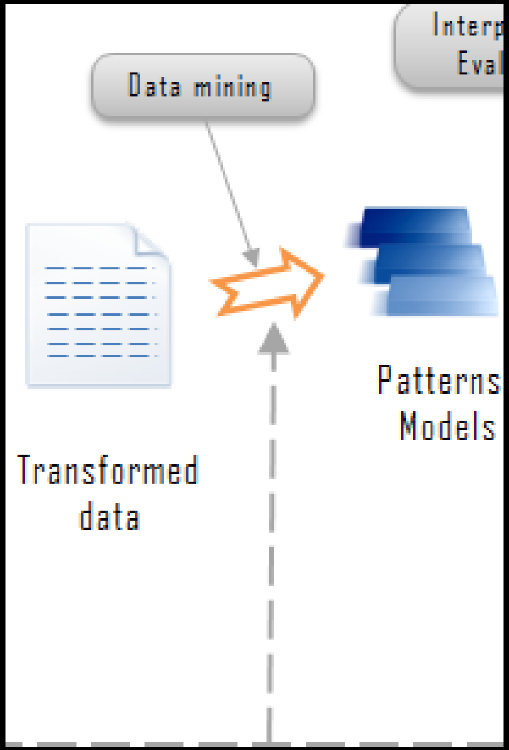
MLP and CNN for dairy farming

Prof. Hélène Soyeurt

SensAIFood – May 2025 – Gembloux, Belgium

LIÈGE université
Gembloux
Agro-Bio Tech

WHAT IS THE OBJECTIVE ?



Regression
- Quantitative y values
- Prediction

Classification
- Quantitative y valyes
- Separation into several classes

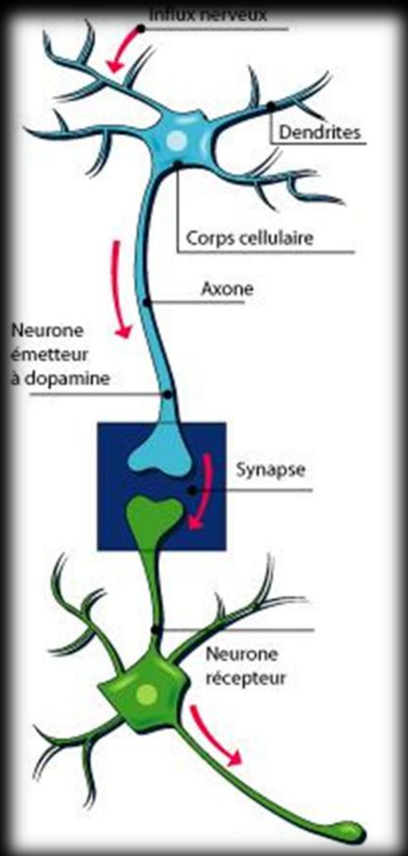
Clustering
- Creation of sub-groups within the same dataset

Supervised

Not supervised

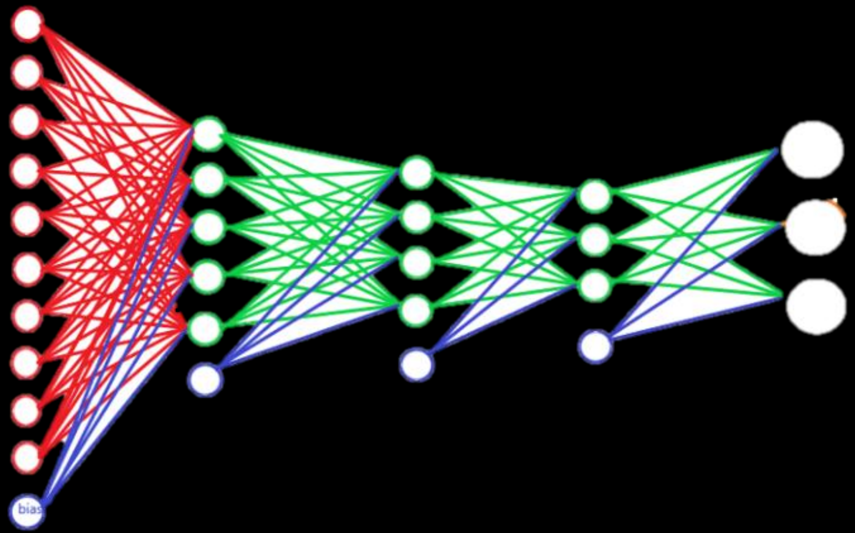
INTRODUCTION

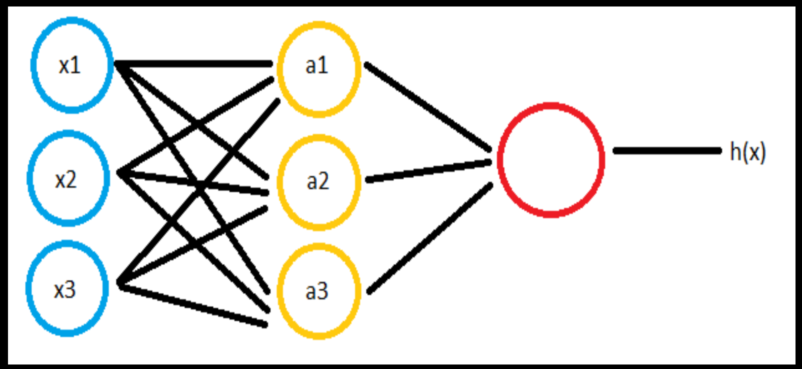
- Artificial neural network (ANN)
- Basic of deep learning
- McCulloch-Pitts (1943) represent the neuron as a binary tool
- Great interest in the 80's and 90's
- Mimic the structure of brain composed of neurons and synapses

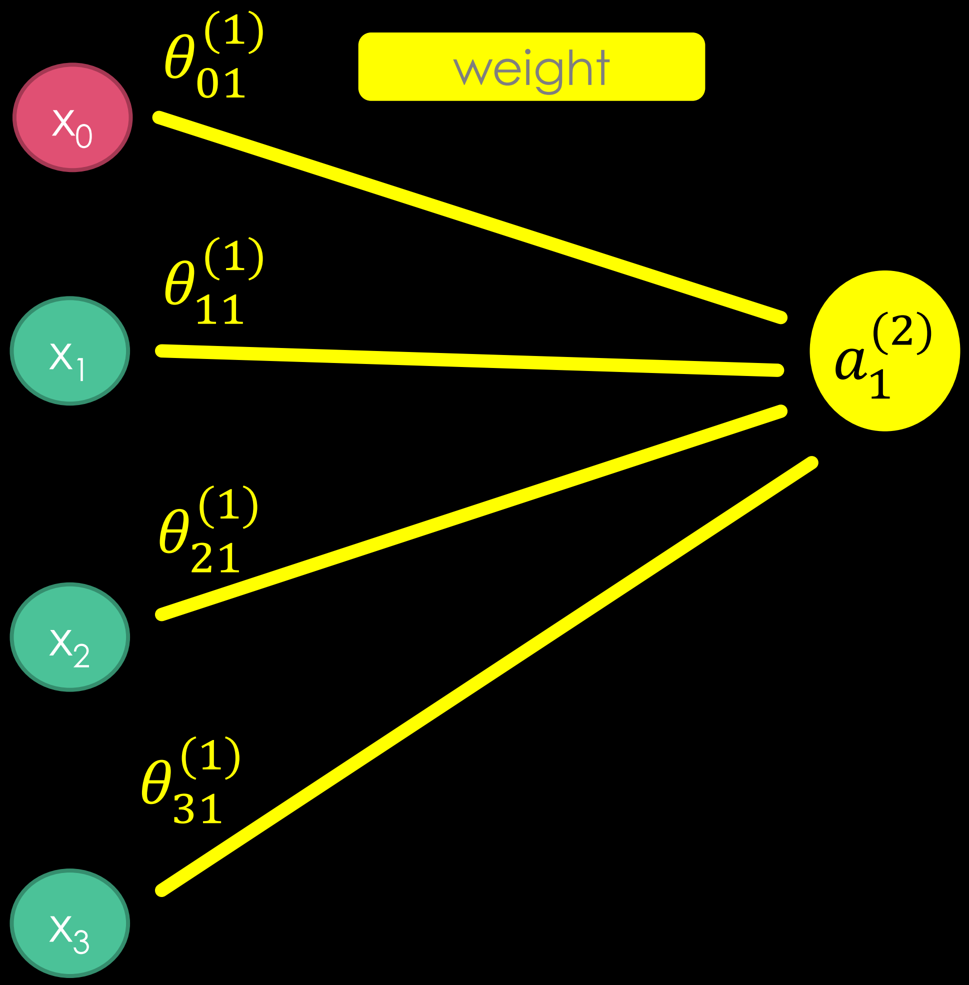


MULTILAYER PERCEPTRON (MLP)

- Units/nodes/neurons
- Fully connected or not
- One or more hidden layers

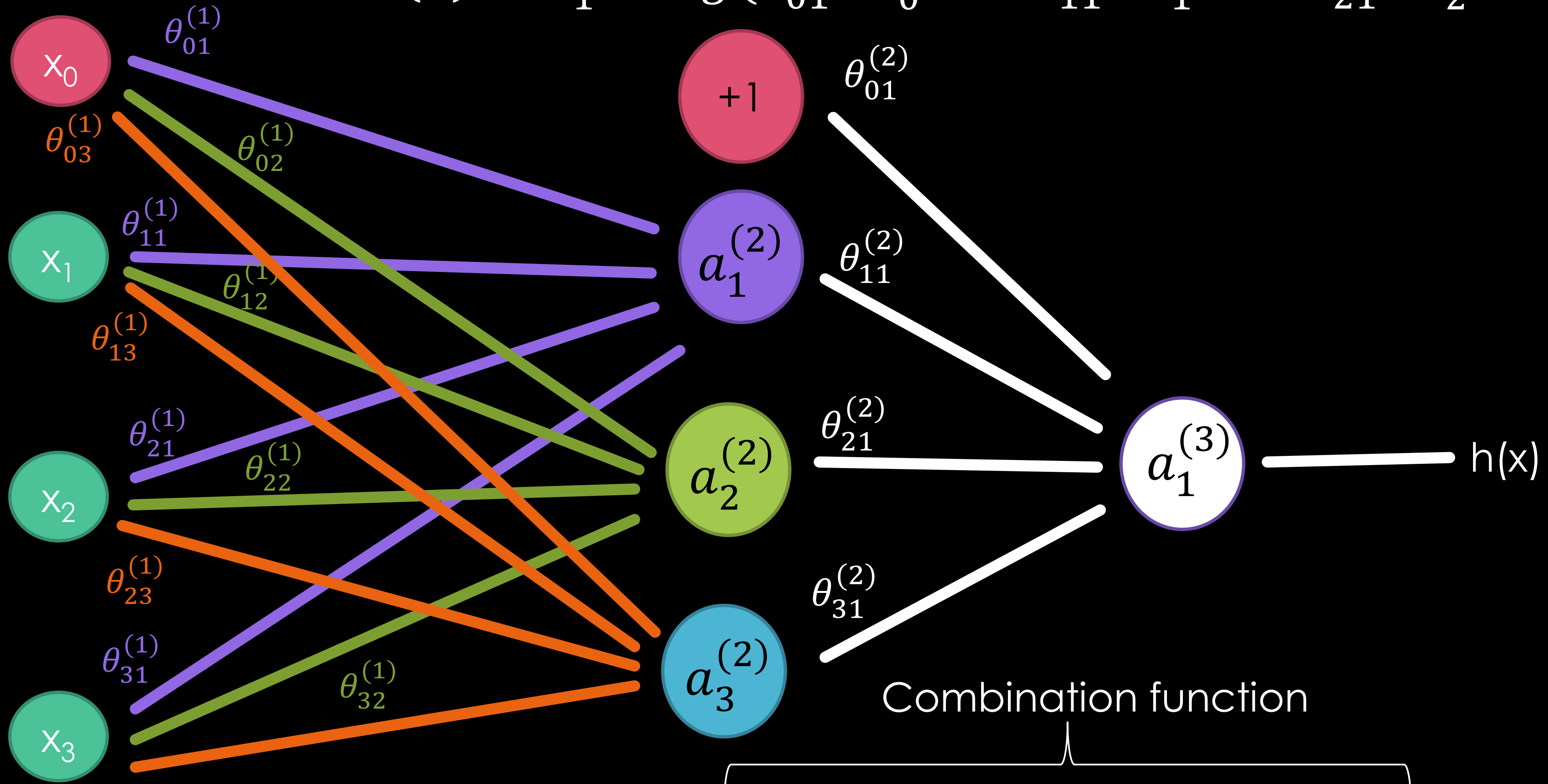






$\theta_{01}^{(1)}$
weight
x_0
$\theta_{11}^{(1)}$
x_1
$a_1^{(2)}$
$\theta_{21}^{(1)}$
x_2
$\theta_{31}^{(1)}$
x_3

$$h(x) = a_1^{(3)} = g(\theta_{01}^{(2)} . a_0^{(2)} + \theta_{11}^{(2)} . a_1^{(2)} + \theta_{21}^{(2)} . a_2^{(2)} + \theta_{31}^{(2)} . a_3^{(2)})$$



$$a_1^{(2)} = g(\theta_{01}^{(1)} . x_0 + \theta_{11}^{(1)} . x_1 + \theta_{21}^{(1)} . x_2 + \theta_{31}^{(2)} . x_3)$$

