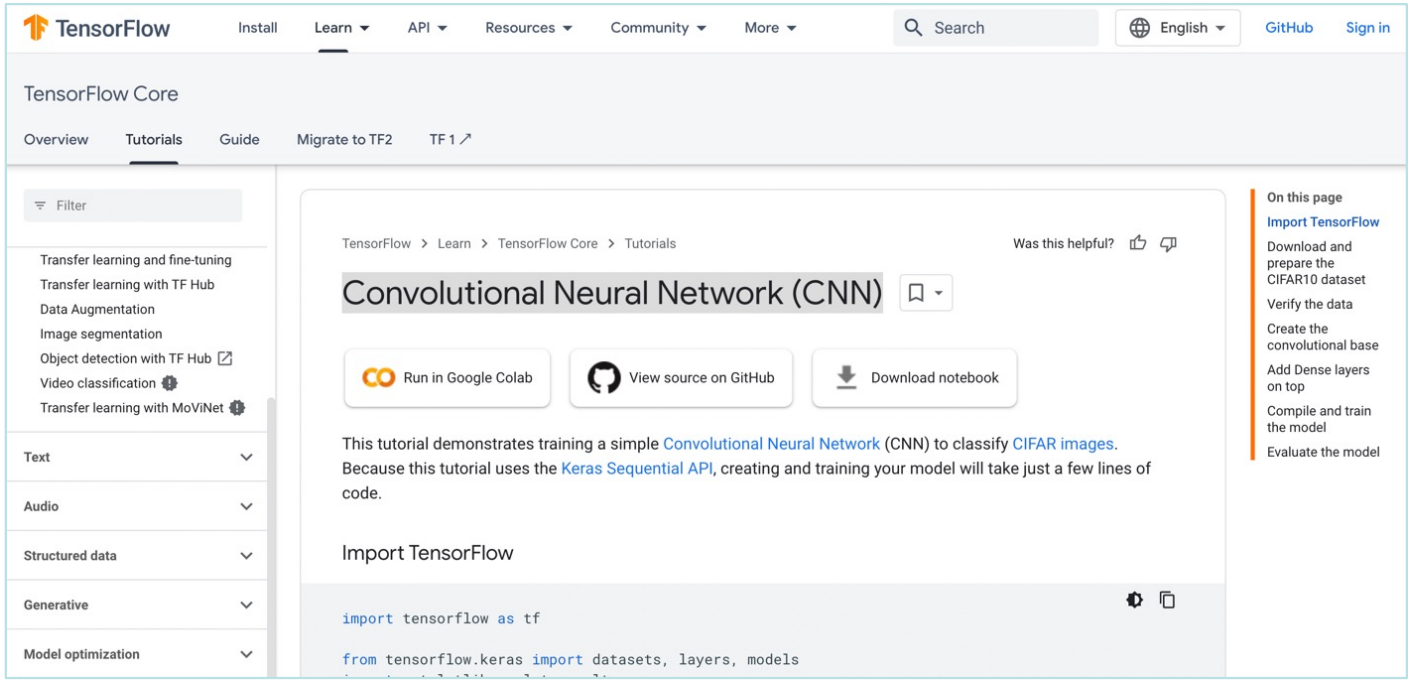
- You can click on "Run in Google Colab"
- Then use "Copy to Drive" to get your own copy for further testing and editing

