International Workshop on Machine Learning for Space Weather: Fundamentals, Tools and Future Prospects

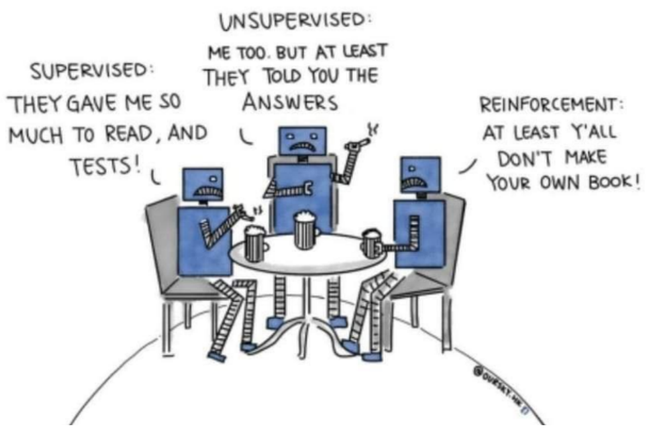
7-11 November 2022
This is a hybrid meeting
Buenos Aires, Argentina

Further information:
https://indico.ictp.it/event/9840/
smr3750@ictp.it
+39-040-2240284
Elizabeth Brancaccio

Dra María Graciela Molina
FACET-UNT / CONICET
Tucumán Space Weather Center - TSWC

https://spaceweather.facet.unt.edu.ar/
IG -> @spaceweatherargentina

gmolina@herrera.unt.edu.ar



MACHINE LEARNING

Algorithms

Supervised

- Regression
  - Linear
  - Polynomial
- Decision Tree
- Random Forest
- Classification
  - KNN
  - Logistic Regression
  - Naive-Bayes
  - SVM

Unsupervised

- Clustering
  - K-means
  - PCA
  - SVD
- Association Rule Learning
- Hidden Markov Model

Continuous data

Categorical data

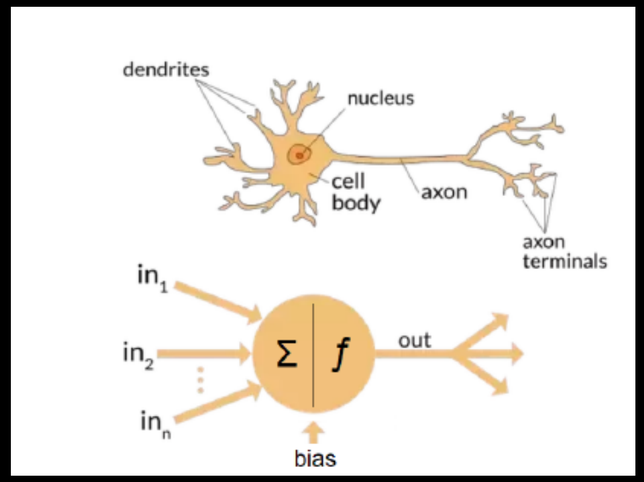
Artificial Neural Networks

TSWC, 2022

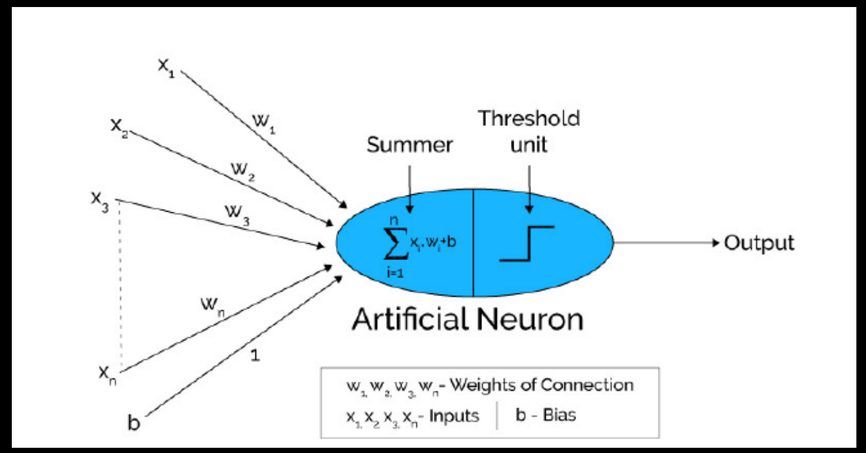
Artificial Neural Networks

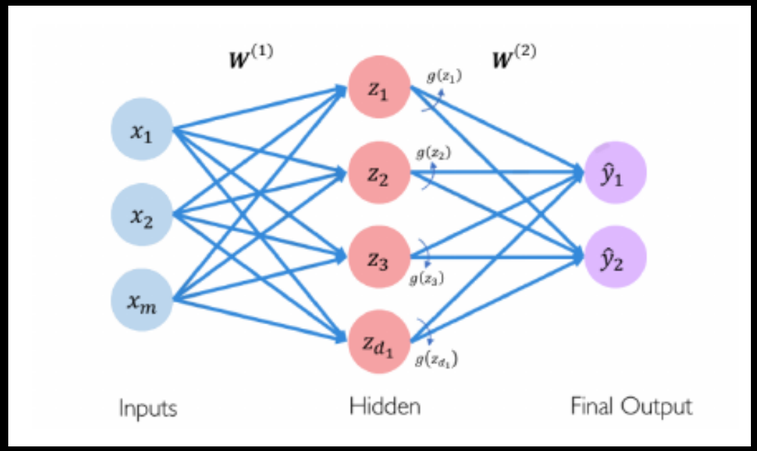
TSWC, 2022

ANN



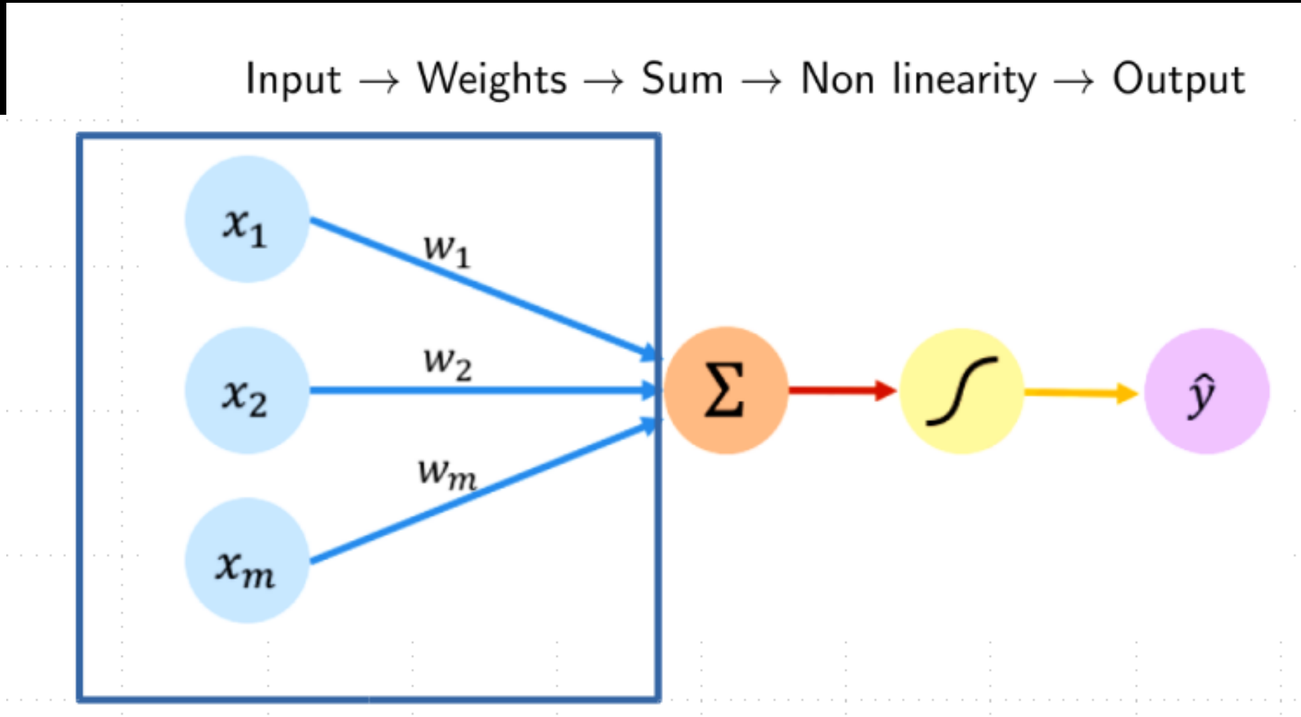
- Data-driven modeling
- Inspired in brain neural networks
- Solving wide number of complex problems: facial recognition, handwrite recognition, weather forecasting, etc.
- Black-box modelling (?) based in the composition of interconnected neurons (neural network)