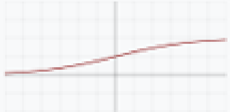
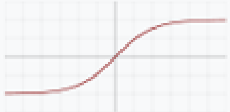
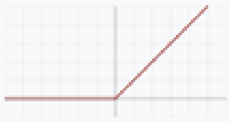
$$a_2^{(2)} = g(\theta_{02}^{(1)} . x_0 + \theta_{12}^{(1)} . x_1 + \theta_{22}^{(1)} . x_2 + \theta_{32}^{(2)} . x_3)$$

$$a_3^{(2)} = g(\theta_{03}^{(1)} . x_0 + \theta_{13}^{(1)} . x_1 + \theta_{23}^{(1)} . x_2 + \theta_{33}^{(2)} . x_3)$$

Activation/transfert function

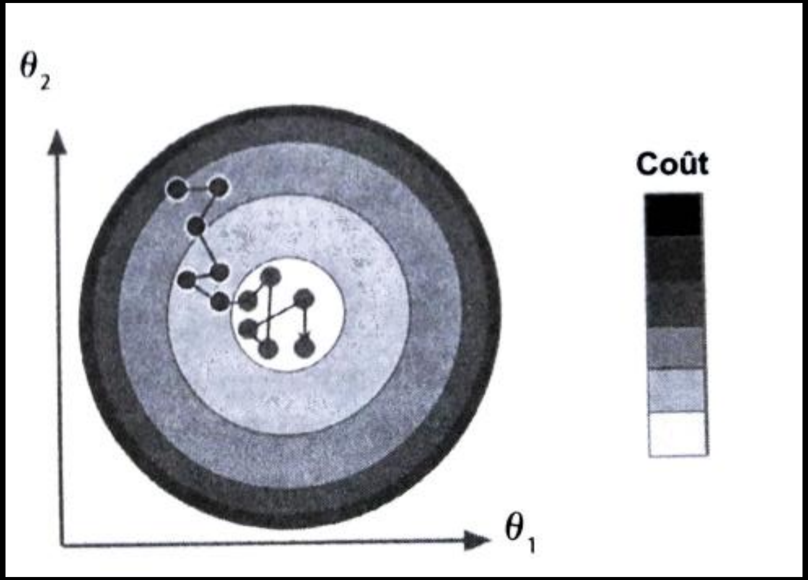
ACTIVATION FUNCTIONS

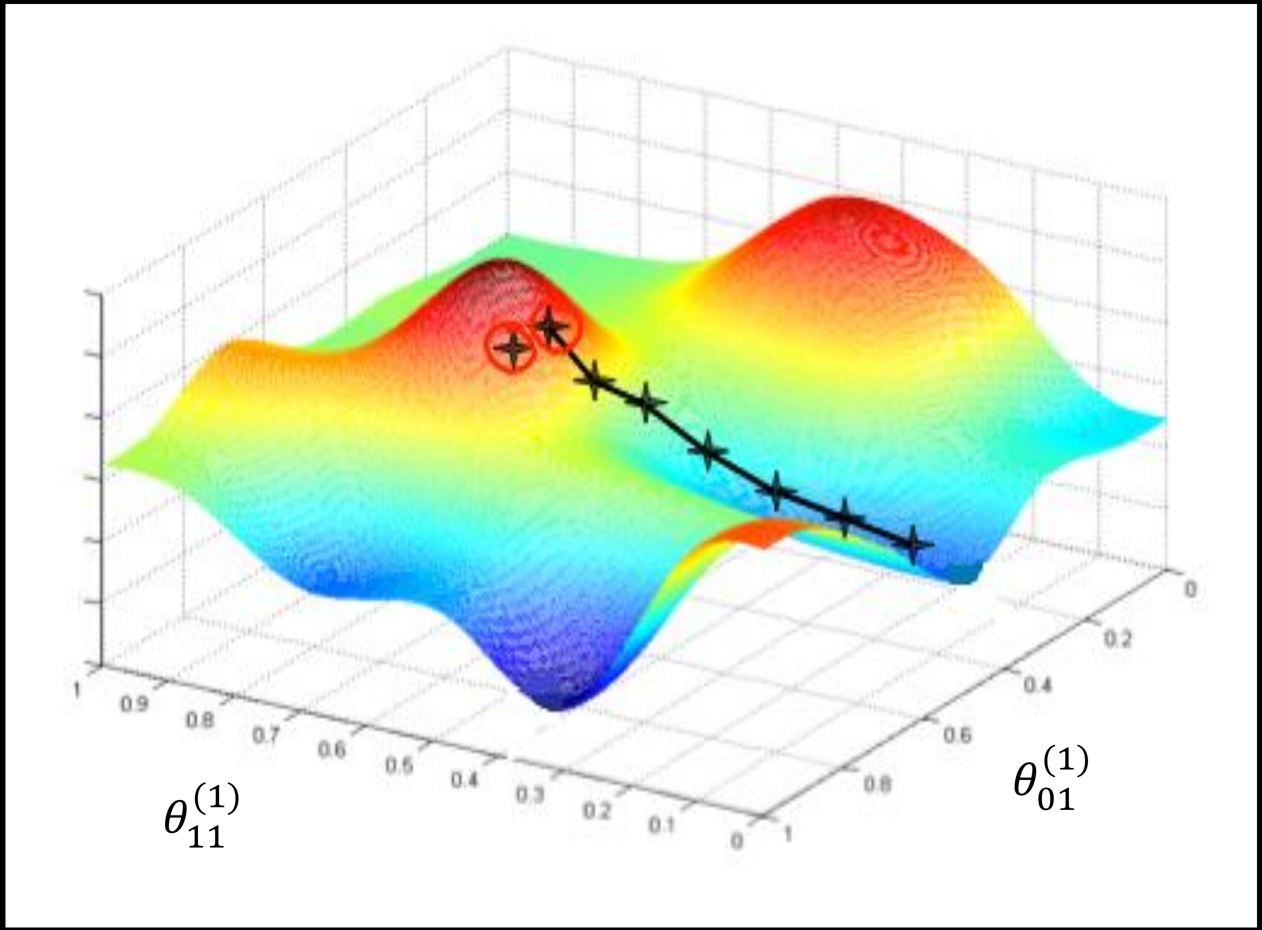
DIFFERENT FUNCTIONS

Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1+e^{-x}}$ [1]	$f(x)(1-f(x))$	$(0,1)$
tanh		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - f(x)^2$	$(-1,1)$
Rectified linear unit (ReLU)[7]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} = \max\{0, x\} = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$

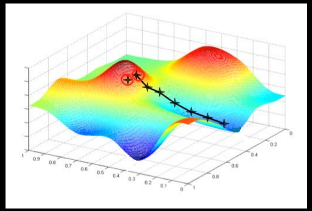
THE SOFTMAX FUNCTION IS RELATED TO THE SIGMOID FUNCTION. IF YOU HAVE 4 CLASSES, THE SOFTMAX FUNCTION WILL CALCULATE A PROBABILITY FOR EACH CLASS AND THE SUM OF THEM WILL BE EQUAL TO 1.

GRADIENT DESCENT

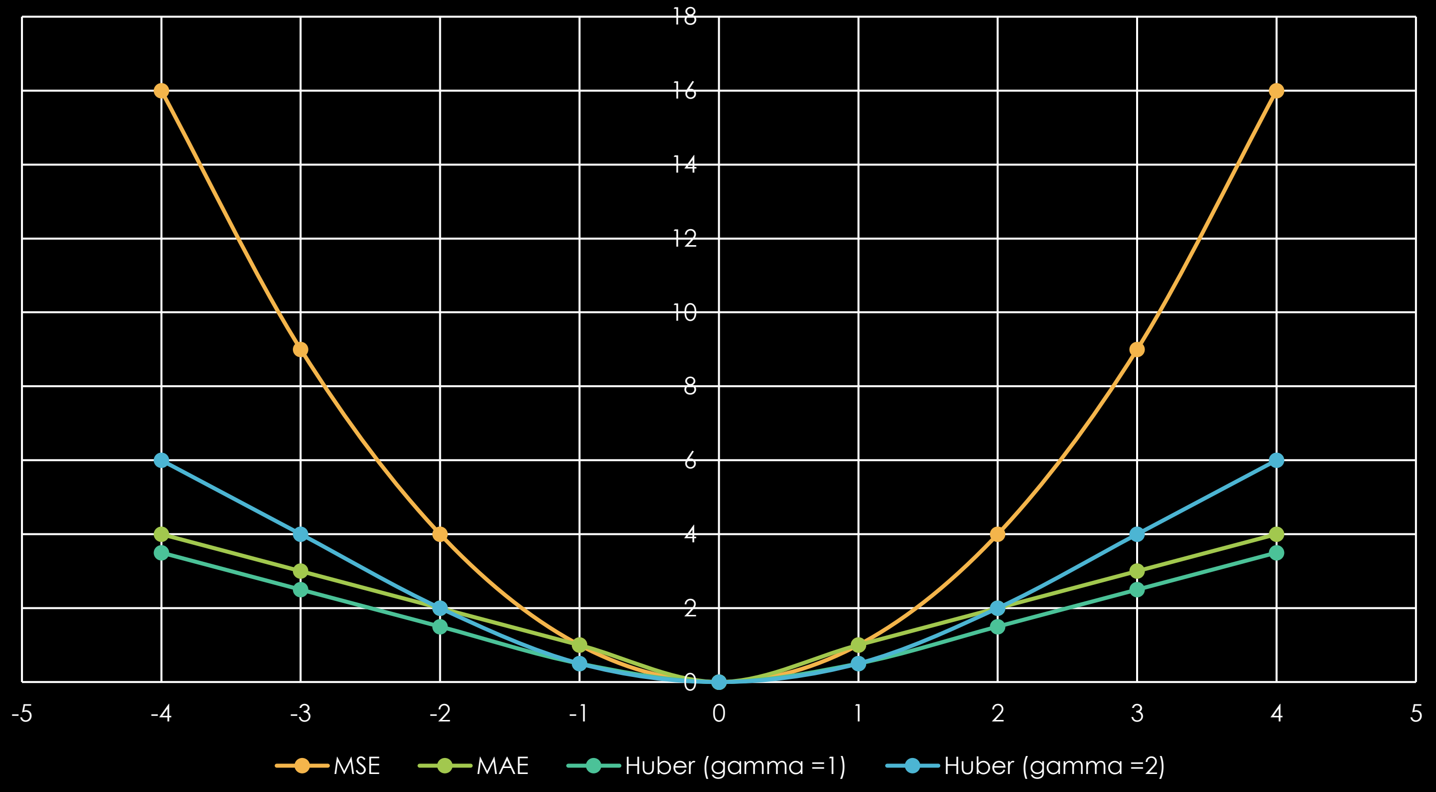




Cost/loss function



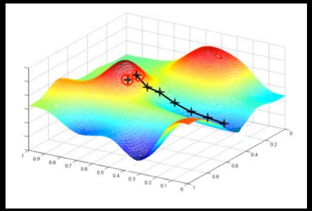
LOSS/COST FUNCTION FOR REGRESSION



$$MAE = \frac{1}{n}\sum_{i=1}^{nsample} \left|y_{obs} - y_{pred}\right|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{nsample} \left(y_{obs} - y_{pred}\right)^2$$

$$Huber = \begin{cases} \frac{1}{2}(y_{obs} - y_{pred})^2 & for\ \left|y_{obs} - y_{pred}\right| \leq \delta \\ \delta\left|y_{obs} - y_{pred}\right| - \frac{1}{2}\delta^2 & otherwise \end{cases}$$