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

CNN with TensorFlow

Assignment to do in lab

- https://www.tensorflow.org/tutorials/images/cnn



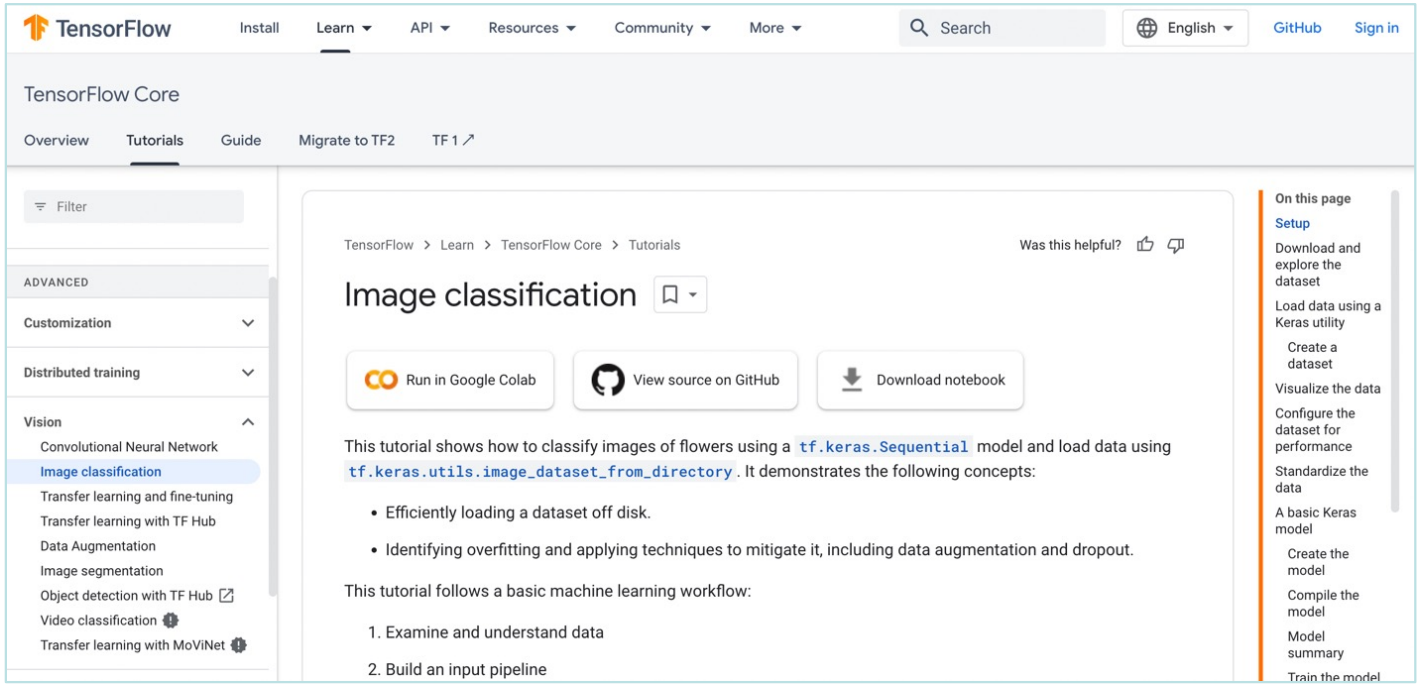
- You can click on "Run in Google Colab"
- Then use "Copy to Drive" to get your own copy for further testing and editing

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Image classification with TensorFlow

Assignment to do in lab

- https://www.tensorflow.org/tutorials/images/classification



- You can click on "Run in Google Colab"
- Then use "Copy to Drive" to get your own copy for further testing and editing

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

AI FOR CPS

AI for CPS?