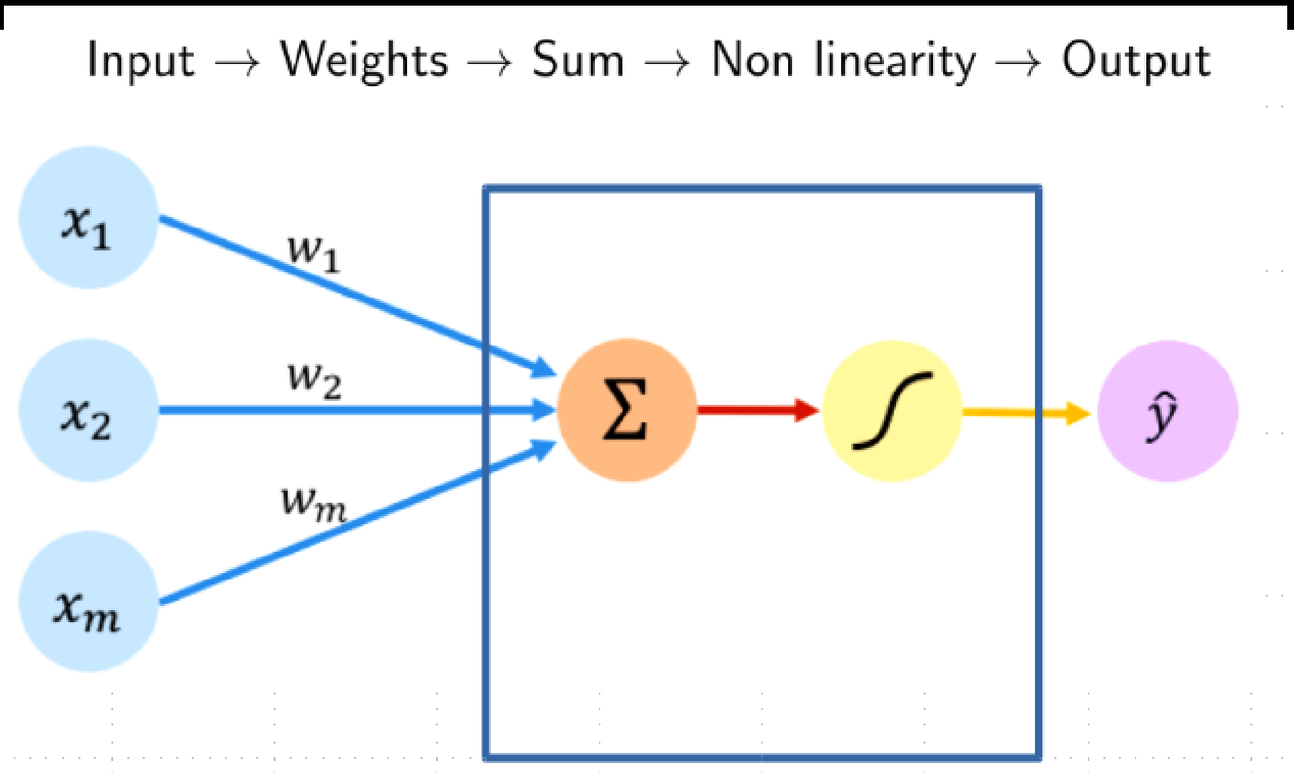
Perceptron

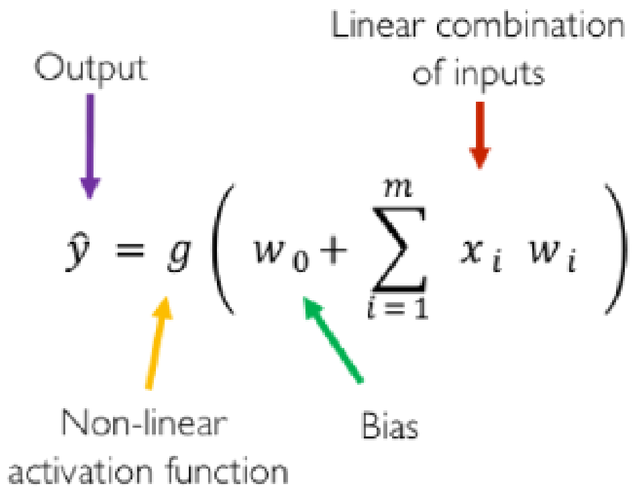
- Unidad fundamental para construir redes neuronales



- Cada una de las entradas corresponde a una característica (feature)
- A cada una de las xi se les aplica un peso (w) (ponderación de los valores de la entrada)
- Los pesos son las vbles que se ajustan durante el entrenamiento

Perceptron

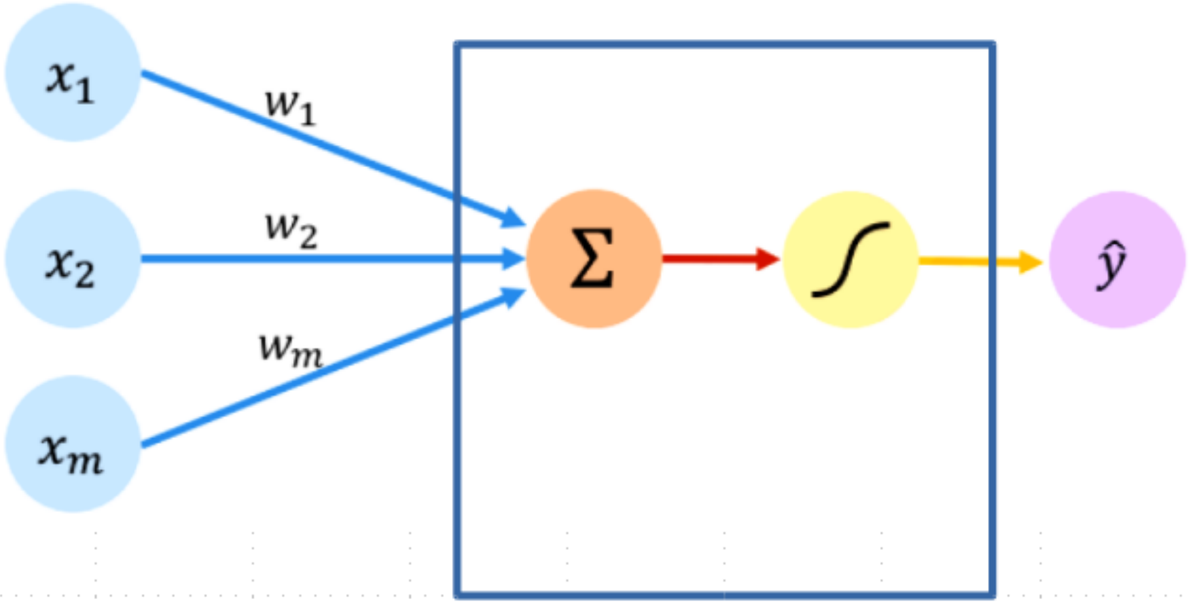




- Capa oculta (hidden layer).
- El bias es un término de ajuste
- La función de activación se aplica a la combinación lineal de las entradas ponderadas teniendo en cuenta el bias

Perceptron

Input → Weights → Sum → Non linearity → Output



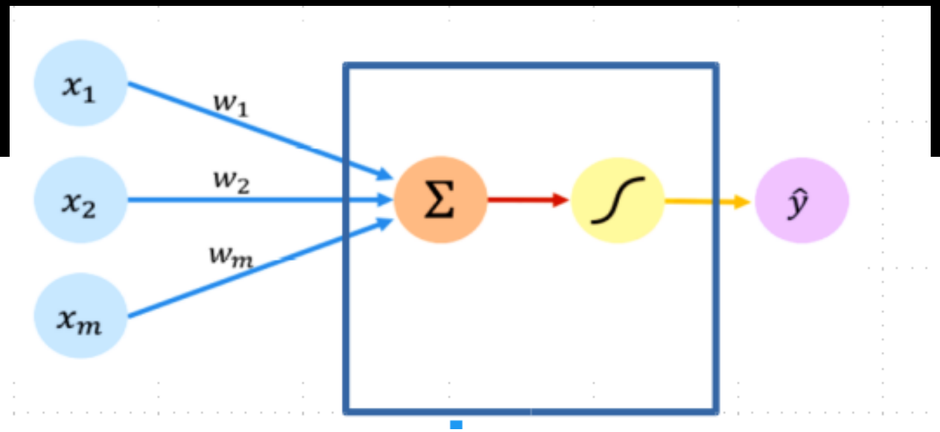
$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i\, w_i\right)$$

$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W}\right)$$

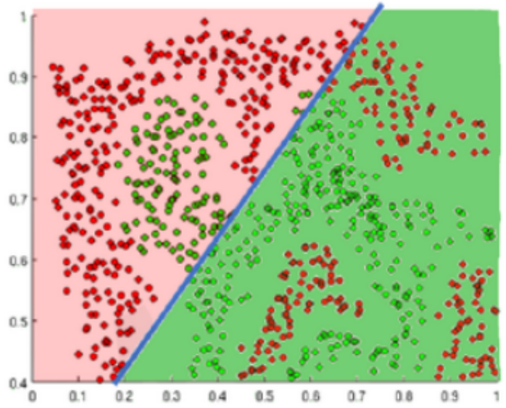
where: $\boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$ and $\boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$

Activation function

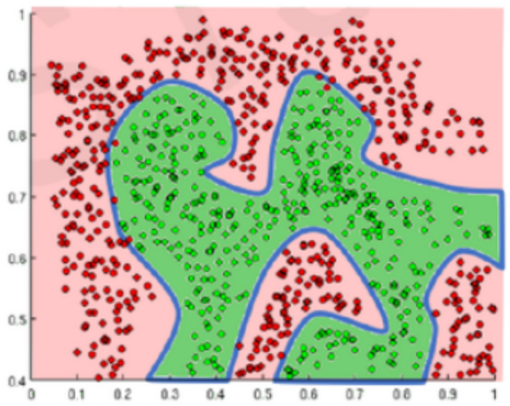
- Agrega no-linealidd al modelo



$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$



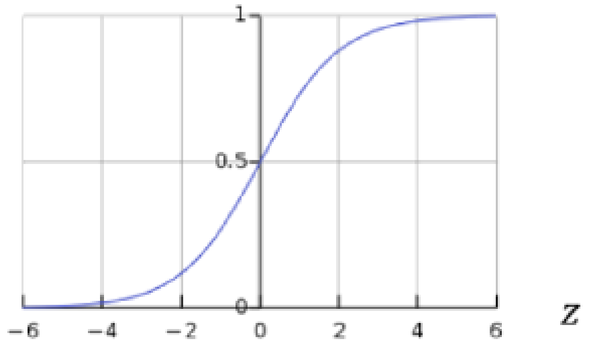
Linear activation functions produce linear decisions no matter the network size