LOSS/COST FUNCTION FOR CLASSIFICATION

Cross-entropy

$$H(p,q) = -\sum_{x} p(x)ln(q(x))$$

p probability distribution

Dataset

Animal	Human
0	1
0	1
1	0
1	0

q probability distribution

Predictions

Animal	Human
0.2	0.8
0.1	0.9
0.6	0.4
0.9	0.1

Average cross-entropy = **0.236**

Cross-entropy per sample

Sample

-(0*ln(0.2)+1*ln(0.8)) = 0.223

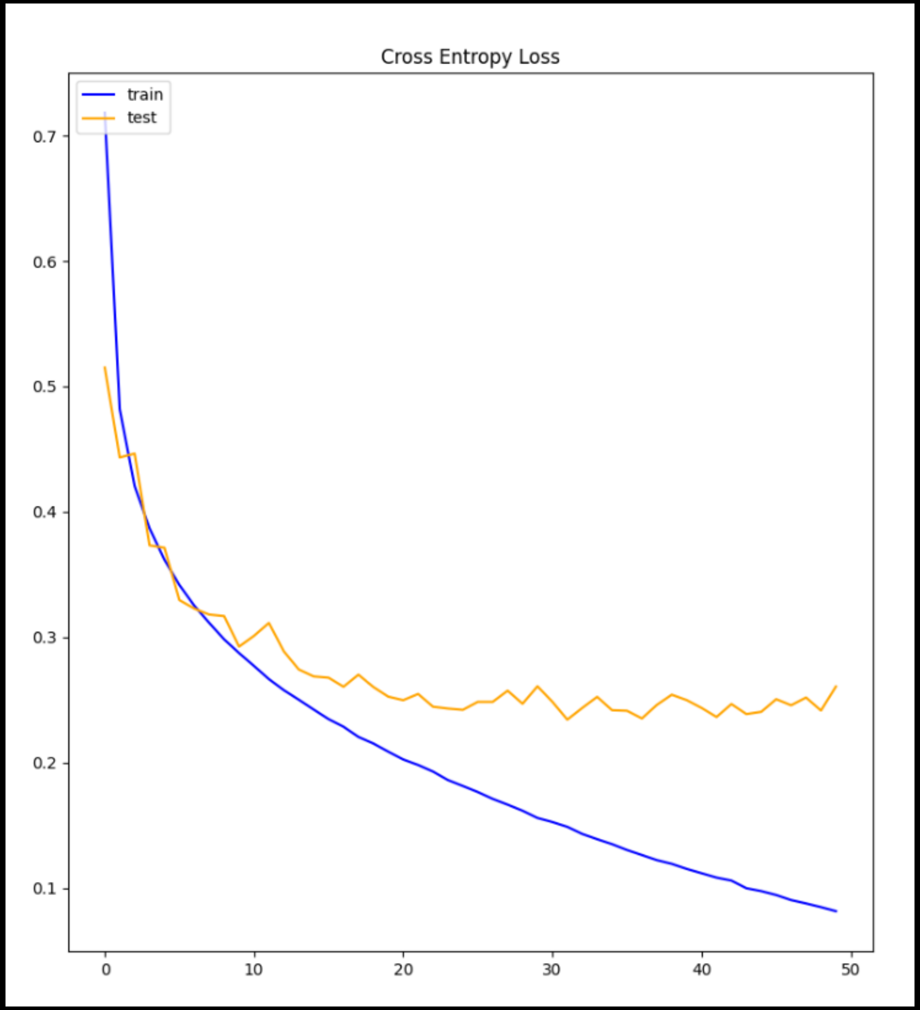
-(0*ln(0.1)+1*ln(0.9)) = 0.105

-(1*ln(0.6)+0*ln(0.4)) = 0.511

-(1*ln(0.9)+0*ln(0.1)) = 0.105

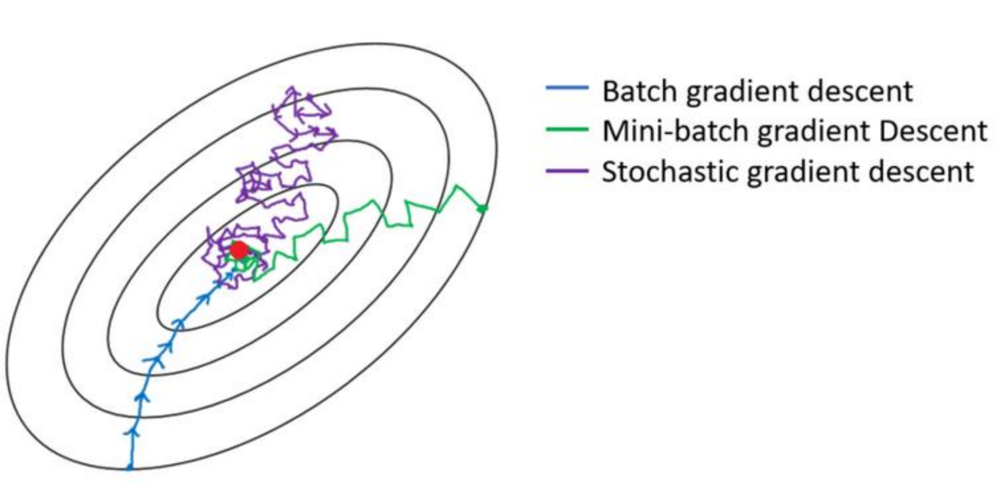
CROSS-ENTROPY

What is your conclusion ?

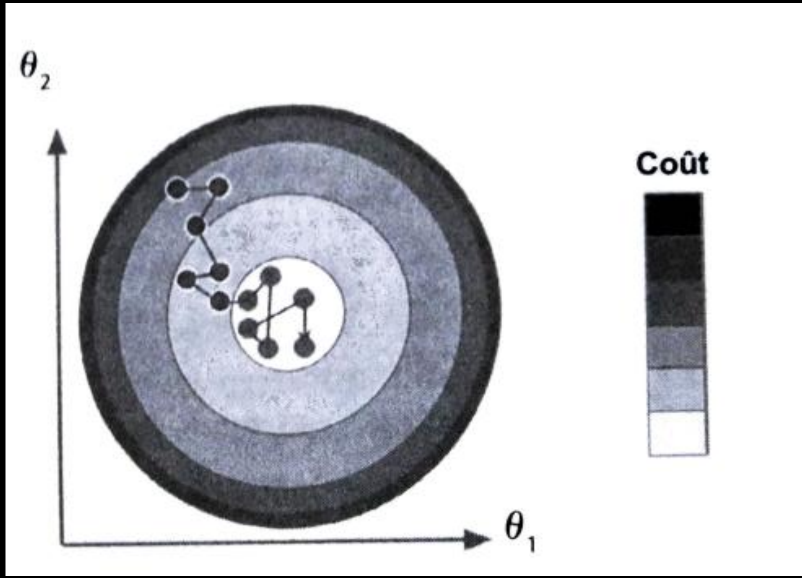


Epochs = iteration

OPTIMIZER



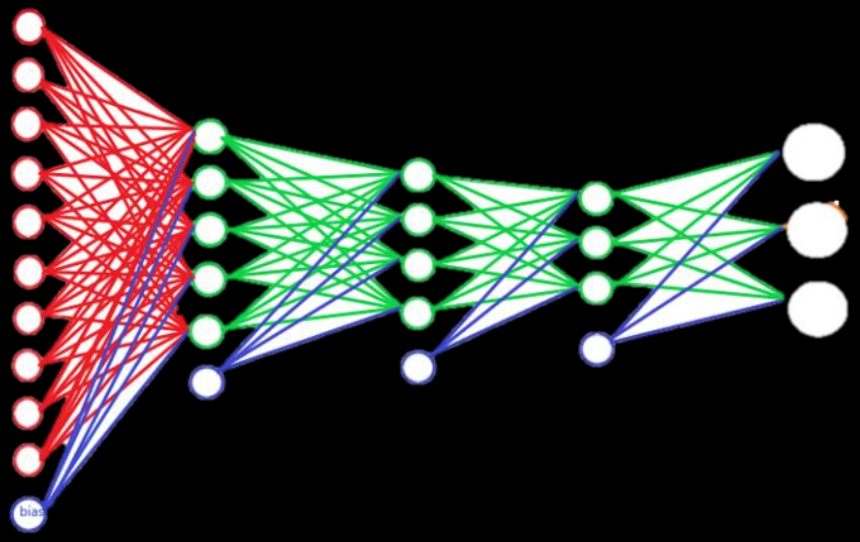
- Ordinary/Batch gradient descent
  - Seen in the last course
  - All records
- Stochastic gradient descent
  - Random selected record
- Mini-batch gradient descent
  - Batch of usually 32 records



Do not reach the optimal value but are close

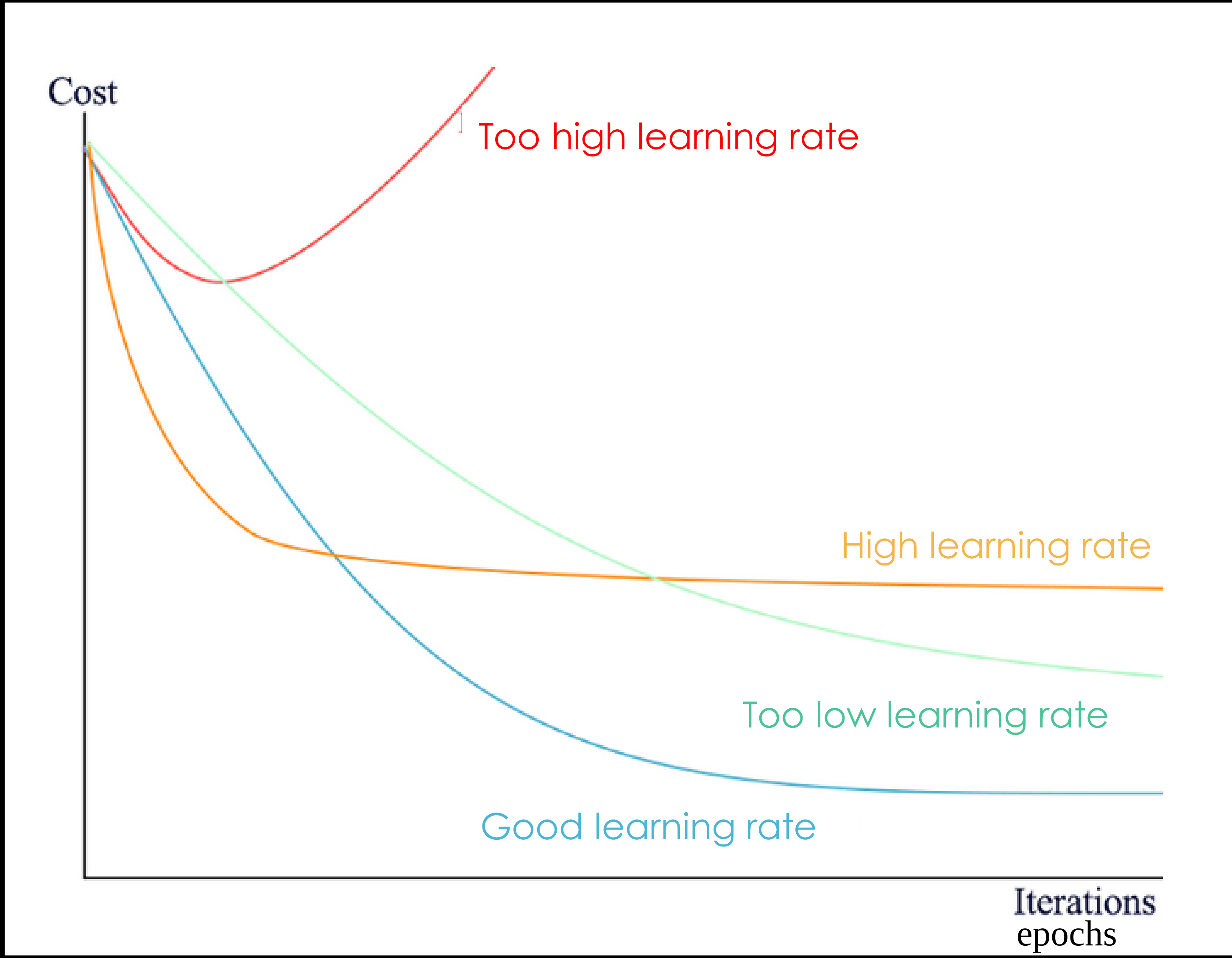
Others exist … (adam…)

MULTILAYER PERCEPTRON



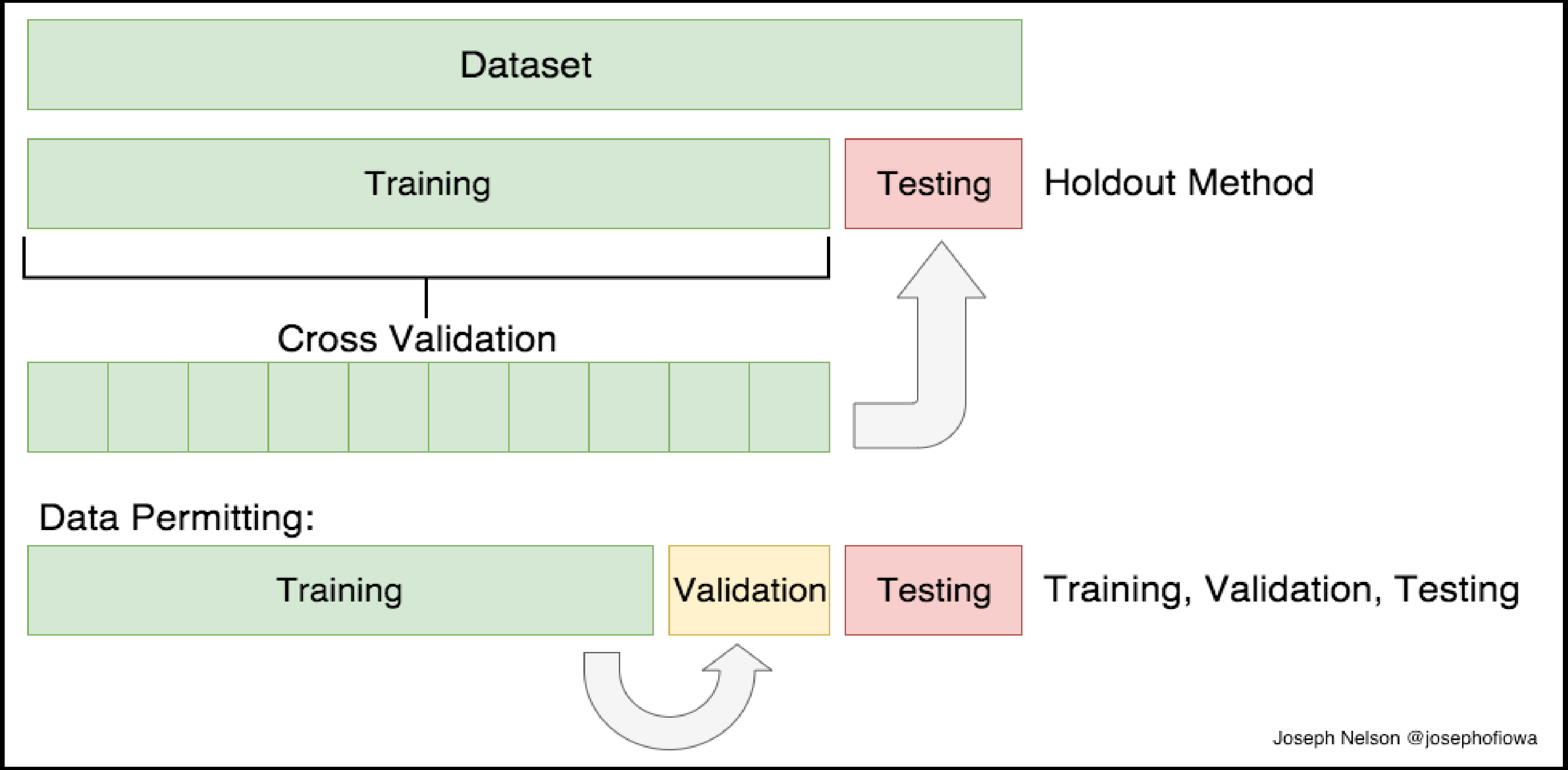
- Higher neurons ➔ ↑ complexity
- Higher hidden layers ➔ ↑ complexity
- ≠ results if other activation function
- ≠ results if other optimizer (more or less impacted by local minima)
- ≠ results if other loss function
- ≠ results if the learning rate is different
- You can also add a regularisation (L2 or L1 norm)