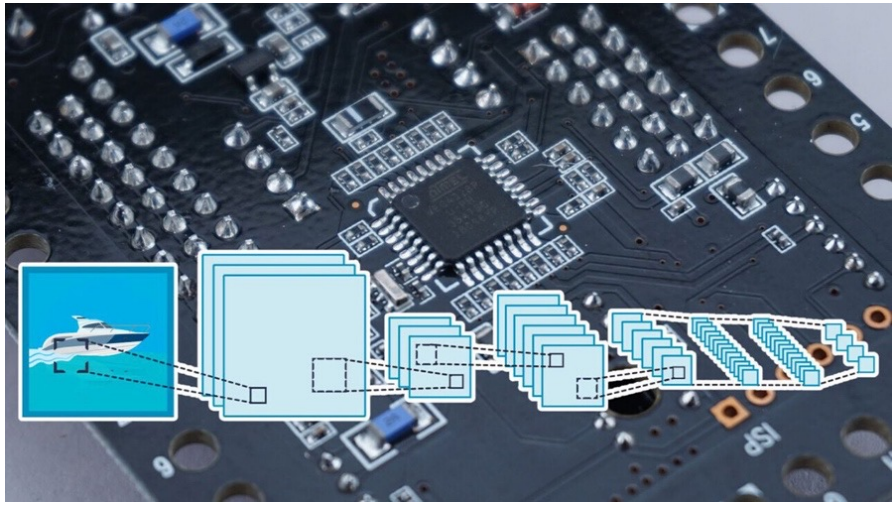
- *"Sensors, computational and physical world are closely integrated in CPS. CPS includes both conventional embedded systems and control systems that are presumed to be remodelled by emerging methodologies and integration of the Internet of Things. IoT is the base or enabling technology for cyber-physical systems."*
- "While deep learning in the cloud has been tremendously successful, it is not applicable in all situations. Many applications require on-device inference." [see ref]
- "And the delay caused by the roundtrip to the cloud is prohibitive for applications that require real-time ML inference" [see ref]

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

ref: https://bdtechtalks.com/2022/01/17/mcunetv2-tinyml-deep-learning-microcontrollers/

AI on microcontrollers?

- "Your iPhone now runs facial recognition and speech recognition on device. Your Android phone can run on-device translation. Your Apple Watch uses machine learning to detect movements and ECG patterns." [see ref]
- BUT AI on microcontrollers needs to go a step further to make AI techniques computationally tractable on much more resource-constrained devices
- "TinyML takes edge AI one step further, making it possible to run deep learning models on microcontrollers". [see ref]
- https://www.tinyml.org/

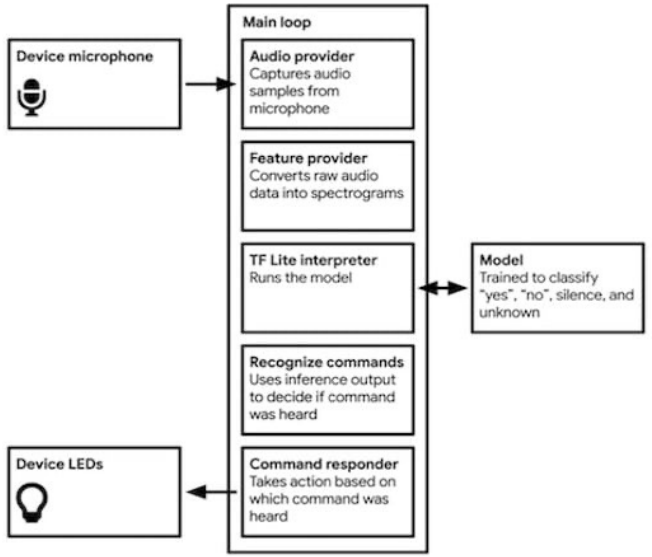


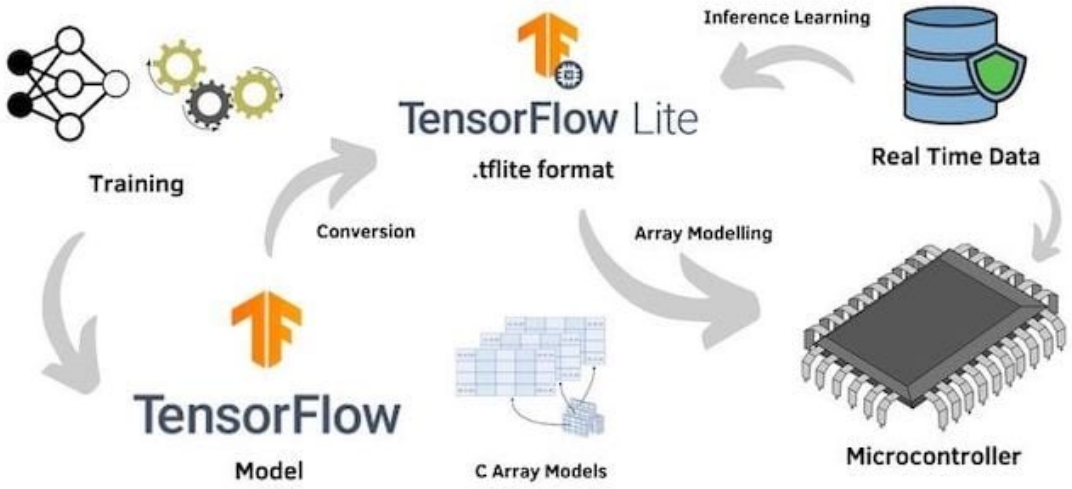
ref: https://bdtechtalks.com/2022/01/17/mcunetv2-tinyml-deep-learning-microcontrollers/