Non-linearities allow us to approximate arbitrarily complex functions

- Una de las más usadas en la función sigmoide.
- Produce resultados entre 0 y 1

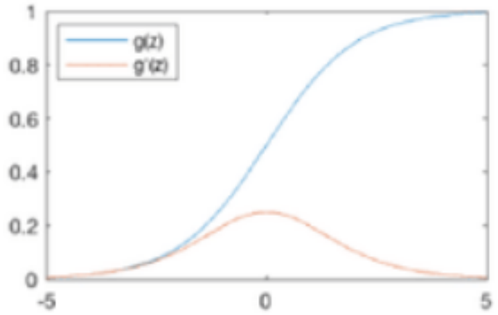
$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



Activation function

$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

Sigmoid Function

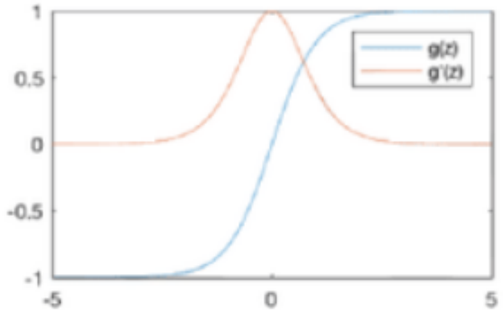


$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

`tf.math.sigmoid(z)`

Hyperbolic Tangent

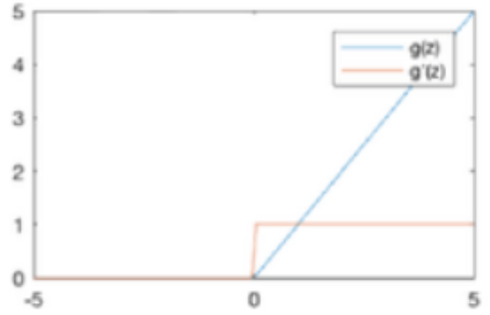


$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

`tf.math.tanh(z)`

Rectified Linear Unit (ReLU)



$$g(z) = \max(0, z)$$

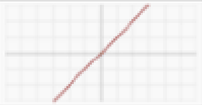
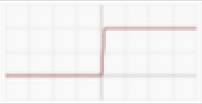
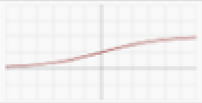
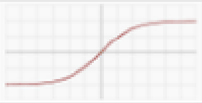
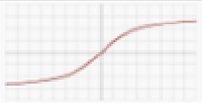
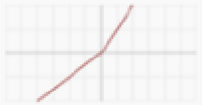
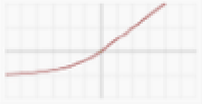
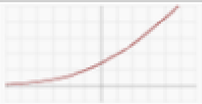
$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

`tf.nn.relu(z)`

Activation function

$$\hat{y} = g(w_0 + \boldsymbol{X}^T \boldsymbol{W})$$

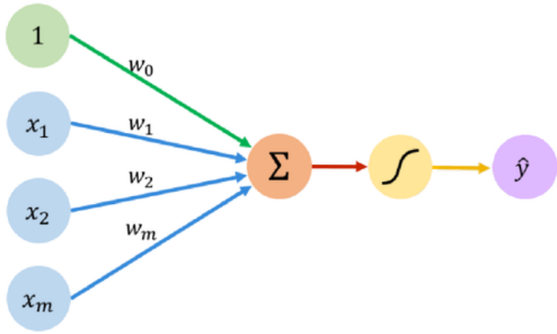
- Adds no-linearity to the model

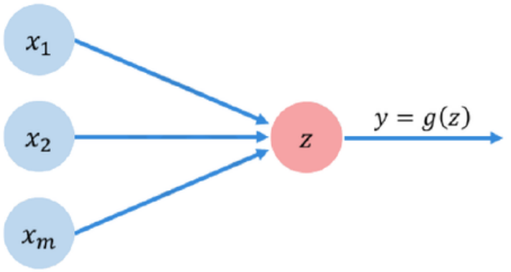
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Feed Forward (FF)

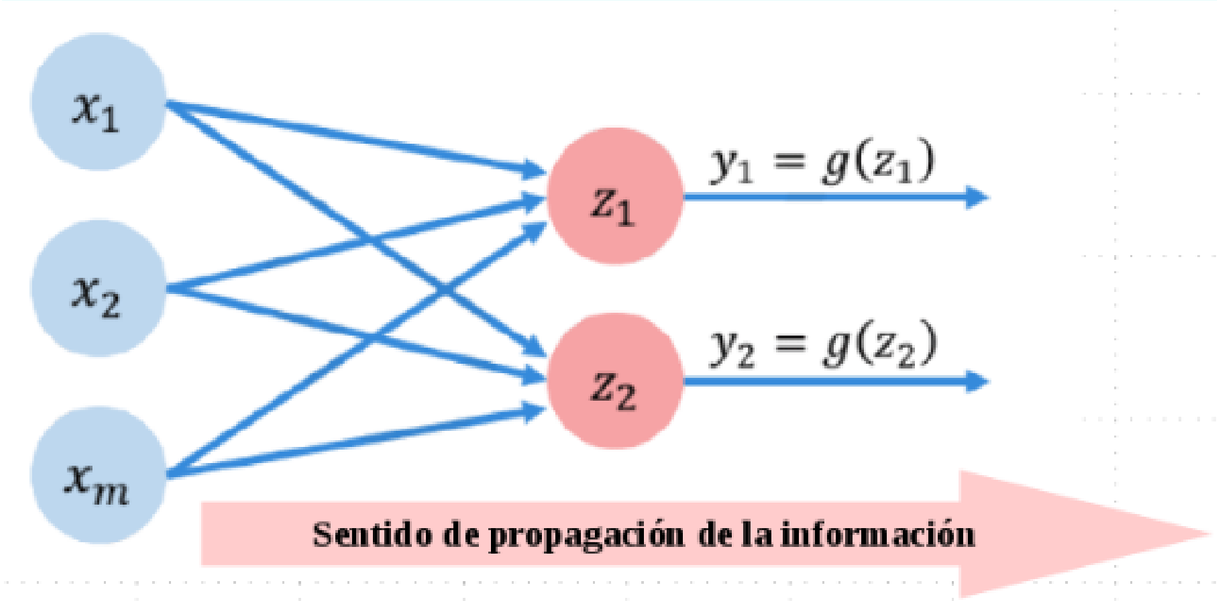
- Perceptrón simplificado (n entradas → 1 salida)

$$z = w_0 + \sum_{j=1}^{m} x_j \, w_j$$





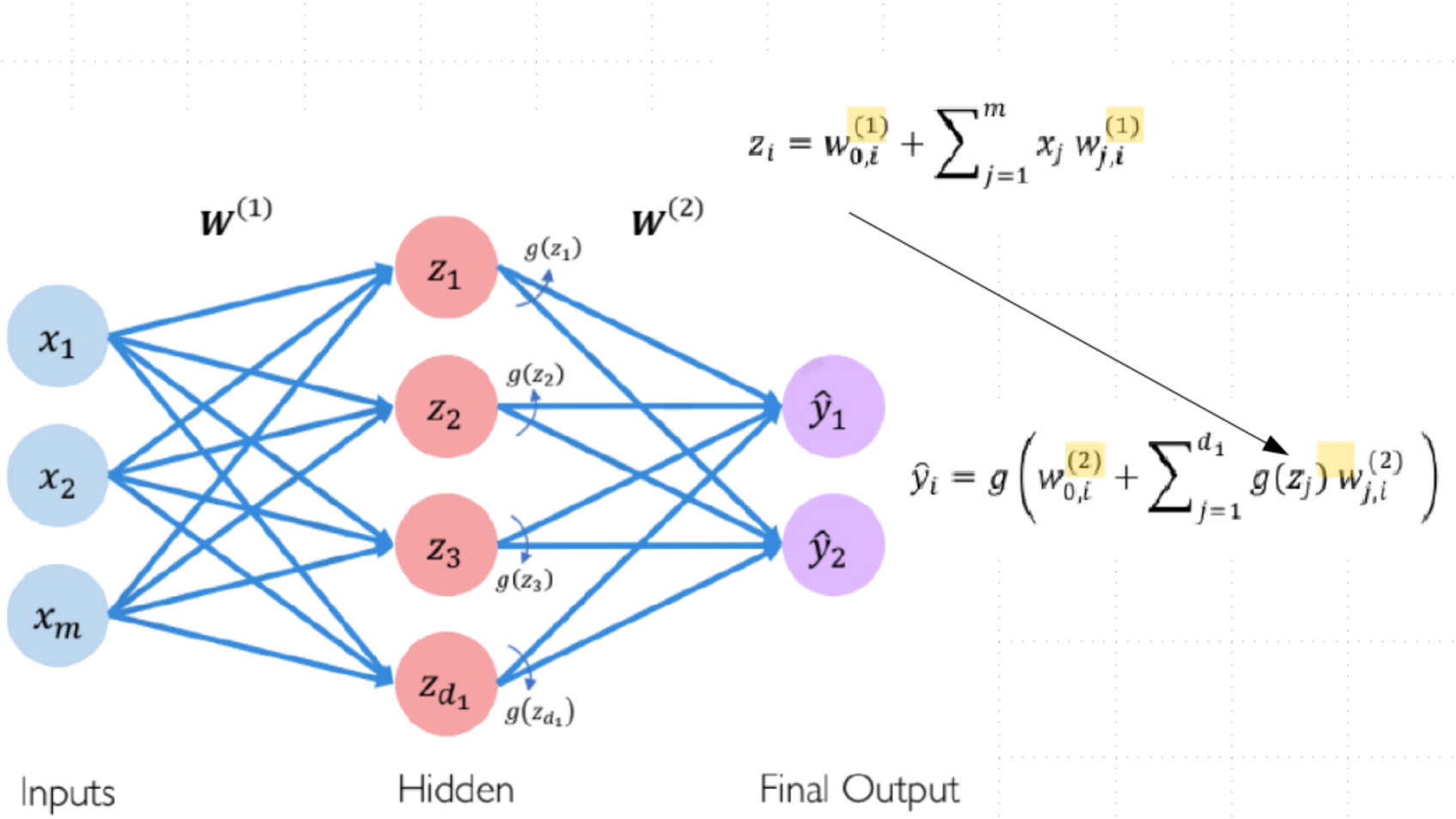
- Armemos una red neuronal compuesta por múltiples perceptrones (n entradas → m salida)