Take care to the overfitting

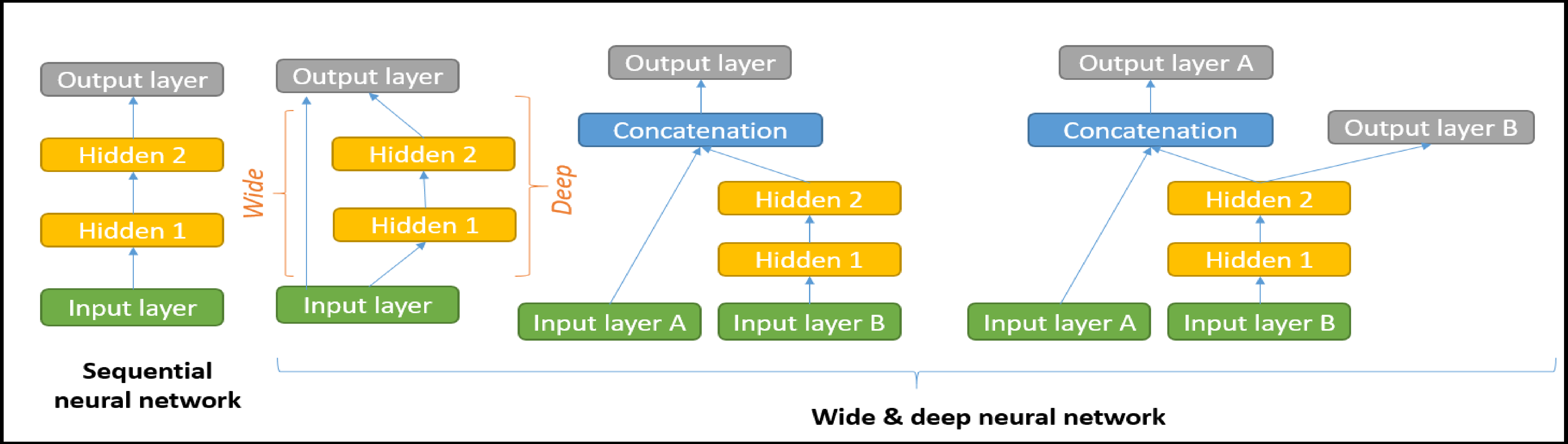


https://jonathan-hui.medium.com/visualize-deep-network-models-and-metrics-part-4-9500fe06e3d0

HYPERPARAMETERS OPTIMIZATION



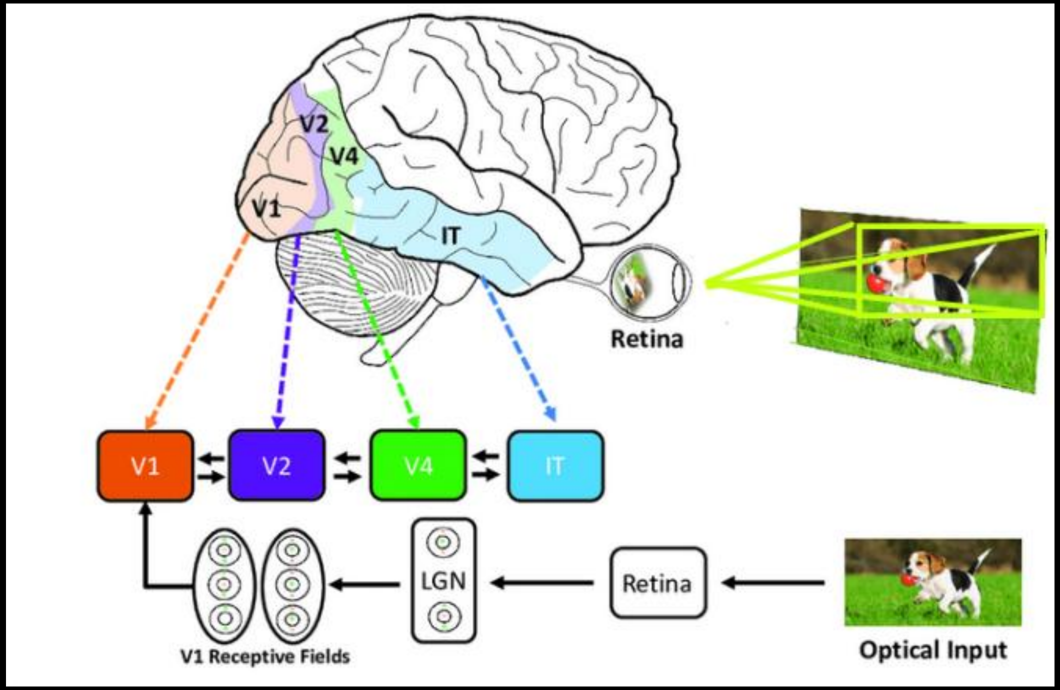
WIDE & DEEP NEURAL NETWORK



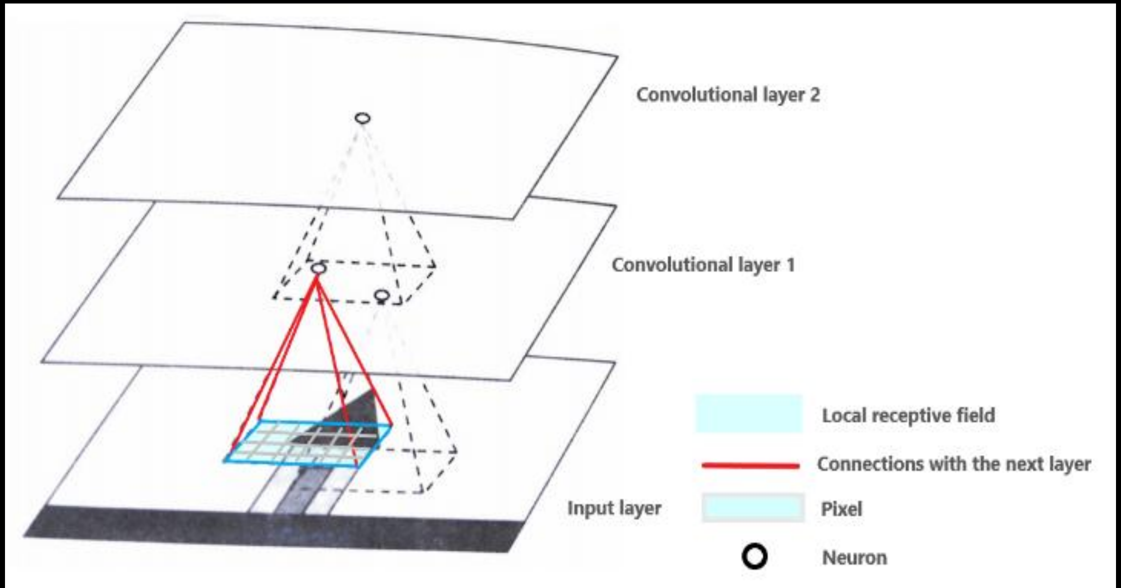
CONVOLUTIONAL NEURAL NETWORK (CANN)

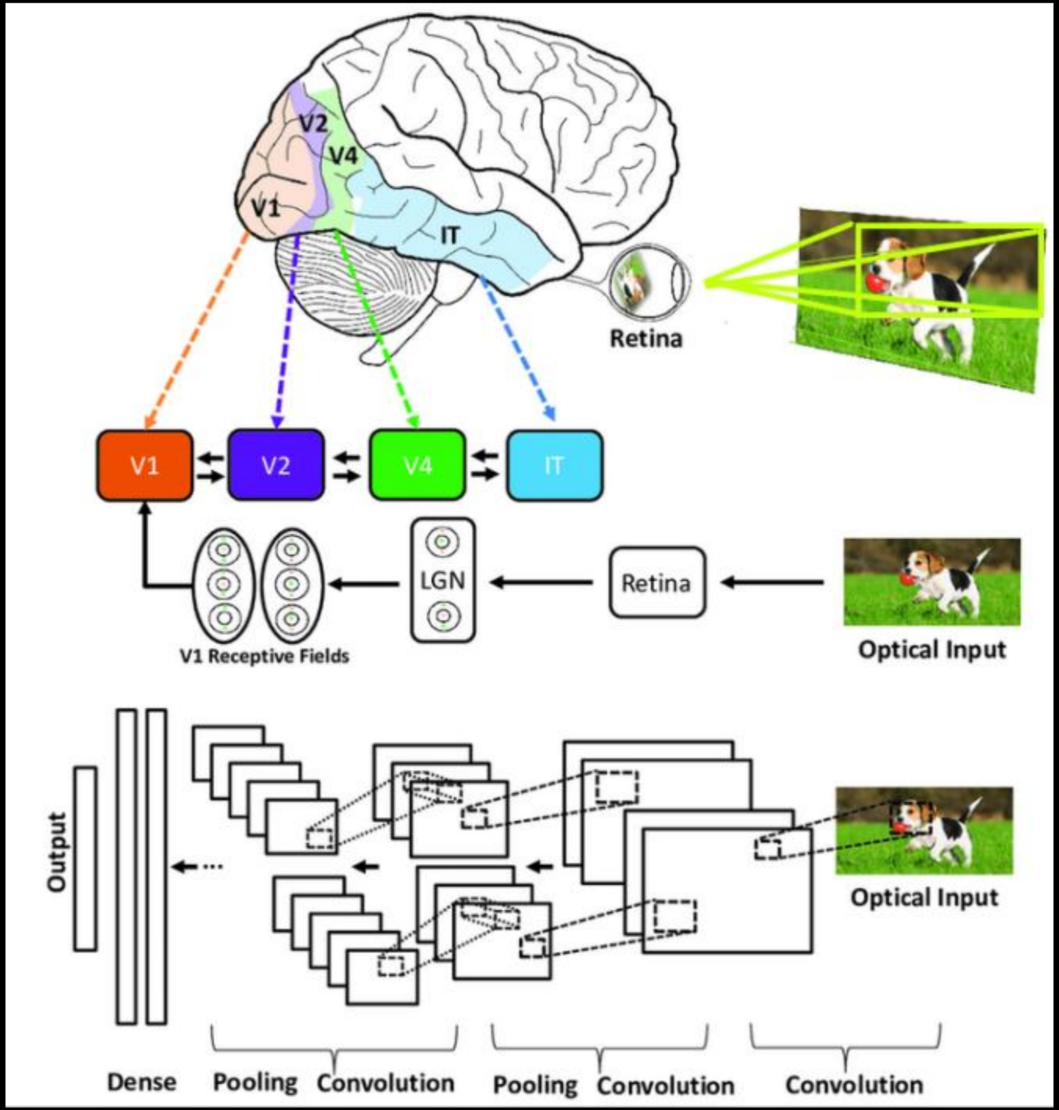
DAVID H. HUBEL AND TORSTEN WIESEL IN THE 50'S OBSERVED THAT SEVERAL NEURONS IN THE VISUAL CORTEX OF CATS IN THE BRAIN FOCUS ON A RESTRICTED REGION OF THE VISUAL FIELD AND INTERACT ONLY IF A VISUAL STIMULI OCCURRED IN THIS REGION.