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

TinyML & Tensorflow Lite

- It is already part of our daily life in most of wake-up word application
- The most popular and built-out ecosystem for TinyML development is TensorFlow Lite for Microcontrollers (TF Lite Micro)



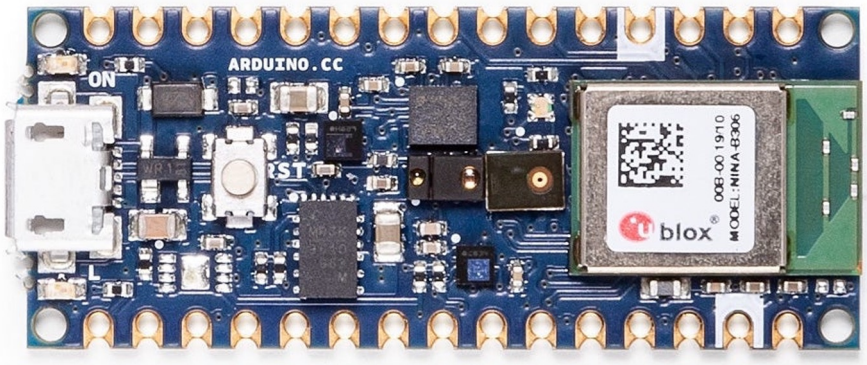


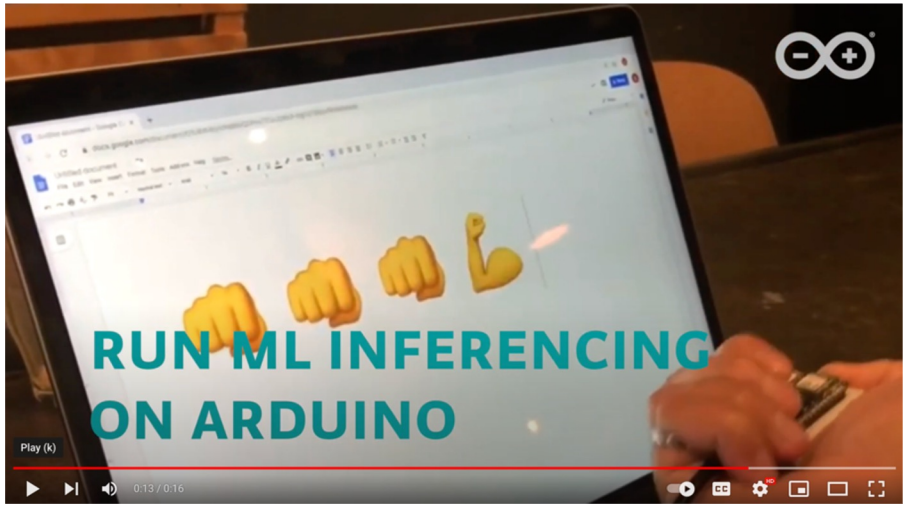
https://www.allaboutcircuits.com/technical-articles/what-is-tinyml/