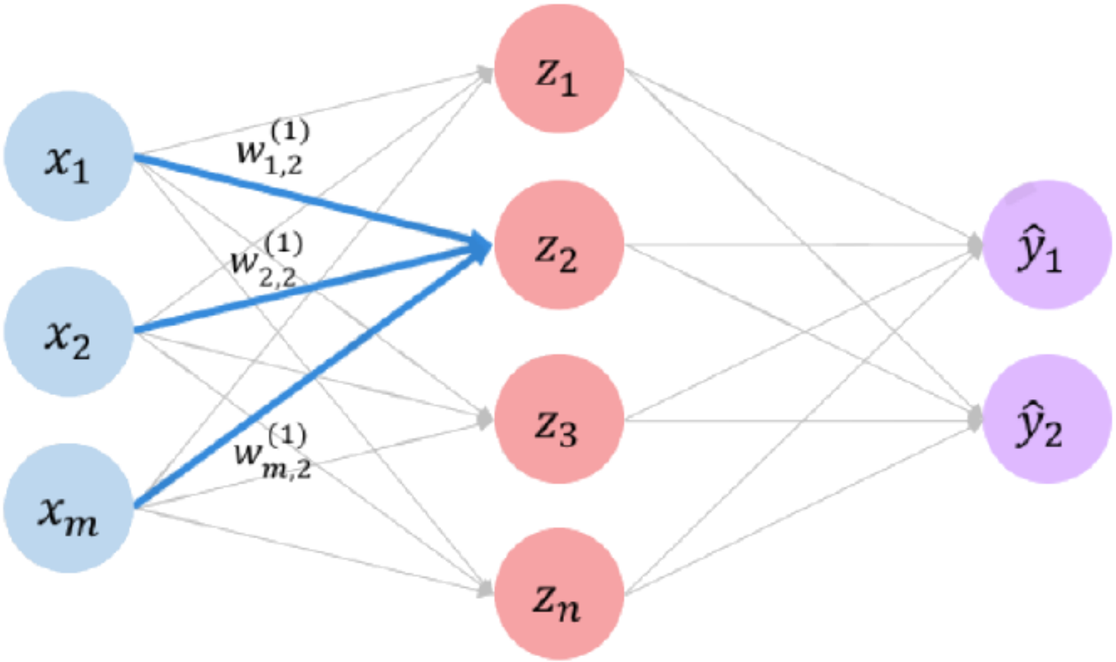
$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j \, w_{j,i}$$

Observación: todas las entradas están conectadas a todas las salidas → dense layers

1-Layer Neural Network



1-Layer Neural Network

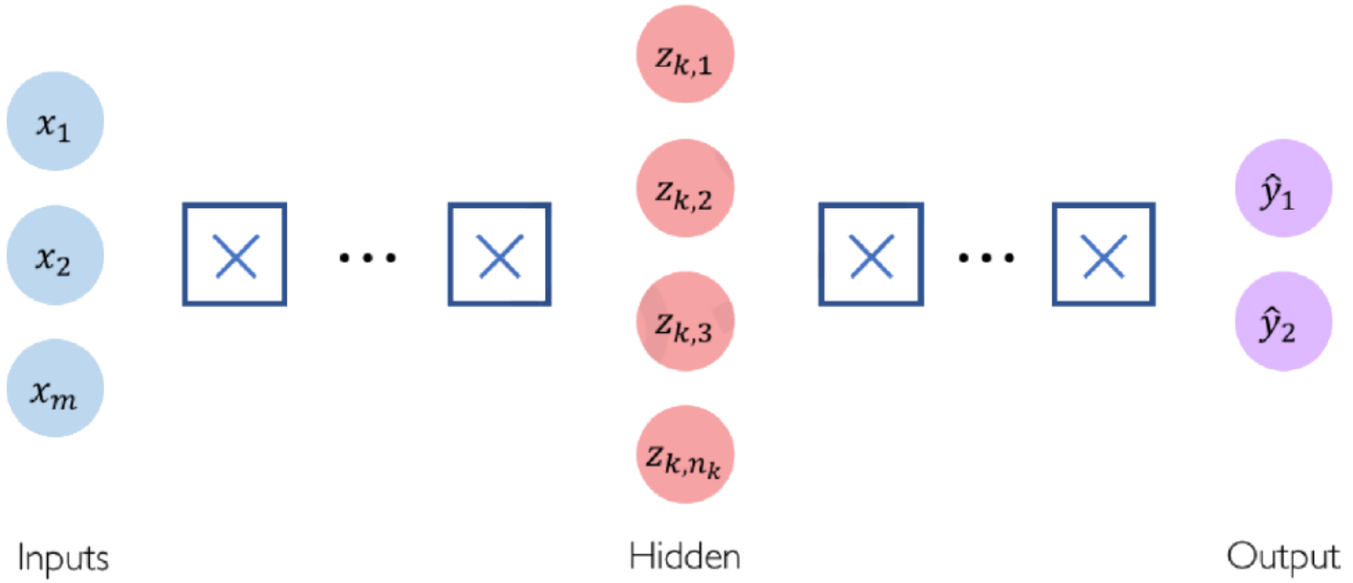


$$z_2 = w_{0,2}^{(1)} + \sum_{j=1}^{m} x_j \, w_{j,2}^{(1)}$$

$$= w_{0,2}^{(1)} + x_1 \, w_{1,2}^{(1)} + x_2 \, w_{2,2}^{(1)} + x_m \, w_{m,2}^{(1)}$$

M-layers Neural Network (multi-layer)

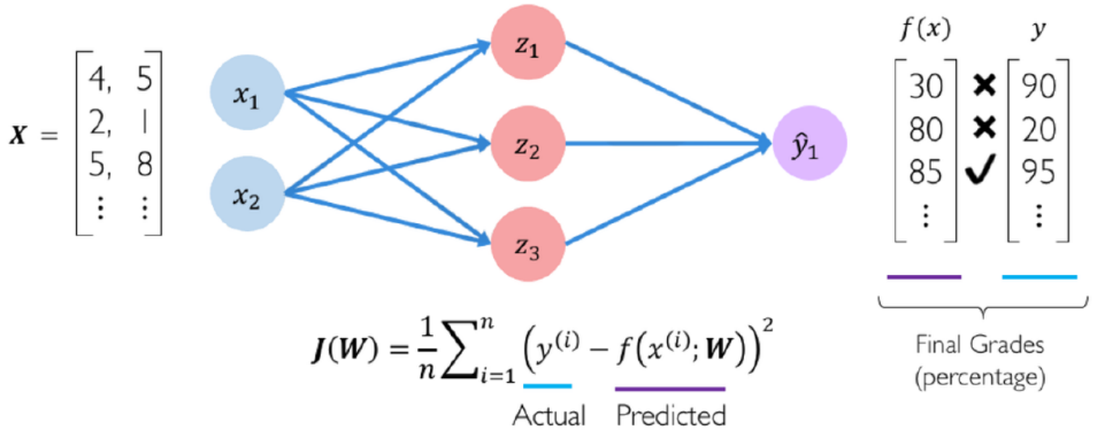
- Deep Neural Network



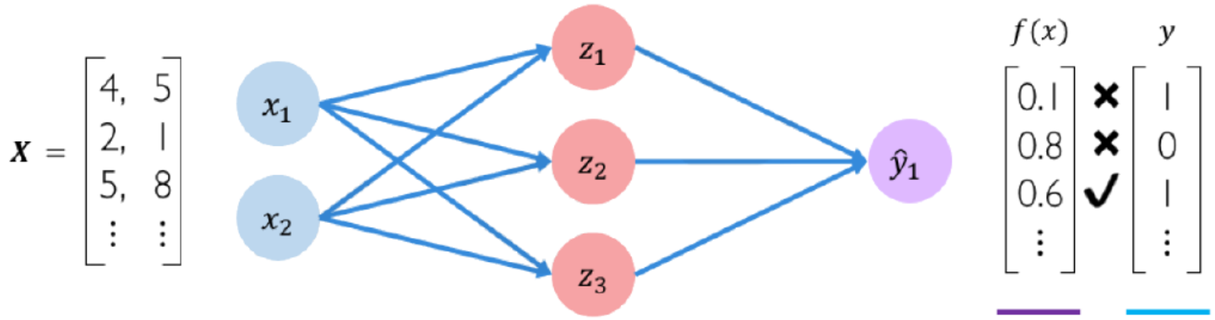
$$z_{k,i} = w_{0,i}^{(k)} + \sum_{j=1}^{n_{k-1}} g(z_{k-1,j})\, w_{j,i}^{(k)}$$

Loss Function

Mean squared error loss *can be used with regression models that output continuous real numbers*