➔ No interest to link all neurons
➔ Local receptive field (LRF)

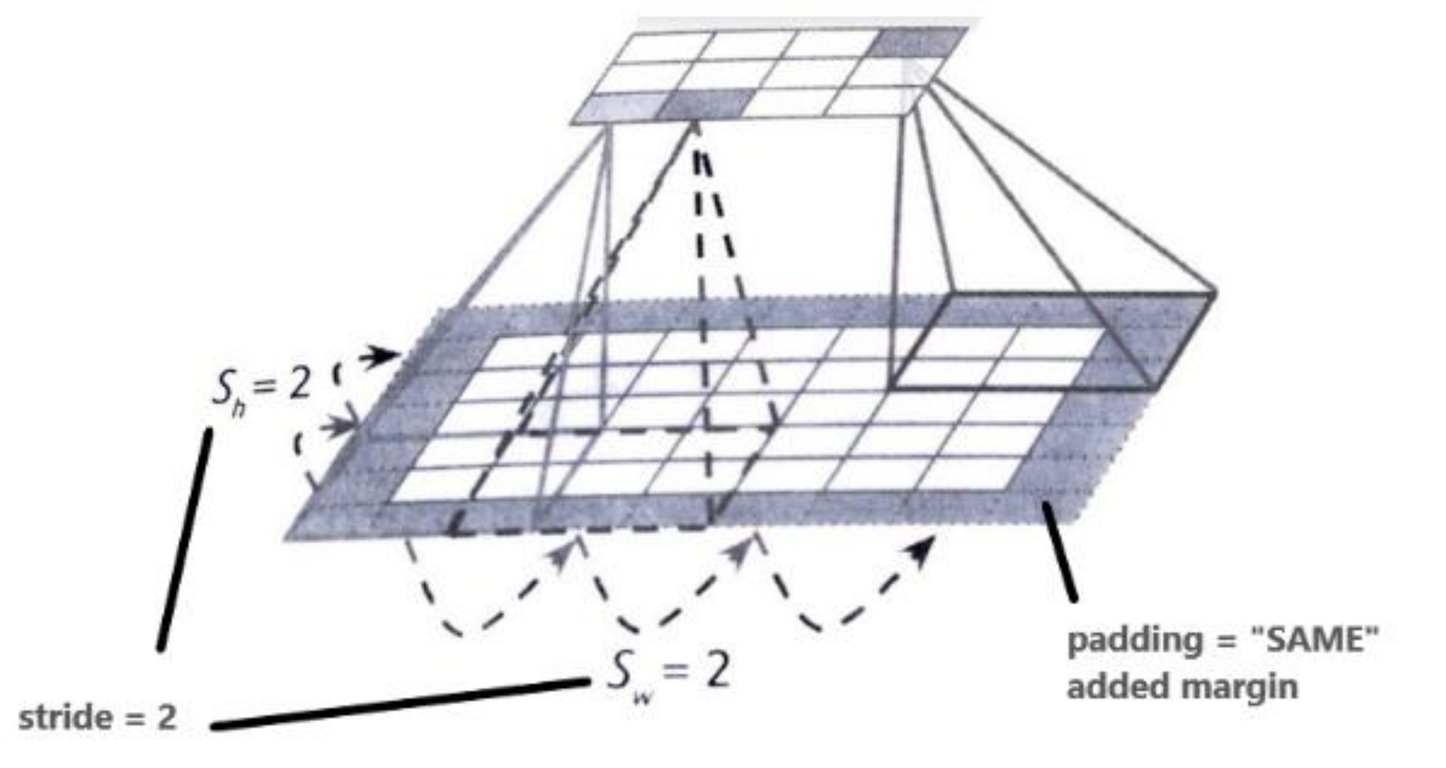


LOCAL RECEPTIVE FIELD (LRF)

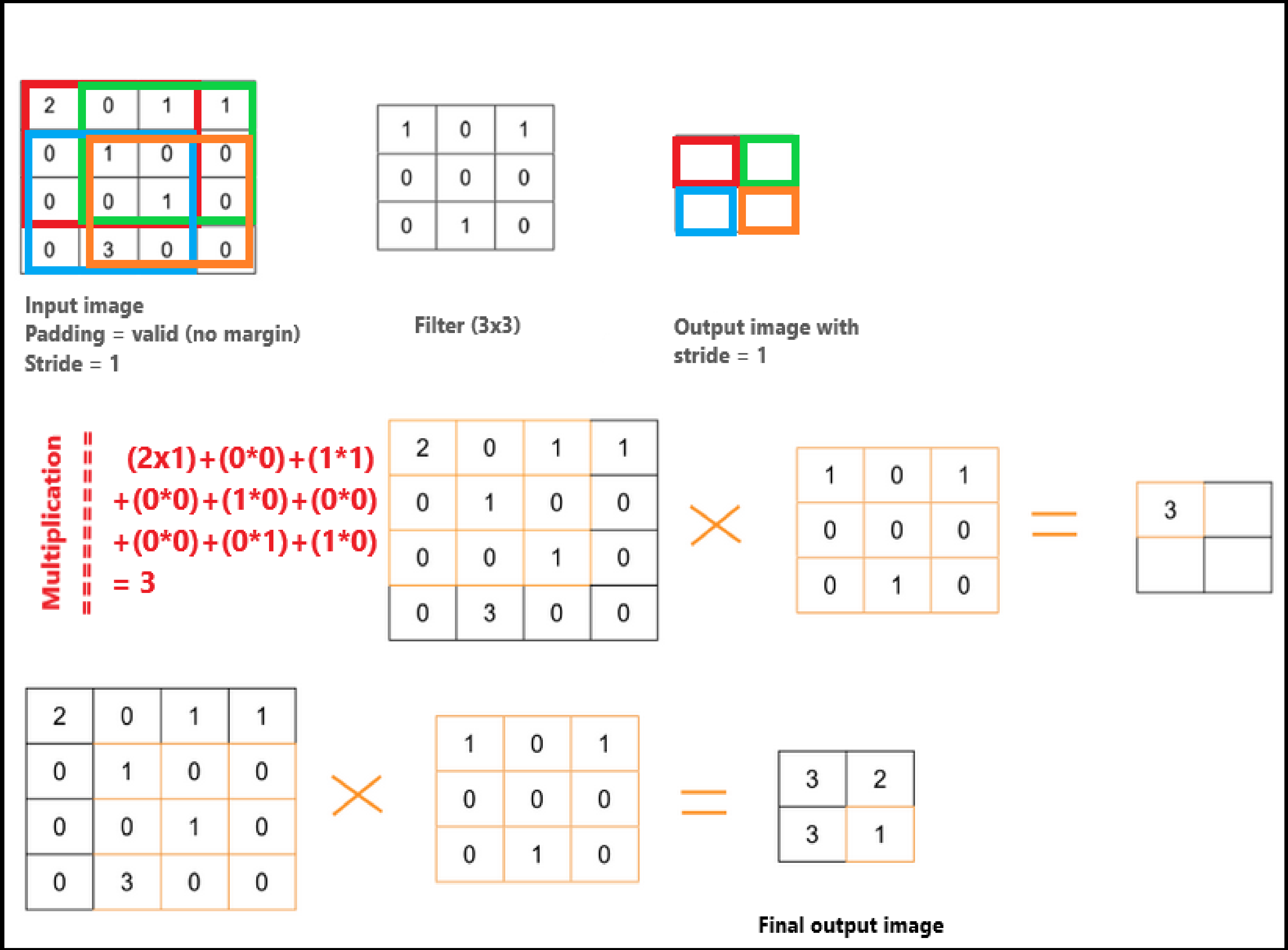




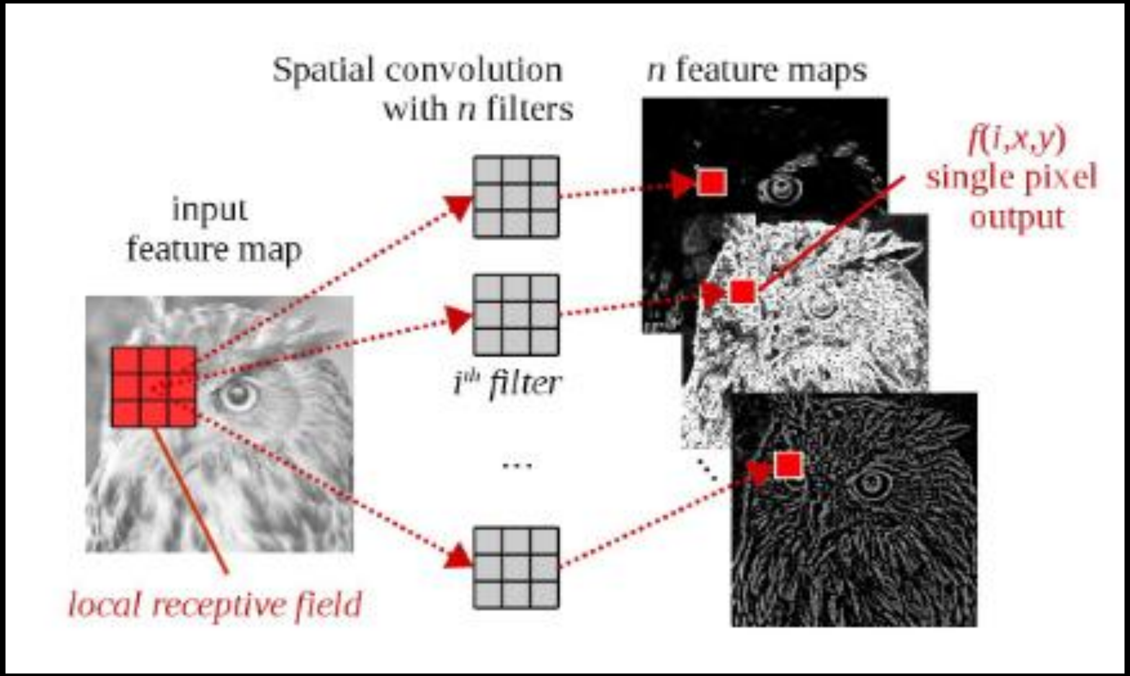
PADDING AND STRIDE

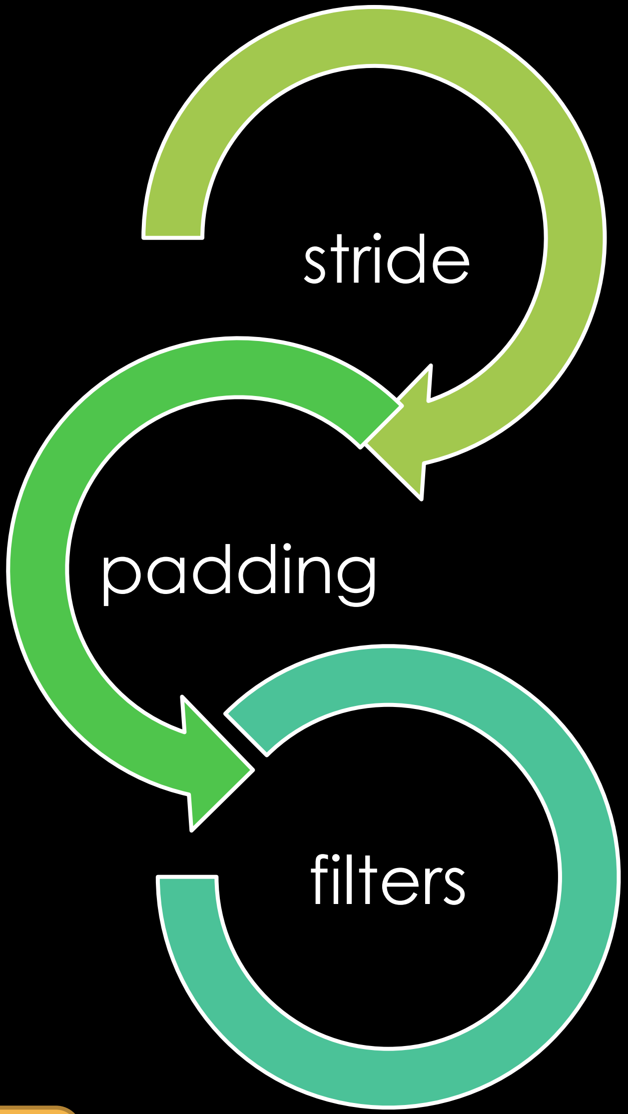


FILTER


Input image
Padding = valid (no margin)
Stride = 1
Filter (3x3)
Output image with
stride = 1
Multiplication
(2x1)+(0*0)+(1*1)
+(0*0)+(1*0)+(0*0)
+(0*0)+(0*1)+(1*0)
= 3
Final output image