Example: the Arduino Nano 33 BLE Sense

- ARM Cortex-M4 at 64 MHz with BLE and plenty of sensors! 3D accelerometer, microphone, gesture, light, proximity, barometric pressure, temperature, humidity. Run AI using TinyML and TensorFlow™ Lite.

https://youtu.be/HzCRZsGJLbI



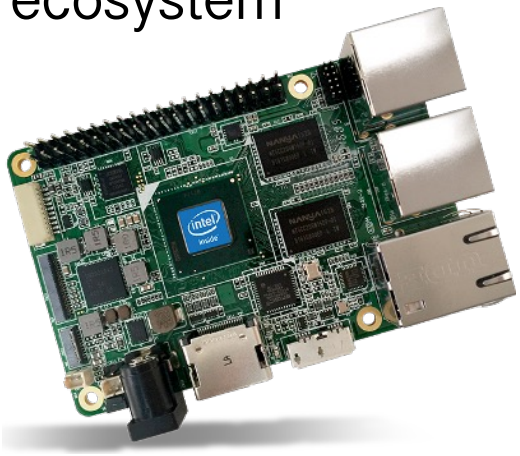


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

AI on single computer boards?

- Single Board Computer can be part of the CPS ecosystem





- They can natively run complex AI models but for some real-time applications, AI accelerators can provide them with specialized AI hardware such as TPU (Tensor Processing Unit), VPU (Visual Processing Unit), …
- For SBC, the most convenient AI accelerators are those embedded on the USB stick format

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

AI accelerators in USB stick format

- Depending on the product, the AI/ML models can be different, with performances depending on how you plan to use them
- Some provide pre-trained DL models for image classification and object detection, allowing real-time applications on SBC
- They can support most of the well-know AI frameworks/libs such as TensorFlow, ApacheMXNet, PyTorch,

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

Example: real-time embedded AI on RPI

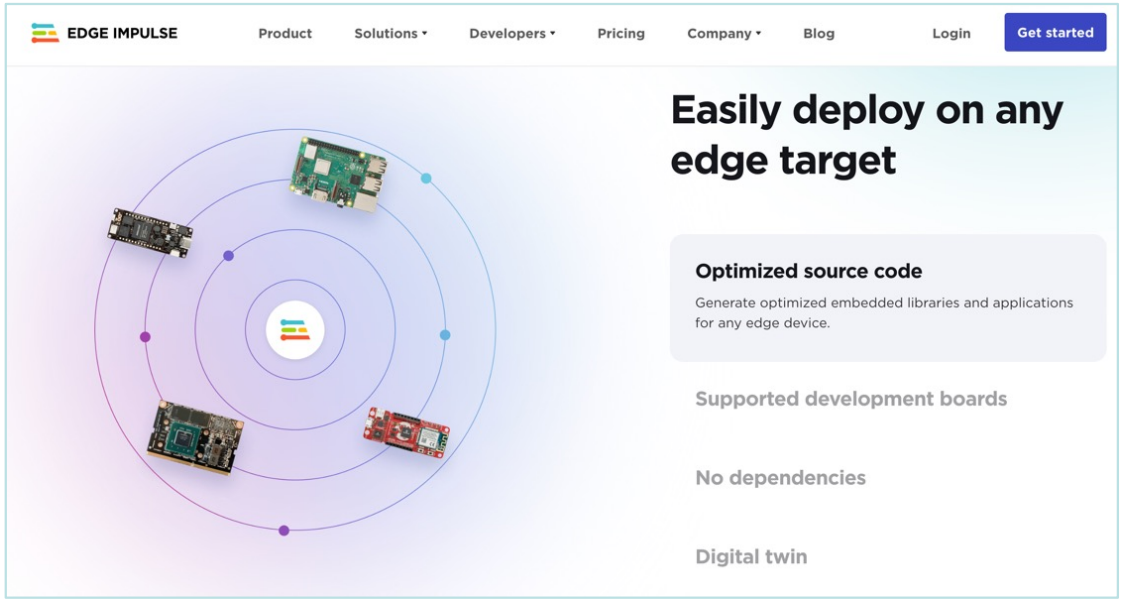
- Dive in the exciting world of real-time embedded AI with hardware accelerators (Intel Neural Compute Stick, Google Coral USB, …)