Cross entropy loss *can be used with models that output a probability between 0 and 1*



symbol	name	equation
$\mathcal{L}_1$	L_1 loss	$\|\mathbf{y} - \mathbf{o}\|_1$
$\mathcal{L}_2$	L_2 loss	$\|\mathbf{y} - \mathbf{o}\|_2^2$
$\mathcal{L}_1 \circ \sigma$	expectation loss	$\|\mathbf{y} - \sigma(\mathbf{o})\|_1$
$\mathcal{L}_2 \circ \sigma$	regularised expectation loss[1]	$\|\mathbf{y} - \sigma(\mathbf{o})\|_2^2$
$\mathcal{L}_\infty \circ \sigma$	Chebyshev loss	$\max_j \lvert\sigma(\mathbf{o})^{(j)} - \mathbf{y}^{(j)}\rvert$
hinge	hinge [13] (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)}\mathbf{o}^{(j)})$
hinge^2	squared hinge (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)}\mathbf{o}^{(j)})^2$
hinge^3	cubed hinge (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)}\mathbf{o}^{(j)})^3$
log	log (cross entropy) loss	$-\sum_j \mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}$
$\log^2$	squared log loss	$-\sum_j [\mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}]^2$
tan	Tanimoto loss	$\frac{-\sum_j \sigma(\mathbf{o})^{(j)}\mathbf{y}^{(j)}}{\|\sigma(\mathbf{o})\|_2^2 + \|\mathbf{y}\|_2^2 - \sum_j \sigma(\mathbf{o})^{(j)}\mathbf{y}^{(j)}}$
D_{CS}	Cauchy-Schwarz Divergence [3]	$-\log \frac{\sum_j \sigma(\mathbf{o})^{(j)}\mathbf{y}^{(j)}}{\|\sigma(\mathbf{o})\|_2 \|\mathbf{y}\|_2}$

- https://arxiv.org/pdf/1702.05659.pdfs

Training (loss optimization)

- Encontrar los valores de los **pesos** de la red neuronal tal que la **pérdida o costo sea mínimo**

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}; \boldsymbol{W}\right), y^{(i)}\right)$$

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}} J(\boldsymbol{W})$$

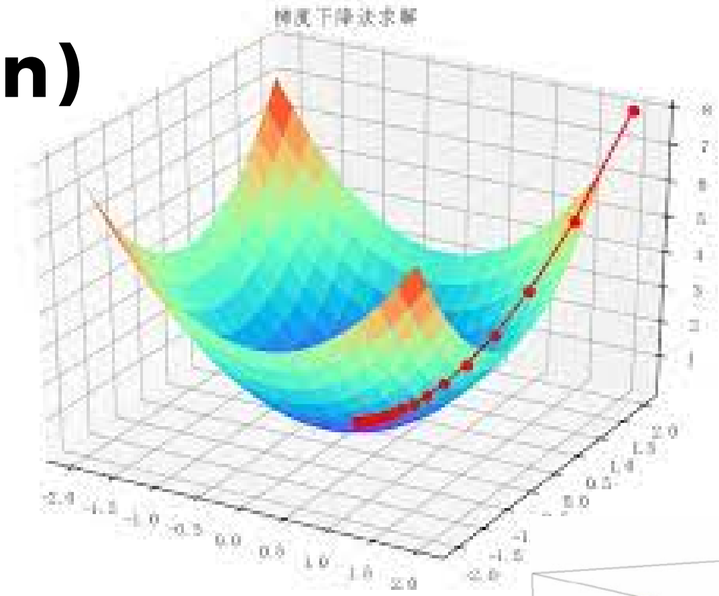
- Es un problema de optimización
- Podemos usar algún método numérico de optimización

MÉTODO DEL GRADIENTE DESCENDIENTE

Training (loss optimization)

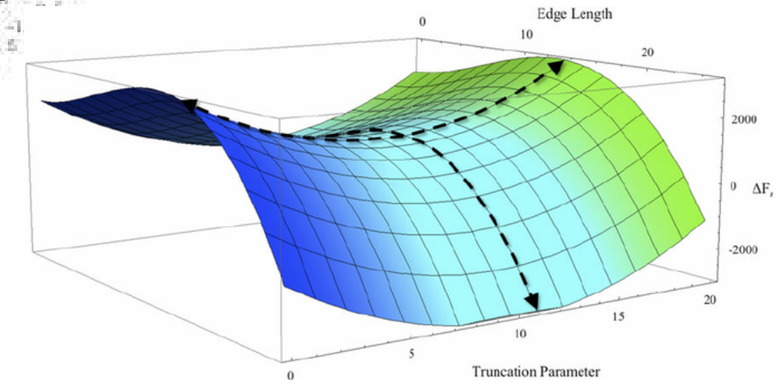
$$W^* = \underset{W}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}; W\right), y^{(i)}\right)$$

$$W^* = \underset{W}{\operatorname{argmin}} J(W)$$



Algorithm

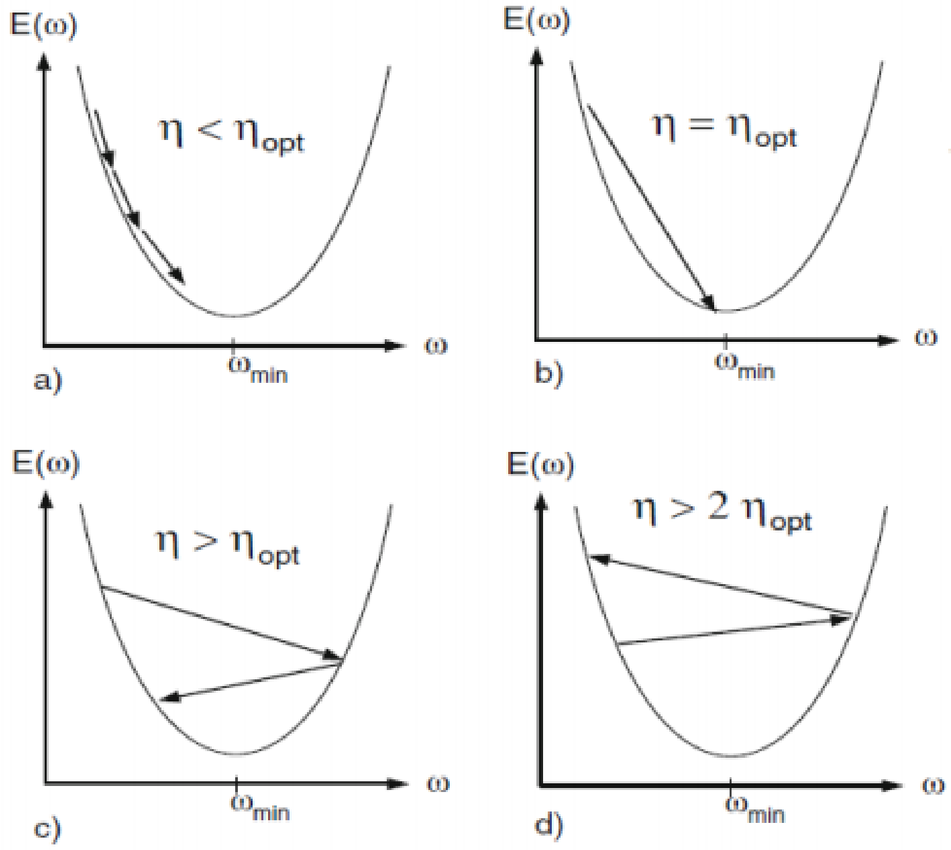
1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Compute gradient, $\frac{\partial J(W)}{\partial W}$
4. Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
5. Return weights



Principales problemas:

- Mínimos locales
- Silla de montar: el g. d. un vez que llega a una región con gradiente cero, no puede escapar de allí independientemenete de la calidad del mínimo.

Training (loss optimization)



$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

How can we set the learning rate?

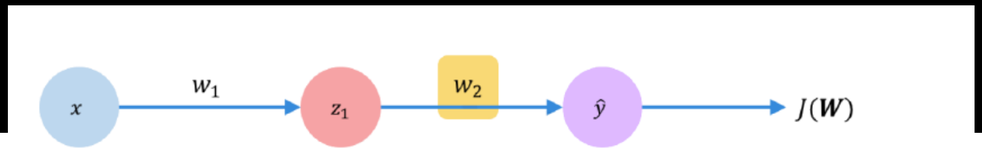
- https://ruder.io/optimizing-gradient-descent/index.html

Gradient Descent Algorithms

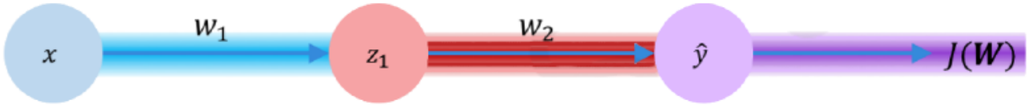
Algorithm	TF Implementation	Reference
• SGD	tf.keras.optimizers.SGD	Kiefer & Wolfowitz. "Stochastic Estimation of the Maximum of a Regression Function." 1952.
• Adam	tf.keras.optimizers.Adam	Kingma et al. "Adam: A Method for Stochastic Optimization." 2014.
• Adadelta	tf.keras.optimizers.Adadelta	Zeiler et al. "ADADELTA: An Adaptive Learning Rate Method." 2012.
• Adagrad	tf.keras.optimizers.Adagrad	Duchi et al. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." 2011.
• RMSProp	tf.keras.optimizers.RMSProp	

Backpropagation

- Qué tanto puede afectar a la función de pérdida un pequeño cambio en uno del los pesos?



- Aplicando la regla de la cadena



$$\frac{\partial J(\boldsymbol{W})}{\partial w_1} = \frac{\partial J(\boldsymbol{W})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z_1} * \frac{\partial z_1}{\partial w_1}$$

- Repetir para cada uno de los pesos en la red usando los gradientes de las capas siguiente!