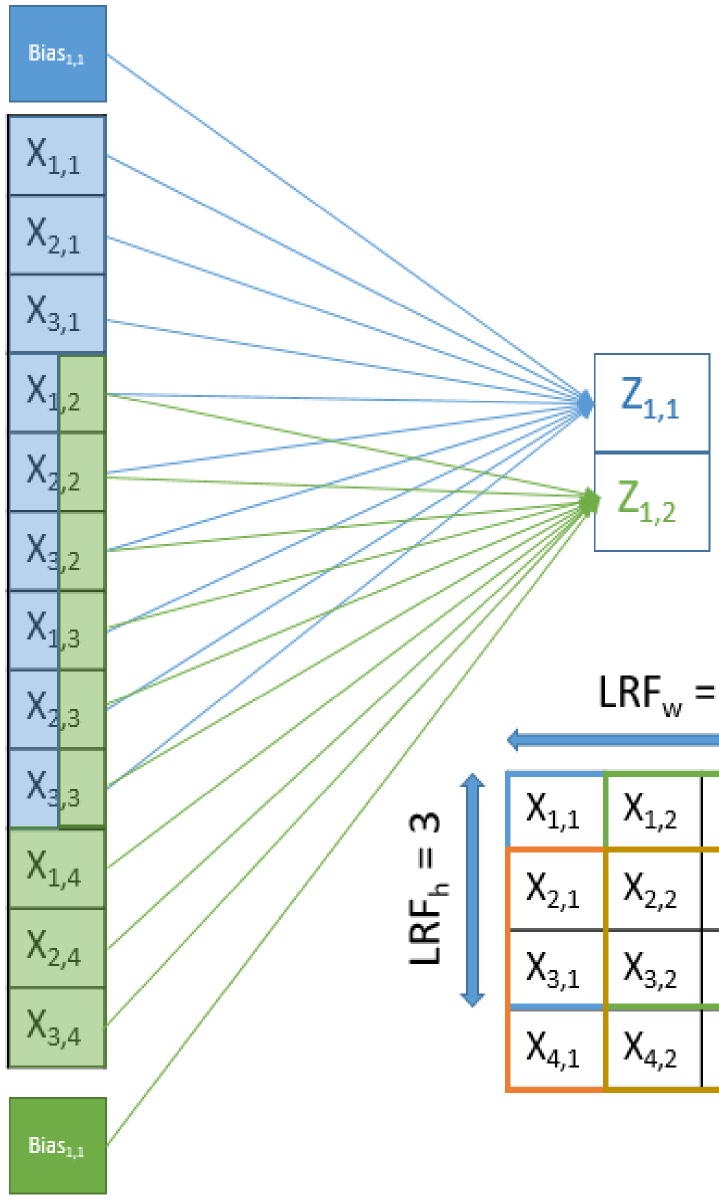
FEATURE MAPS





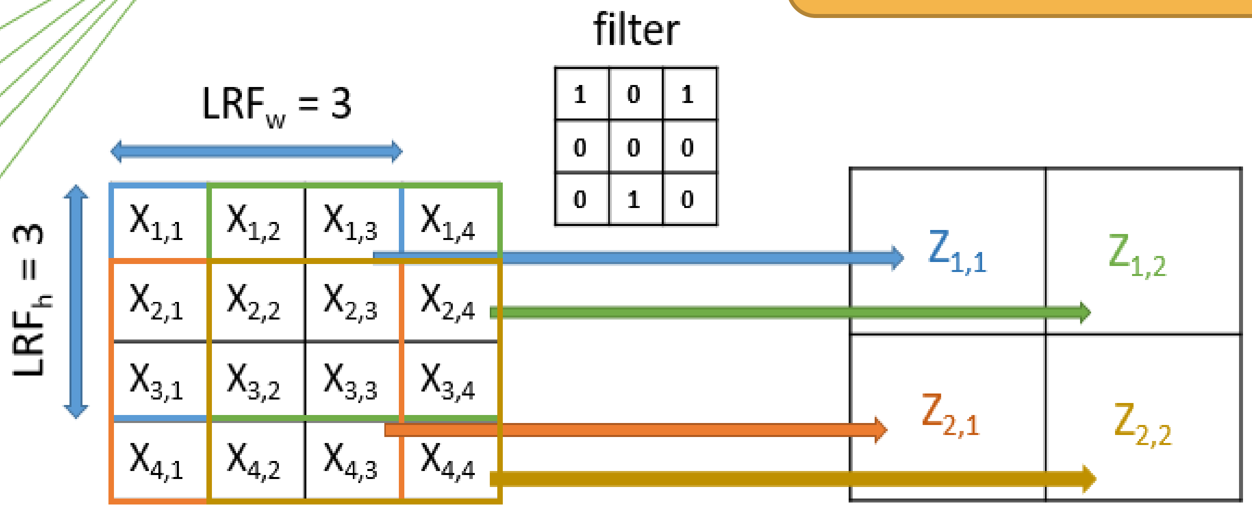
Where are the weights ?

FEATURE MAP

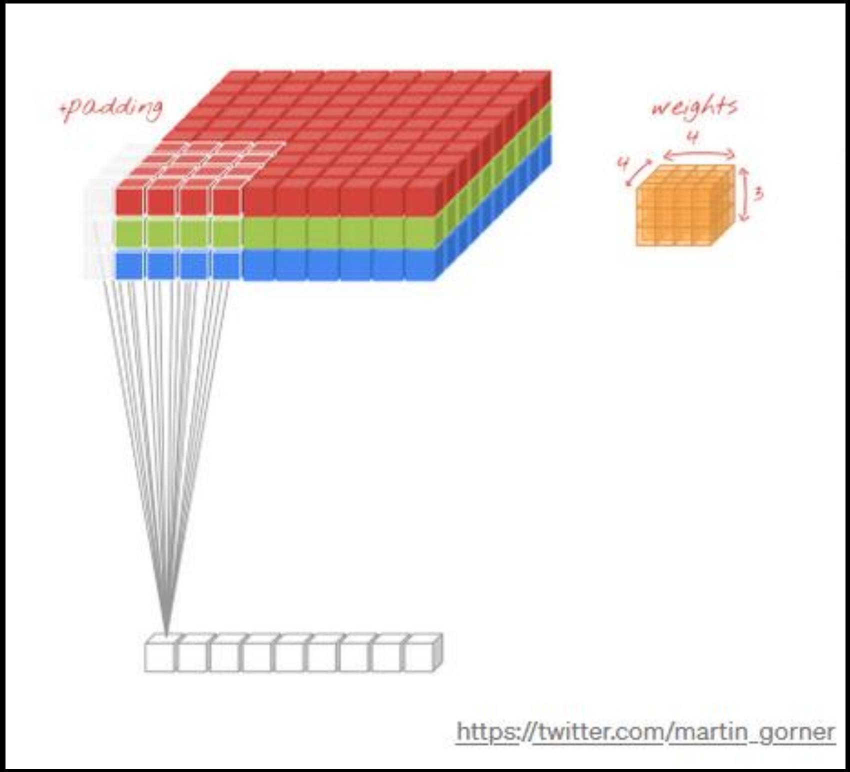


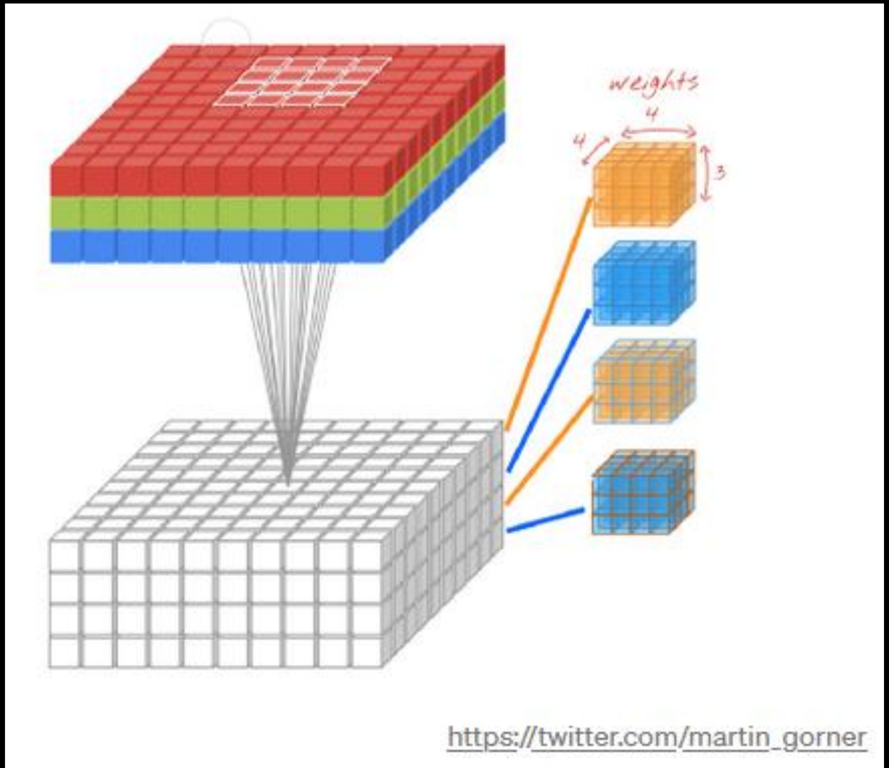
$$z_{1,1} = (f_{1,1} * w_{1,1} * x_{1,1}) + (f_{1,2} * w_{1,2} * x_{1,2}) + (f_{1,3} * w_{1,3} * x_{1,3})$$
$$+ (f_{2,1} * w_{2,1} * x_{2,1}) + (f_{2,2} * w_{2,2} * x_{2,2}) + (f_{2,3} * w_{2,3} * x_{2,3})$$
$$+ (f_{3,1} * w_{3,1} * x_{3,1}) + (f_{3,2} * w_{3,2} * x_{3,2}) + (f_{3,3} * w_{3,3} * x_{3,3})$$
$$+ bias_{1,1}$$

Not all neurons are connected



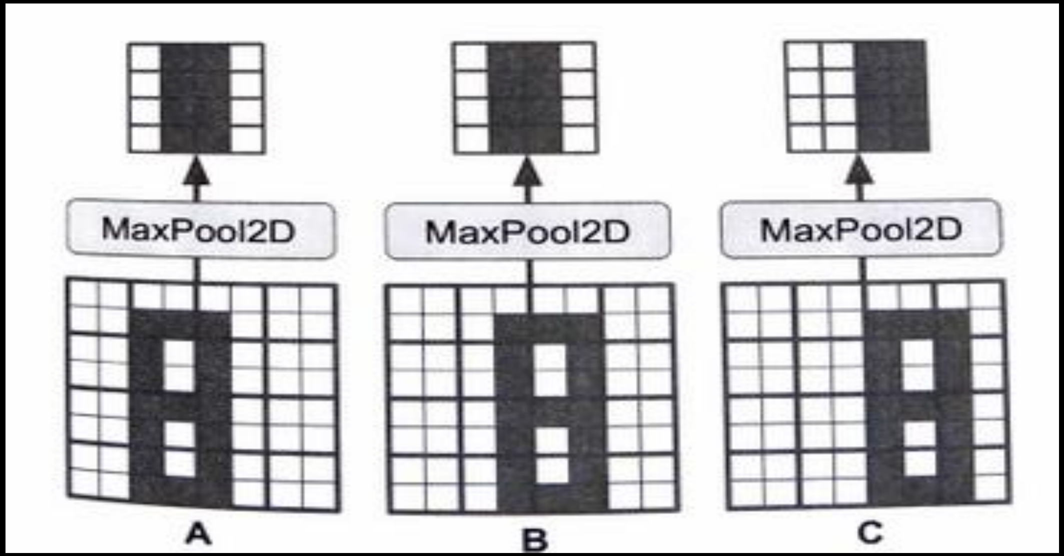
2D AND 3D CANN ➔ FEATURE MAPS



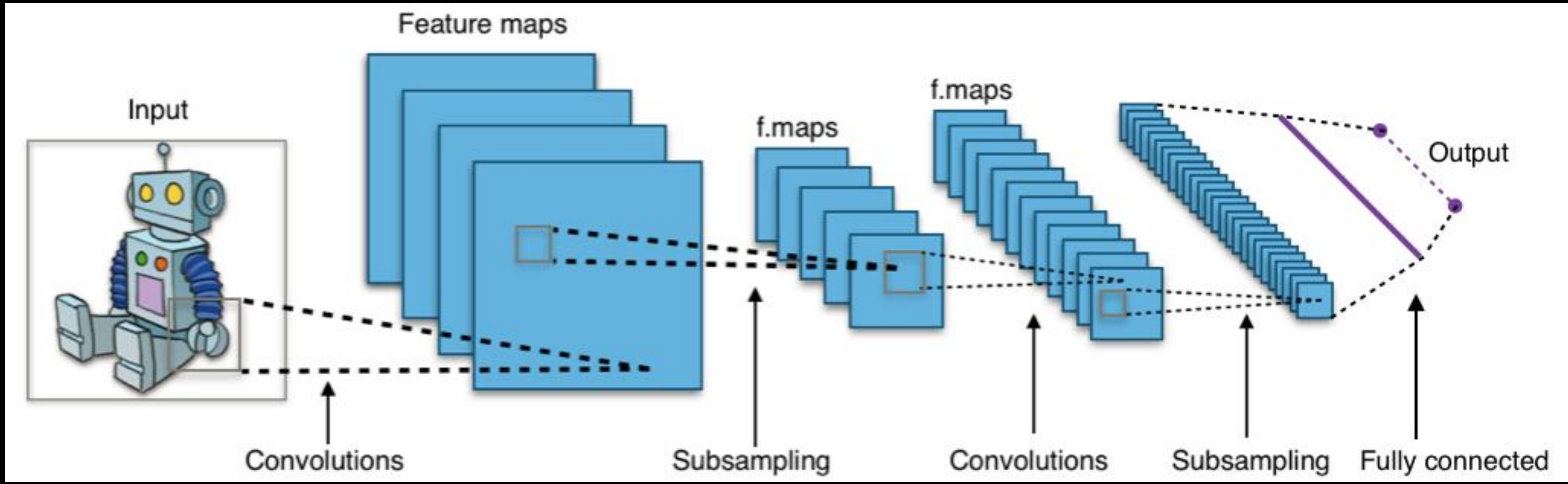


POOLING LAYER

- DECREASE THE DIMENSIONALITY OF THE LAYER BY SUB-SETTING THE IMAGE
- 2*2 IS PREFERRED
- MAX OR MEAN IS USED



CANN


Feature maps
Input
f.maps
f.maps
Output
Convolutions
Subsampling
Convolutions
Subsampling
Fully connected

CURRENT CONCLUSIONS FOR DAIRY FARMING

THE COMPLEXITY OF MODELS CHANGES

- Structure
- Number of parameters

Models already tested in milk spectroscopic analysis

Ordinary multivariate regression	Penalized linear Regression (Ridge, Lasso, Elastic)	Principal Component regression	Partial Least Squares Regression	Random Forest	Neural Network – Multilayer perceptron	Convolutional neural network
				Support vector machine		

Linear until non-linear relationships

Higher computational resources

Table 1. The 10-fold cross-validation and external-validation performances for predicting lactoferrin content in milk using 4 different machine learning algorithms[1]

		PLSR	PLS + Linear SVR	PLS + Polynomial SVR	PLS + ANN
Selection function		oneSE	oneSE	best	best = oneSE[2]
Calibration (n = 5,541)	Parameters	nLV[3] = 23	C[4] = 5	degree = 3; scale = 0.01; C = 1	size = 4; decay = 0.5
	R^2c	0.53	0.53	0.64	0.60
	RMSEc	140.94	144.32	125.89	130.59
Cross-validation	R^2cv	0.51	0.53	0.56	0.55
	R^2cv SD	0.03	0.03	0.03	0.03
	RMSEcv	144.31	144.60	138.40	139.01
	RMSEcv SD	5.77	5.61	8.08	5.05
	RPD	1.43	1.42	1.49	1.48
External validation (n = 836)	R^2v	0.61	0.63	0.62	0.60
	RMSEv	163.76	174.92	166.75	162.17