https://youtu.be/6XnktdajhxU

Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

EdgeImpulse

- https://www.edgeimpulse.com/
- Edge Impulse is a cloud-based machine learning operations (MLOps) platform for developing embedded and edge ML (TinyML) systems that can be deployed to a wide range of hardware targets
- It enables the deployment of highly-optimized ML on hardware ranging from MCUs to CPUs and custom AI accelerators

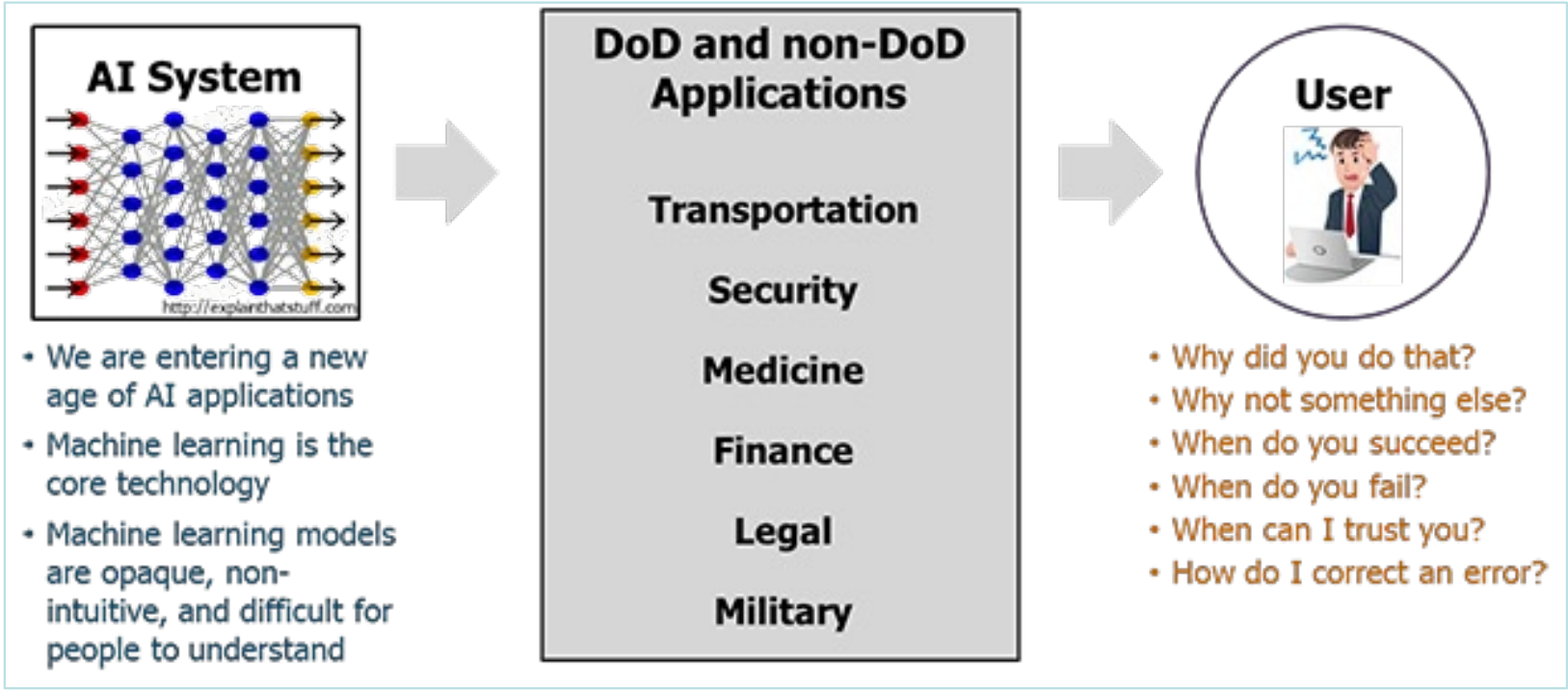


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

X AI ?

eXplainable AI?