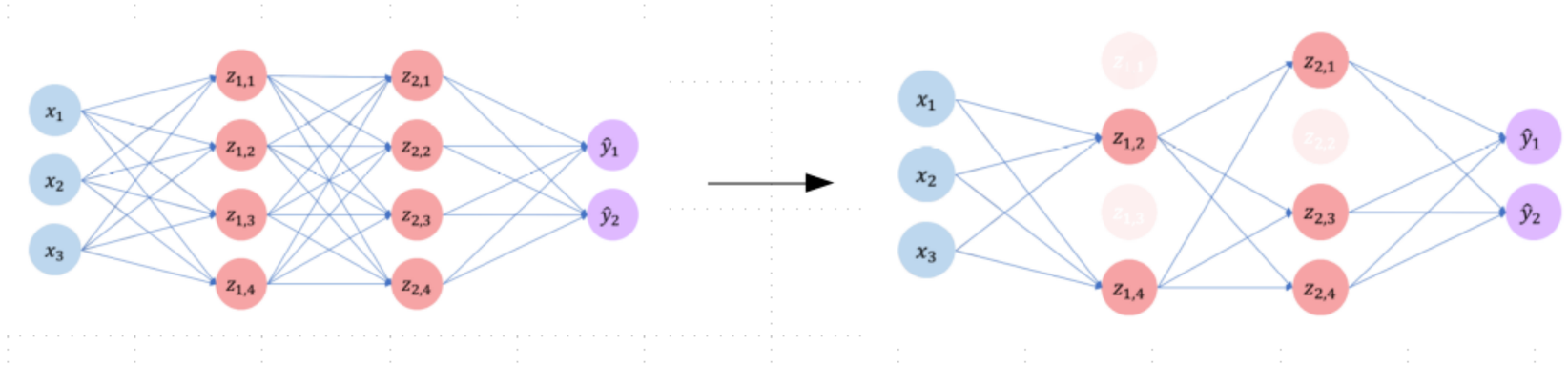
- Info se propaga desde las entradas hacia adelante mediante sus parámetros hasta que logra hacer una predicción

- Luego realiza una propagación hacia atrás a lo largo de la red para ir modificando los parámetros de manera que el error final sea el mínimo.

- El error asociado a una mala predicción es usado para ajustar los parámetros (el aprendizaje puede ser visto como una optimización). Para esto la salida y es comparada con el valor esperado (target) del conjunto de entrenamiento (z). La diferencia se entre el valor esperado y el de salida se llama error o residuo.

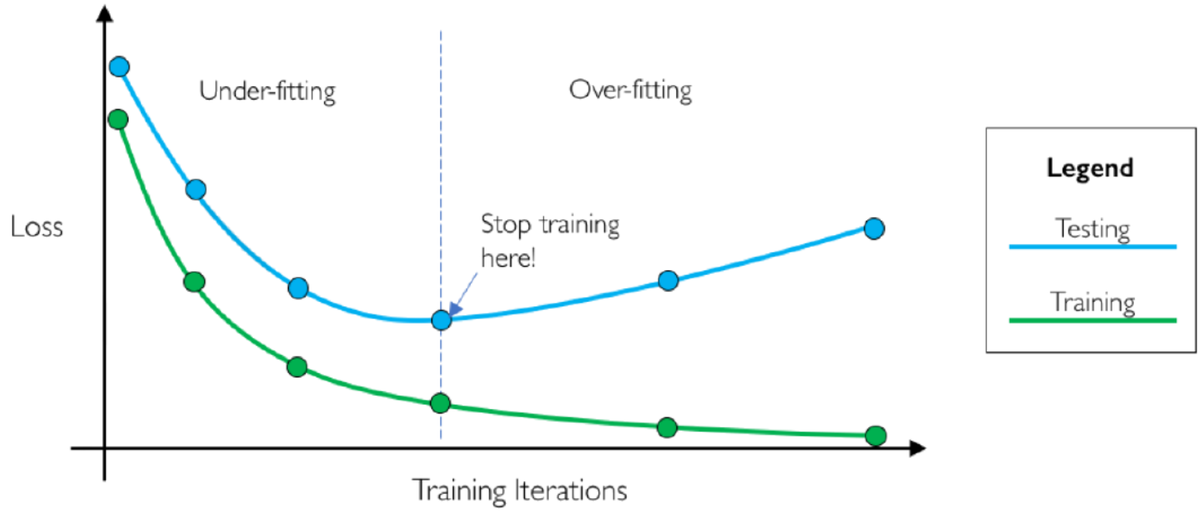
Regularización

- Técnica que consiste en establecer restricciones al problema de optimización para evitar llegar a modelos complejos
- La idea es poder generalizar nuestro modelo para nuevos datos



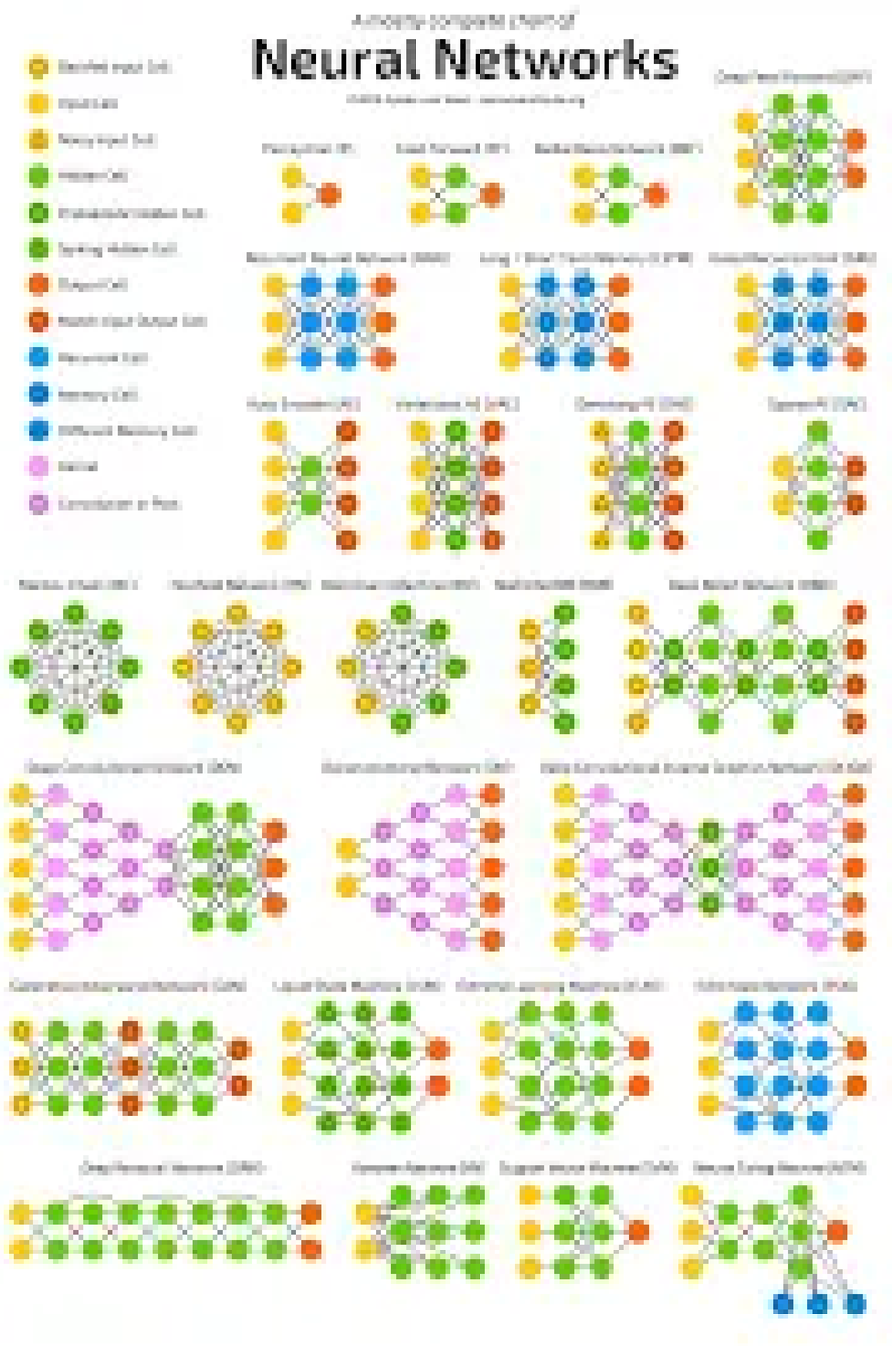
DROP OUT

- Durante el entrenamiento de manera aleatoria se setean algunas activaciones en 0.
- << overfitting



EARLY STOPING

- Consiste en parar antes de llegar al overfitting observando los datos de training y testing.


A mostly complete chart of
Neural Networks


ZOO

HANDS-ON LAB

TP - 1

2 problems

- Radar signal classification (mandatory)
- Flare classification (optional)

From science to data science

PROBLEM 1 - Classify radar returns from the ionosphere as either suitable for further analysis or not (“good” or “bad”).
The 17 pairs of numbers, representing 17 discrete values of the real part and the corresponding 17 values of the complex part of an ACF, are the input to the neural network.