Soyeurt et al., JDS 2020

Journal of Dairy Science
Volume 103, Issue 12, December 2020, Pages 11585-11596

Research

A comparison of 4 different machine learning algorithms to predict lactoferrin content in bovine milk from mid-infrared spectra

H. Soyeurt[1], C. Grelet[2], S. McParland[3], M. Calmels[4], M. Coffey[5], A. Tedde[1], P. Delhez[1,6], F. Dehareng[2], N. Gengler[1]

Example : **Lactoferrin**

Mean : 260 mg/L

|Difference| $_{\text{PLS vs ANN}}$: 1.59 mg/L → 0.61%

|Difference| $_{\text{PLS vs LSVM}}$: 11.16 mg/L → 4.29%

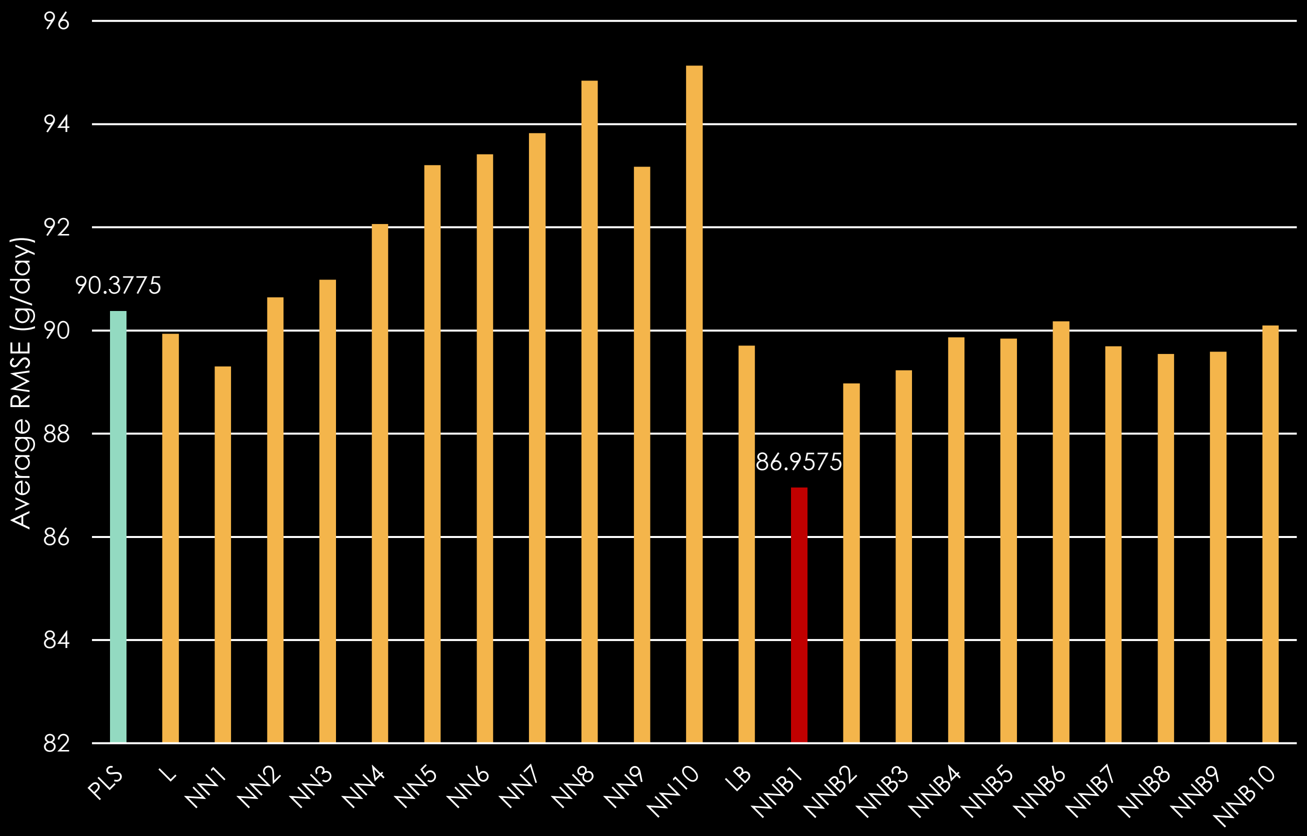
|Difference| $_{\text{PLS vs PSVM}}$: 2.99 mg/L → 1.15%

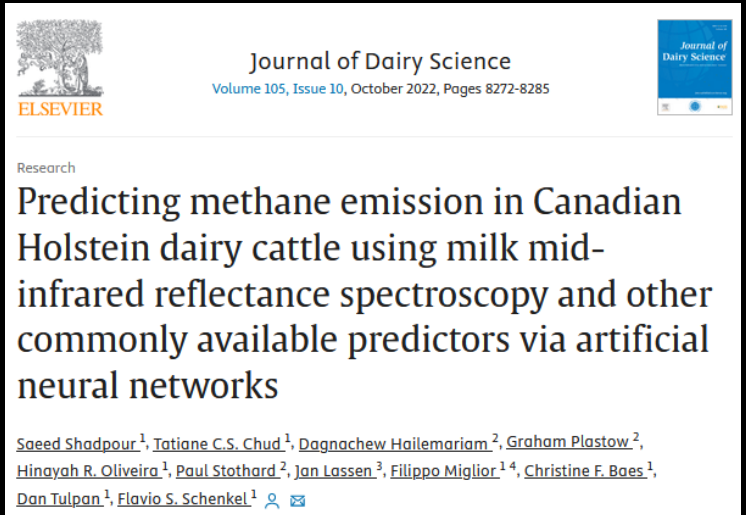
|Difference| $_{\text{ANN vs LSVM}}$: 12.75 mg/L → 4.90%

|Difference| $_{\text{ANN vs PSVM}}$: 4.58 mg/L → 1.76%

|Difference| $_{\text{PSVM vs LSVM}}$: 8.17 mg/L → 3.14%

Small differences !




ELSEVIER
Journal of Dairy Science
Volume 105, Issue 10, October 2022, Pages 8272-8285

Research

Predicting methane emission in Canadian Holstein dairy cattle using milk mid-infrared reflectance spectroscopy and other commonly available predictors via artificial neural networks

Saeed Shadpour [1], Tatiane C.S. Chud [1], Dagnachew Hailemariam [2], Graham Plastow [2], Hinayah R. Oliveira [1], Paul Stothard [2], Jan Lassen [3], Filippo Miglior [1 4], Christine F. Baes [1], Dan Tulpan [1], Flavio S. Schenkel [1]

Example : **Methane**

Mean : +/- 400g/jour

| Difference | $_{\text{PLS vs NN}}$: 3g/day ➔ 0.75%

Method	Predictor	Statistic											
		Bias	RMSE	r	b	RPIQ	Method		Bias	RMSE	r	b	RPIQ
Partial least squares	Previous						**Neural networks**	Previous					
	a.m.	−0.23 (8.97)	43.04 (2.36)	0.64 (0.03)	0.94 (0.11)	1.76 (0.16)		a.m.	0.12 (8.66)	42.50 (1.64)	0.64 (0.02)	0.90 (0.11)	1.76 (0.16)
	p.m.	0.03 (9.49)	43.73 (2.48)	0.63 (0.03)	0.93 (0.11)	1.74 (0.16)		p.m.	−0.11 (10.00)	43.15 (2.04)	0.64 (0.03)	0.87 (0.10)	1.73 (0.18)
	a.m. and p.m.	−0.01 (8.89)	42.15 (1.52)	0.65 (0.03)	0.94 (0.10)	1.77 (0.17)		a.m. and p.m.	−0.42 (8.26)	41.14 (1.86)	0.67 (0.02)	0.90 (0.09)	1.81 (0.16)
	Average a.m. and p.m.	−0.09 (8.81)	41.76 (1.98)	0.66 (0.03)	0.95 (0.11)	1.79 (0.18)		Average a.m. and p.m.	−0.42 (7.96)	41.45 (1.45)	0.66 (0.03)	0.91 (0.12)	1.80 (0.15)
	Following							Following					
	a.m.	−0.49 (9.34)	44.62 (1.46)	0.61 (0.02)	0.95 (0.09)	1.71 (0.11)		a.m.	−0.73 (8.93)	43.59 (1.42)	0.64 (0.02)	0.92 (0.11)	1.75 (0.11)
	p.m.	−0.06 (9.93)	45.23 (0.90)	0.60 (0.04)	0.93 (0.12)	1.69 (0.12)		p.m.	−0.36 (9.08)	44.36 (0.37)	0.62 (0.03)	0.89 (0.07)	1.72 (0.11)
	a.m. and p.m.	−0.20 (9.14)	44.14 (0.76)	0.62 (0.03)	0.94 (0.10)	1.73 (0.11)		a.m. and p.m.	−0.96 (8.84)	42.48 (1.02)	0.66 (0.03)	0.89 (0.08)	1.80 (0.13)
	Average a.m. and p.m.	−0.56 (9.43)	44.00 (1.40)	0.63 (0.03)	0.95 (0.10)	1.74 (0.11)		Average a.m. and p.m.	−0.46 (8.23)	42.33 (1.38)	0.66 (0.03)	0.92 (0.09)	1.81 (0.13)
	Flanking							Flanking					
	a.m.	−0.42 (9.08)	38.09 (1.87)	0.68 (0.03)	0.95 (0.11)	1.87 (0.17)		a.m.	0.16 (8.97)	38.23 (1.50)	0.68 (0.03)	0.92 (0.12)	1.86 (0.17)
	p.m.	−0.11 (9.71)	39.25 (1.57)	0.66 (0.03)	0.93 (0.12)	1.81 (0.15)		p.m.	−0.46 (9.27)	39.16 (0.94)	0.66 (0.02)	0.87 (0.07)	1.82 (0.14)
	a.m. and p.m.	−0.10 (8.83)	37.98 (1.36)	0.68 (0.03)	0.95 (0.11)	1.87 (0.16)		a.m. and p.m.	−0.77 (8.43)	37.17 (1.53)	0.70 (0.02)	0.91 (0.09)	1.91 (0.15)
	Average a.m. and p.m.	−0.25 (8.92)	37.56 (2.00)	0.69 (0.03)	0.96 (0.11)	1.90 (0.16)		Average a.m. and p.m.	−0.33 (7.95)	37.46 (4.01)	0.71 (0.03)	0.92 (0.11)	1.95 (0.16)

Example : **Methane**

Also low differences

ELSEVIER
Journal of Dairy Science
Volume 107, Issue 2, February 2024, Pages 978-991

Research

Predicting methane emissions of individual grazing dairy cows from spectral analyses of their milk samples

S. McParland, M. Frizzarin, B. Lahart, M. Kennedy, L. Shalloo, M. Egan, K. Starsmore, D.P. Berry

Open Access Article

Identifying Health Status in Grazing Dairy Cows from Milk Mid-Infrared Spectroscopy by Using Machine Learning Methods

by Brenda Contla Hernández [1], Nicolas Lopez-Villalobos [2] and Matthieu Vignes [1,*]

[1] School of Fundamental Sciences, Massey University, Palmerston North 4442, New Zealand
[2] School of Agriculture and Environment, Massey University, Palmerston North 4442, New Zealand
* Author to whom correspondence should be addressed.

Animals **2021**, *11*(8), 2154; https://doi.org/10.3390/ani11082154

More nuanced in classification

Table 2. Performance of classification models obtained in 10 Monte Carlo cross-validation for classifying any health problem and healthy cows during lactation (early, mid and lactation) at two dairy farms during the 2016 production season [1].

Models [2]	Sensitivity	Specificity	Accuracy	PPV	NPV	AUC	MCC
PLS-DA	65.60 ± 5.97	79.59 ± 2.36	78.85 ± 2.23	15.25 ± 3.07	97.66 ± 0.5	72.59 ± 3.27	0.24 ± 0.04
RF	46.22 ± 8.62	79.26 ± 2.15	77.51 ± 1.75	10.94 ± 1.88	96.38 ± 0.73	62.74 ± 3.78	0.14 ± 0.04
SVM	66.39 ± 6.80	76.39 ± 2.92	75.84 ± 2.42	13.48 ± 1.62	97.61 ± 0.61	71.39 ± 2.37	0.22 ± 0.02
NN	61.74 ± 15.99	97.00 ± 2.85	95.16 ± 3.26	59.99 ± 26.20	97.87 ± 0.87	79.37 ± 9.16	0.58 ± 0.22
CNN	57.02 ± 12.70	92.5 ± 5.27	90.63 ± 4.98	33.82 ± 13.41	97.5 ± 0.75	74.76 ± 6.88	0.39 ± 0.13
ESA	57.15 ± 12.38	87.61 ± 6.19	86.02 ± 6.21	24.06 ± 13.07	97.36 ± 0.77	72.38 ± 8.48	0.31 ± 0.16
ESMJ	60.75 ± 5.98	83.57 ± 2.56	82.36 ± 2.27	17.18 ± 3.21	97.46 ± 0.55	72.16 ± 2.9	0.25 ± 0.04
ESWA	56.43 ± 14.56	85.13 ± 7.41	83.61 ± 7.36	21.33 ± 14.18	97.22 ± 0.97	70.78 ± 9.71	0.27 ± 0.17

[1] These values correspond to the mean ± SD obtained by 10-fold Monte Carlo cross-validation for classifying any health problem (lameness, mastitis, reproductive disorder, etc.). From the cows' records, the positive cases were cows that had any illness (lameness, mastitis, reproductive disorder, etc.) and negative cases were cows who were healthy (no diagnosed disease); SD = Standard deviation; PPV = positive predicted value; NPV = negative predicted value; AUC = area under the receiver operating characteristic curve; MCC = Matthews correlation coefficient. [2] Models used to perform the classification: PLS-DA = partial least squares discriminant analysis, RF = random forest, SVM = support vector machine, NN = neural network, CNN = convolutional neural network, ESA = ensemble stacking average, ESMJ = ensemble stacking major voting and ESWA = ensemble stacking weighted average.