- "the effectiveness of AI systems is limited by the machine's current inability to explain their decisions and actions to human users" [see ref]

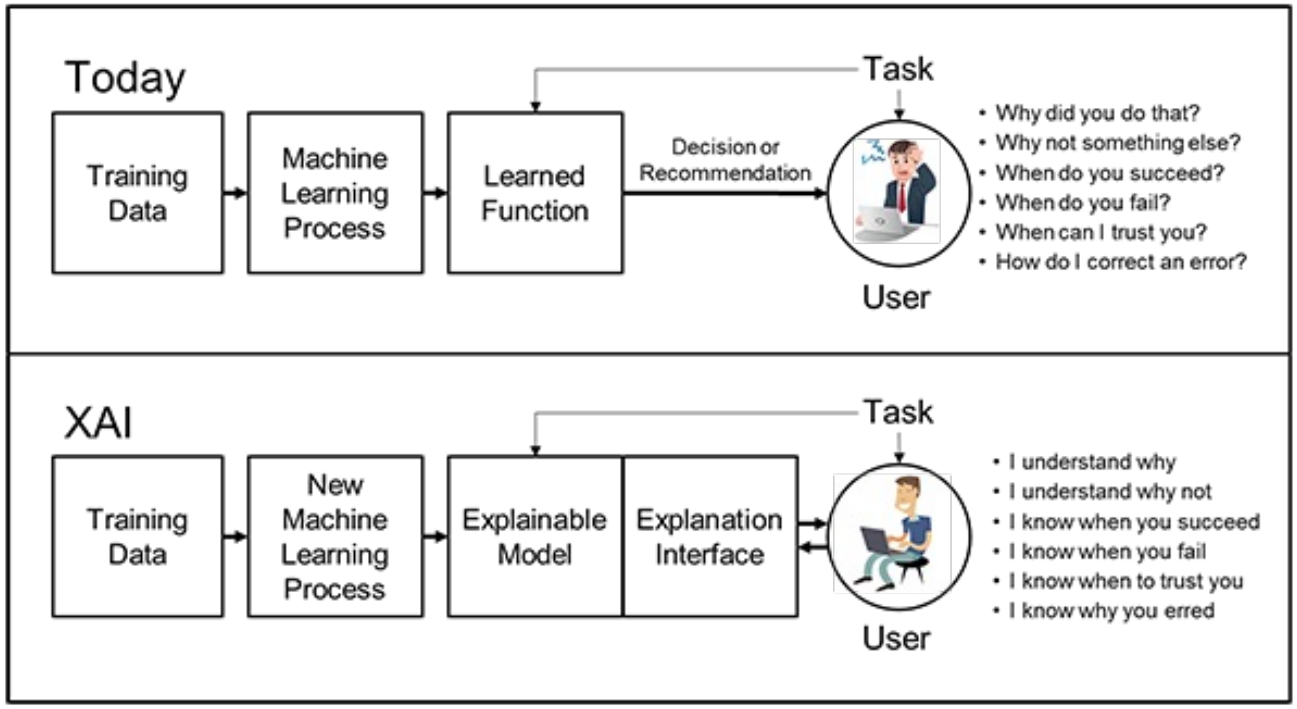


Pr. Congduc Pham
http://www.univ-pau.fr/~cpham

ref: https://www.darpa.mil/program/explainable-artificial-intelligence

eXplainable AI?

- "New machine-learning systems will have the ability to explain their rationale, characterize their strengths and weaknesses, and convey an understanding of how they will behave in the future" [see ref]



ref: https://www.darpa.mil/program/explainable-artificial-intelligence