VINCENT G. SIGILLITO, SIMON P. WING, LARRIE V. HUTTON, and KILE B. BAKER

CLASSIFICATION OF RADAR RETURNS FROM THE IONOSPHERE USING NEURAL NETWORKS

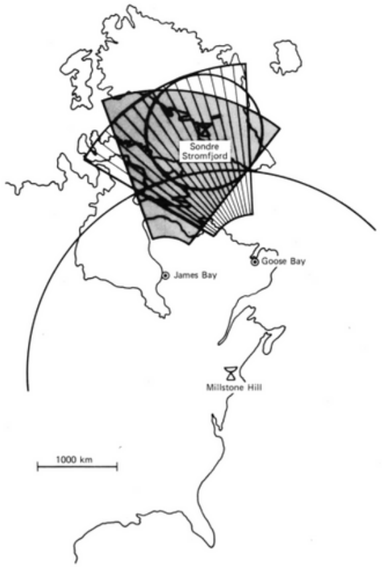
In ionospheric research, we must classify radar returns from the ionosphere as either suitable for further analysis or not. This time-consuming task has typically required human intervention. We tested several different feedforward neural networks to investigate the effects of network type (single-layer versus multilayer) and number of hidden nodes upon performance. As expected, the multilayer feedforward networks (MLFN’S) outperformed the single-layer networks, achieving 100% accuracy on the training set and up to 98% accuracy on the testing set. Comparable figures for the single-layer networks were 94.5% and 92%, respectively. When measures of sensitivity, specificity, and proportion of variance accounted for by the model are considered, the superiority of the MLFN’s over the single-layer networks is even more striking.

Paper source:
Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266.

Data source:
Space Physics Group of the Johns Hopkins University Applied Physics Laboratory.

From science to data science

Dr. Z. Saavedra - FACET, UNT



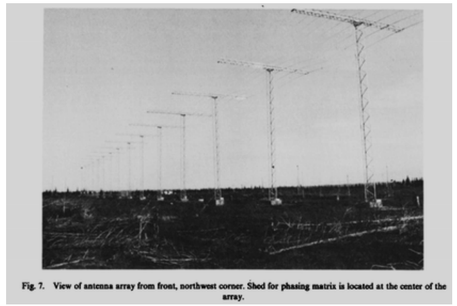
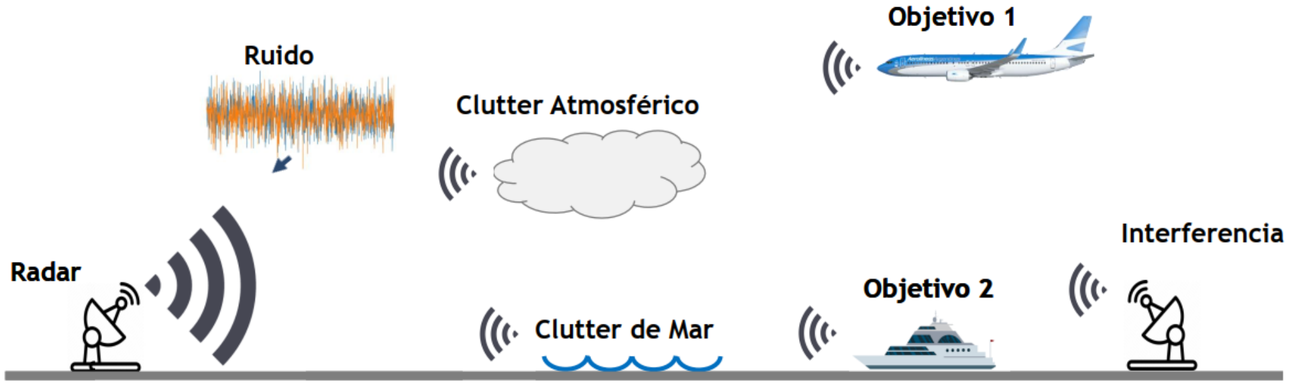
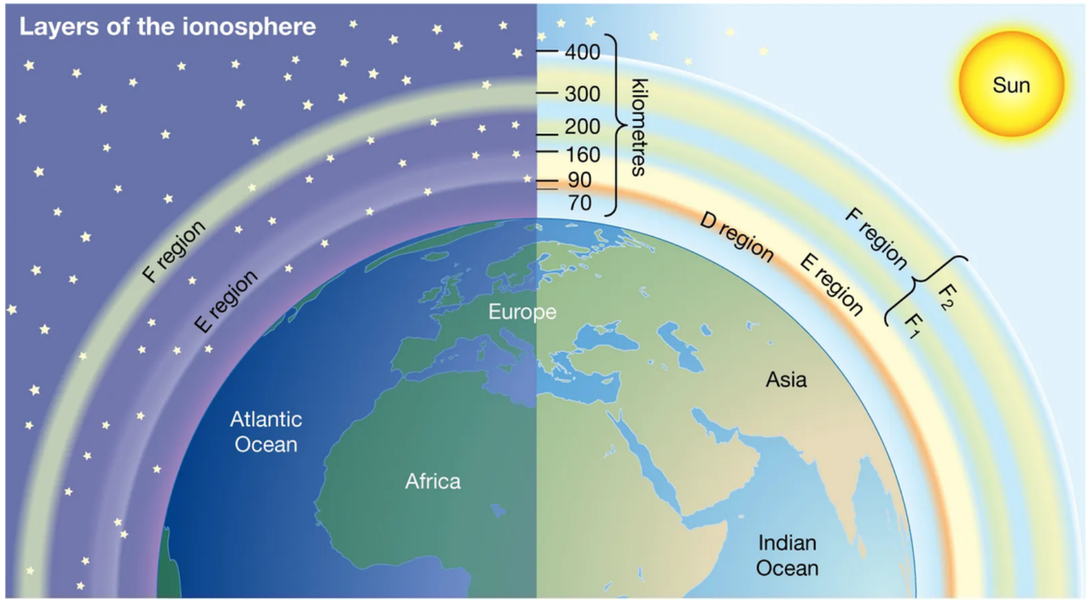


Fig. 7. View of antenna array from front, northwest corner. Shed for phasing matrix is located at the center of the array.





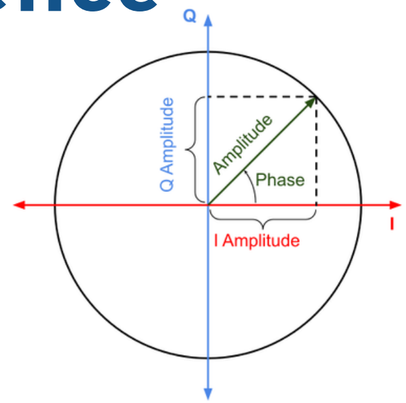
$$Eco(t) = Atte \cdot RCS \cdot u(t) \cdot m(t) \cdot e^{j\omega_c t \pm j\omega_D t + \varphi}$$

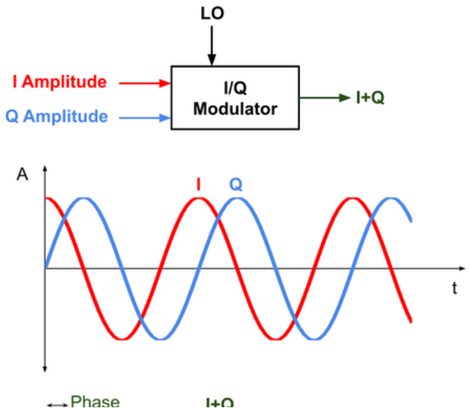
$$S_R(t) = Eco(t) + Clutter\ (t) + Ruido(t) + Interferencia(t)$$

*: "target" = ionosphere

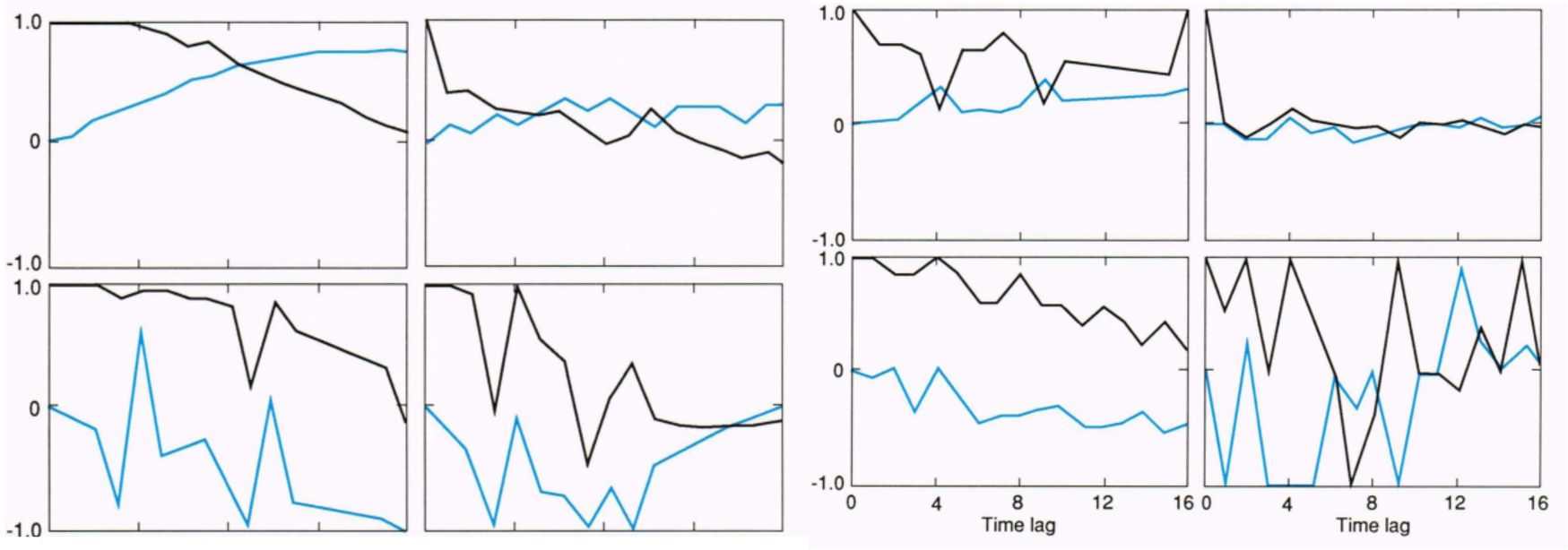
From science to data science

IQ Modulation: it is a specific Phase-Modulation type, that provides us certain information in the real component of the signal and another posible coded information in the imaginary part. We can obtain both of them using demodulation techniques such as quadrature hybrids.