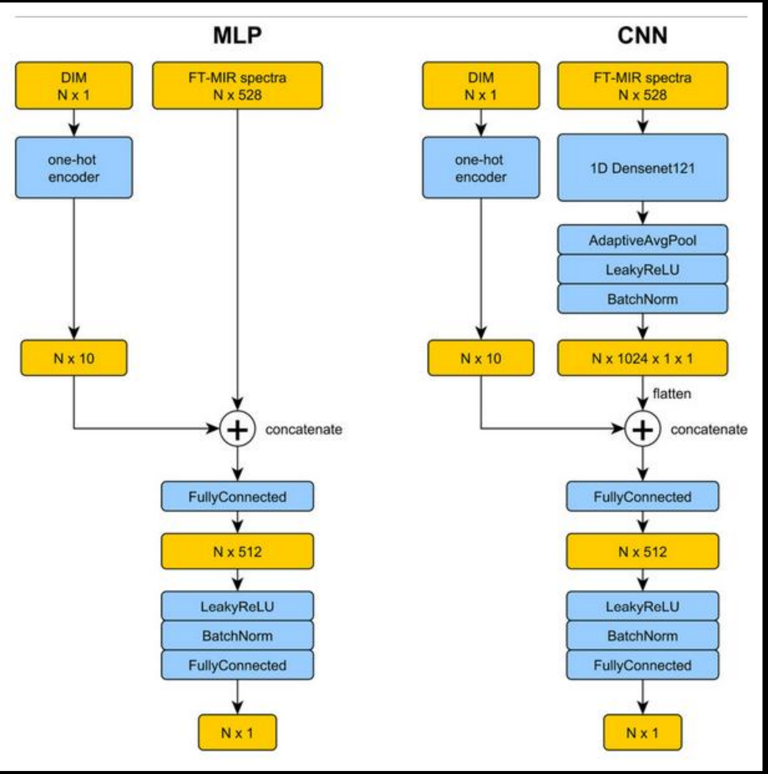
ELSEVIER

Journal of Dairy Science
Volume 105, Issue 4, April 2022, Pages 3615-3632

Research

Pregnancy status predicted using milk mid-infrared spectra from dairy cattle

K.M. Tiplady[1 2], M.-H. Trinh[1], S.R. Davis[1], R.G. Sherlock[1], R.J. Spelman[1], D.J. Garrick[2], B.L. Harris[1]



More nuanced in classification

Table 3. Model performance for multilayer perceptron (MLP) and convolutional neural network (CNN) approaches based on strategy 3 data[1]: accuracy (Acc), sensitivity (Sens), specificity (Spec), and area under the receiver operating characteristic curve (AUC) values within the training, herd-independent validation (VAL-Test) and pregnancy-associated glycoproteins validation (VAL-PAG) data sets

Deep learning approach[2] and model[3]	Training				Test validation (VAL-Test)				Glycoprotein-based validation (VAL-PAG)			
	Acc	Sens	Spec	AUC	Acc	Sens	Spec	AUC	Acc	Sens	Spec	AUC
MLP approach												
FT-MIR spectra	0.592	0.574	0.611	0.628	0.586	0.580	0.607	0.632	0.664	0.672	0.569	0.669
FT-MIR spectra + DIM	0.594	0.621	0.566	0.631	0.614	0.629	0.564	0.635	0.692	0.709	0.499	0.647
FT-MIR spectra (pre-adjusted for DIM)	0.559	0.554	0.564	0.583	0.562	0.567	0.547	0.581	0.554	0.547	0.636	0.636
CNN approach												
FT-MIR spectra	0.625	0.625	0.625	0.675	0.611	0.620	0.582	0.641	0.684	0.696	0.554	0.676
FT-MIR spectra + DIM	0.645	0.670	0.620	0.700	0.636	0.659	0.563	0.654	0.723	0.741	0.519	0.685
FT-MIR spectra (pre-adjusted for DIM)	0.982	0.975	0.988	0.998	0.668	0.790	0.273	0.551	0.759	0.805	0.266	0.564

The variability is bigger within the same model using different sets of features

ELSEVIER

Journal of Dairy Science
Volume 105, Issue 10, October 2022, Pages 8272-8285

Journal of Dairy Science

Research

Predicting methane emission in Canadian Holstein dairy cattle using milk mid-infrared reflectance spectroscopy and other commonly available predictors via artificial neural networks

Saeed Shadpour [1], Tatiane C.S. Chud [1], Dagnachew Hailemariam [2], Graham Plastow [2], Hinayah R. Oliveira [1], Paul Stothard [2], Jan Lassen [3], Filippo Miglior [1 4], Christine F. Baes [1], Dan Tulpan [1], Flavio S. Schenkel [1]

Features	PLS	L	NN1	NN2	NN3	NN4	NN5	NN6	NN7	NN8	NN9	NN10	LB	NNB1	NNB2	NNB3	NNB4	NNB5	NNB6	NNB7	NNB8	NNB9	NNB10	SD
1	96.47	96.55	93.08	93.16	93.33	94.74	93.98	94.36	97.65	97.65	96.54	99.39	96.26	91.67	93.01	92.59	92.07	92.41	92.2	92.2	92.22	92.08	92.45	**2.26**
2	95.68	95.29	91.4	93.47	94.64	94.05	95.54	96.11	95.44	98.04	97.66	96.16	95.61	91.93	92.55	93.26	92.89	93.12	93.07	93.06	92.97	92.92	93.1	**1.78**
3	96.32	96.36	93.58	94.44	93.96	94.55	94.71	97.27	97.69	97.39	95.15	98.47	95.99	92.49	93.19	91.46	91.91	92.35	92.45	92.3	92.31	92.46	92.59	**2.13**
4	95.84	95.73	93.4	94.18	96.15	93.58	96.82	94.14	96.05	97.44	96.65	96.57	95.59	91.8	93.41	93.14	93.35	92.84	92.66	92.81	93.17	93.22	92.79	**1.67**
5	96.5	96.72	95.23	94.89	93.41	97.01	98.35	98.94	97.31	99.91	100.94	100.9	96.11	92.6	93.87	93.05	92.82	93.08	92.66	92.98	92.83	92.8	92.83	**2.84**
6	96.75	96.07	94.41	95.86	96.8	97.44	97.89	101.16	100.12	100.65	98.93	101.93	95.66	92.55	93.46	93.87	93.33	93.07	93.4	93.45	93.34	93.35	93.62	**2.94**
7	73.7	72.44	77.56	81.19	81.51	85.42	84.69	82.72	84.49	82.27	80.79	84.57	71.61	71.75	76.2	78.96	79.48	81.59	82.58	78.65	80.36	79.33	81.66	**4.15**
8	71.76	70.31	75.75	77.95	78.08	79.72	83.64	82.61	81.85	85.36	78.72	83.08	70.83	70.87	76.12	77.5	83.09	80.28	82.42	82.1	79.16	80.56	81.71	**4.44**
SD	**10.91**	**11.48**	**7.90**	**6.94**	**7.08**	**6.20**	**5.78**	**7.02**	**6.76**	**6.96**	**8.48**	**7.26**	**11.42**	**9.67**	**7.92**	**6.83**	**5.41**	**5.52**	**4.75**	**5.84**	**6.06**	**5.98**	**5.20**	

➔ **Not only focus on the model type**

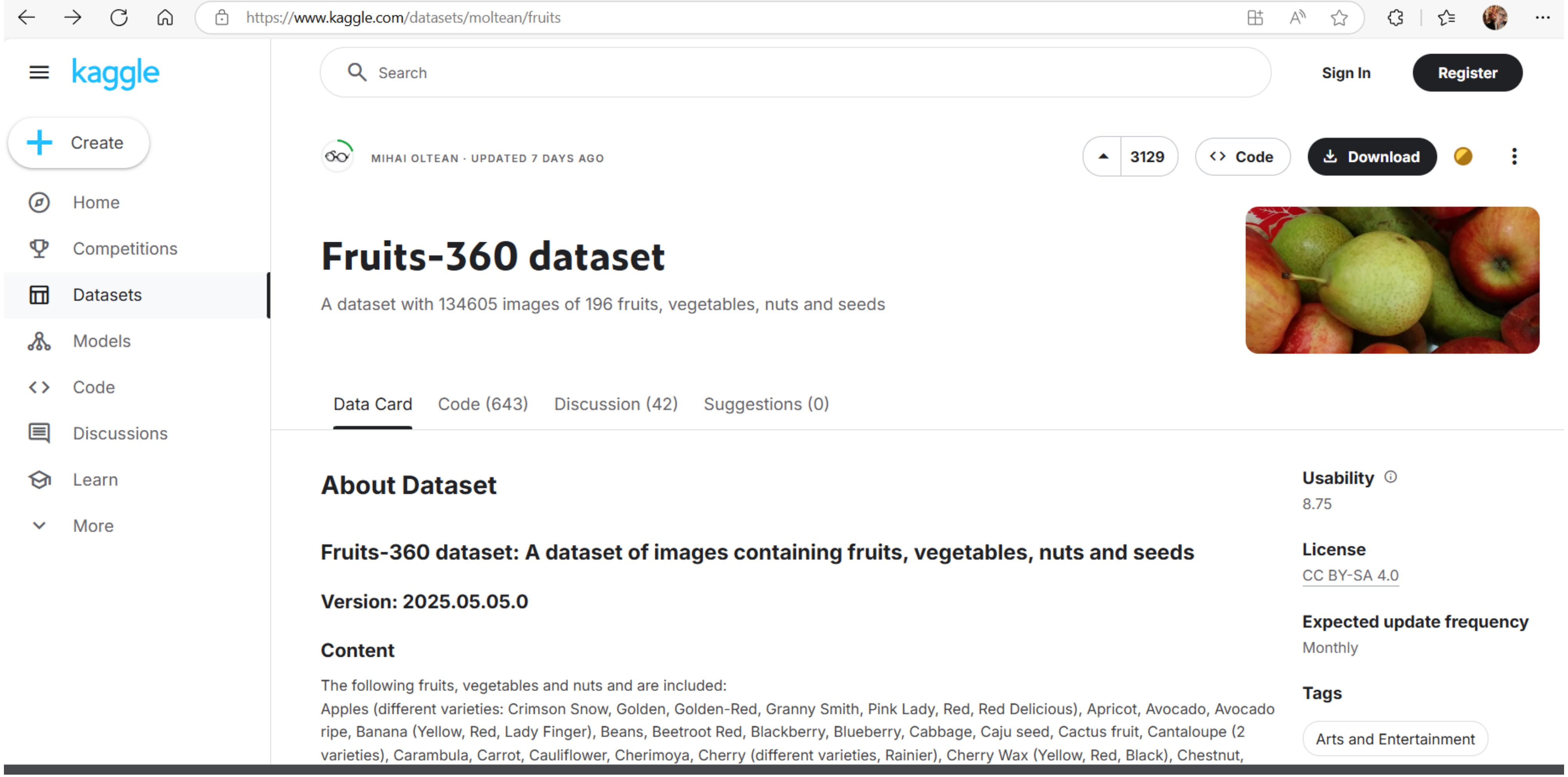
CONCLUSIONS

- **A lot of hyperparameters** to tune
- Infinite possibility of cANN and MLP structure
- Take care to the reproducibility of the model
- Limited interest for regression but high interest for classification and **image processing**
- However, **transfer learning** can solve problems of data sharing or allows to deal with large database

PYTHON SCRIPT EXAMPLE

kaggle

https://www.kaggle.com/datasets/moltean/fruits



```python
#####################
# MLP
#####################

# MLP
ANN1 = tf.keras.Sequential([
    tf.keras.Input(shape=(100, 100, 3)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(700, activation="relu"),
    tf.keras.layers.Dense(300, activation="relu"),
    tf.keras.layers.Dense(100, activation="relu"),
    tf.keras.layers.Dense(31, activation="softmax")
])

print("Perceptron with one hidden layer",ANN1.summary())
tf.keras.utils.plot_model(ANN1)

# network compilation
ANN1.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

# Fit the network
history = ANN1.fit( x = xtrain,
          y = ytrain,
          epochs = 50, # 50 epochs
          batch_size = 64, # mini-batch gradient descent with 32 samples
          validation_split = 0.2,
          callbacks = [tf.keras.callbacks.EarlyStopping(patience=3)]) # patience option is interesting to be sure that the minimum is not a local minimum.
```

```python
# model creation : 64 filters - 256 neurons in the last hidden layer
cANN = tf.keras.Sequential([
    tf.keras.Input(shape=(100, 100, 3)),  # declare input shape separately
    tf.keras.layers.Conv2D(10, 7, activation="relu", padding="same"),
    tf.keras.layers.MaxPooling2D(2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation="relu"),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(64, activation="relu"),
    tf.keras.layers.Dense(31, activation="softmax")  # adjust output units to your number of classes
])

print("Convolutional neural network",cANN.summary())
tf.keras.utils.plot_model(cANN)

# network compilation
cANN.compile(tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9), # optimizer "adam" "sgd"
              loss='categorical_crossentropy',   #tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

# Fit the network
history2 = cANN.fit(x=xtrain,
          y=ytrain,
          batch_size=64,
          epochs=50,
          verbose=1,
          validation_data=(xtest, ytest),
          callbacks = [tf.keras.callbacks.EarlyStopping(patience=3)]) # patience option is interesting to be sure that the minimum is not a local minimum.
```