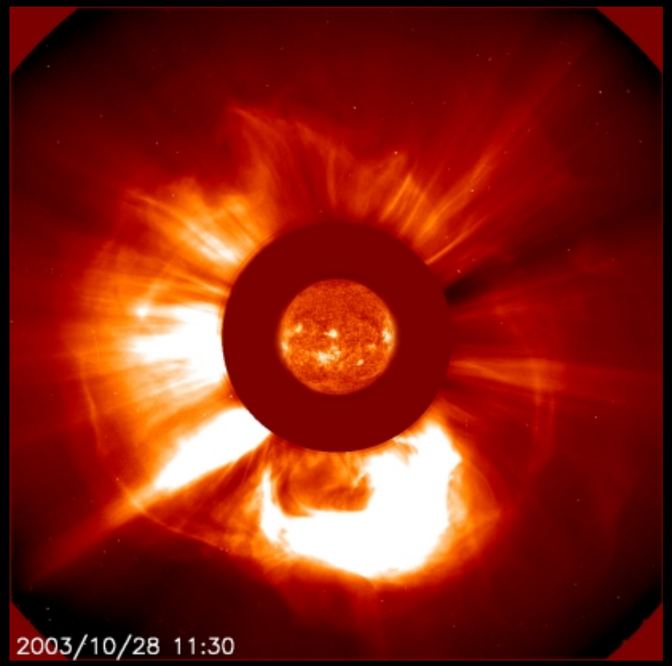
$$C(t) = A(t) + iB(t)$$



Good echo *Bad echo* *Good echo* *Bad echo*

Auto-Correlation Function Output

From science to data science



28 Oct 2003 - X 17.2 flare -> SPE + CME (geoeffective, 29 Oct)

29 Oct 2003 - X 10.0 flare (same AR) -> CME

The problem (optional)

- Automatic **classification of CMEs** according to their origin (flare/no-flare)
- From images (C2,C3 LASCO - SOHO) + expert observations
- source: https://cdaw.gsfc.nasa.gov/CME_list/index.html
- Aknowlwdgements: David Salazar de la Escuela de Física de la Universidad de Costa Rica + Salas-Matamoros & Klein, 2016.

From science to data science

CME						Origen del CME								
						FLARE				Filamento				
Date			LASCO onset		Vprop	Coordinates		SXR onset		Observation date (Ha)			Coordinates	
DD	MM	YY	HH	MM		lat	long	HH	MM	DD	MM	YY	lat	long
7	4	97	14	27	592	-28	-11	14	8					
12	5	97	6	30	355	21	8	4	55					
21	6	98	5	35	619	18	39	5	18					
13	4	99	3	30	309	16	0	2	14					
6	6	0	15	54	1557	21	-10	15	26					
20	6	0	9	10	516	-27	38	8	27					
25	7	0	3	30	712	6	8	2	50					
21	7	6	13	54	403					20	7	6	-20	-25
29	12	6	21	30	236					29	12	6	-20	35
23	5	10	18	30	258					21	5	10	15	-20
30	1	11	19	36	521*					30	1	11	25	25
11	1	12	21	36	302					11	1	12	45	-25
16	3	12	20	36	862*					16	3	12	20	60
1	4	12	3	12	410*					31	3	12	20	-20
27	12	12	21	28	267					27	12	12	5	-20

Data Transformation

Columnas 1-3: Fecha de observación del CME
Columnas 4-5: Momento de la primera observación en C2 de LASCO reportado en el catálogo https://cdaw.gsfc.nasa.gov/CME_list/
Columna 6: Velocidad de propagación del CME calculado mediante la fórmula empírica en Salas-Matamoros & Klein, 2016.
En eventos asociados a erupción de filamento, la velocidad de propagación del CME es la reportada en el catálogo https://cdaw.gsfc.nasa.gov/CME_list/, donde dicha velocidad es calculada por un 'first-order polynomial fit' a partir de mediciones de altura vrs tiempo.
Columnas 7-8: Localización de la región activa que dio origen al Flare.
Columnas 9-10: Momento de inicio del busrt de SXR observado en los datos de GOES.
Columnas 11-13: Fechas de observación del filamento en imágenes en Ha (en www.solarmonitor.org).
Columnas 14-15: La posición central de cada filamento.

From science to data science

- Features need selection & transformation

- Target needs transformation

CME						Origen del CME								
						FLARE				Filamento				
Date			LASCO onset		Vprop	Coordinates		SXR onset		Observation date (Ha)			Coordinates	
DD	MM	YY	HH	MM		lat	long	HH	MM	DD	MM	YY	lat	long
7	4	97	14	27	592	-28	-11	14	8					
12	5	97	6	30	355	21	8	4	55					
21	6	98	5	35	619	18	39	5	18					
13	4	99	3	30	309	16	0	2	14					
6	6	0	15	54	1557	21	-10	15	26					
20	6	0	9	10	516	-27	38	8	27					
25	7	0	3	30	712	6	8	2	50					
21	7	6	13	54	403					20	7	6	-20	-25
29	12	6	21	30	236					29	12	6	-20	35
23	5	10	18	30	258					21	5	10	15	-20
30	1	11	19	36	521*					30	1	11	25	25
11	1	12	21	36	302					11	1	12	45	-25
16	3	12	20	36	862*					16	3	12	20	60
1	4	12	3	12	410*					31	3	12	20	-20
27	12	12	21	28	267					27	12	12	5	-20
.														
.														
.														

Automatic **classification of CMEs** (supervised)

From science to data science

Data Transformation

CME						Origen del CME								
						FLARE				Filamento				
Date			LASCO onset		Vprop	Coordinates		SXR onset		Observation date (Ha)			Coordinates	
DD	MM	YY	HH	MM		lat	long	HH	MM	DD	MM	YY	lat	long
7	4	97	14	27	592	-28	-11	14	8					
12	5	97	6	30	355	21	8	4	55					
21	6	98	5	35	619	18	39	5	18					
13	4	99	3	30	309	16	0	2	14					
6	6	0	15	54	1557	21	-10	15	26					
20	6	0	9	10	516	-27	38	8	27					
25	7	0	3	30	712	6	8	2	50					
21	7	6	13	54	403					20	7	6	-20	-25
29	12	6	21	30	236					29	12	6	-20	35
23	5	10	18	30	258					21	5	10	15	-20
30	1	11	19	36	521*					30	1	11	25	25
11	1	12	21	36	302					11	1	12	45	-25
16	3	12	20	36	862*					16	3	12	20	60
1	4	12	3	12	410*					31	3	12	20	-20
27	12	12	21	28	267					27	12	12	5	-20

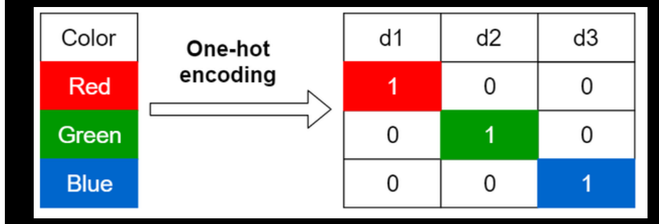
categorical

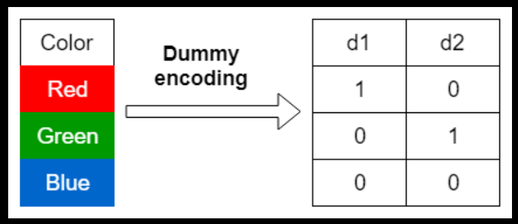
numeric data

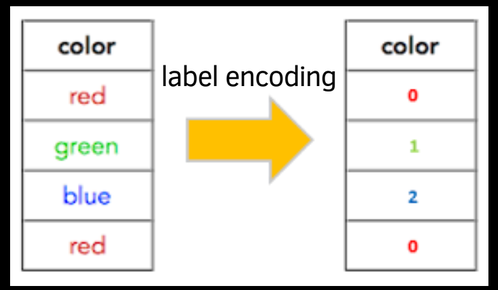
Categorical features (generally):

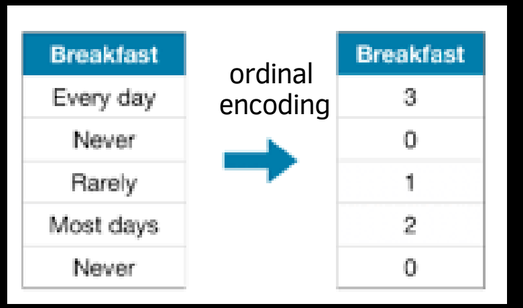
- Binary: Yes, No ; True, False
- B. Ordinal: Specific ordered Groups: low, medium, high; cold, hot, lava Hot
- Nominal: Unordered Groups: cat, dog, tiger -pizza, burger, coke

ENCODING









From science to data science

- The training/test set

	A	B	C
1	SunActivity	vprop	Origin
2	low	592	flare
3	low	355	flare
4	low	619	flare
5	high	309	flare
6	high	1557	flare
7	high	516	flare
8	high	712	flare

To Do:

- data exploration (outliers? nulls?, # samples, balanced? etc)
- data modeling preparation: standarization/normalization, one-hot encoding (or others), etc
- data splitting (70/30?, 80/20?, etc)
- MLP architecture! hyperparameters: #layers, Optimization algorithm, etc
- Fitting the model
- Obtain metrics
- Tunning

Questions

- Which is the scores for the proposed model?
- How confident are we about the model performance/validation?
- Is the dataset representative enough
- propose other questions?!

International Workshop on Machine Learning for Space Weather: Fundamentals, Tools and Future Prospects

7-11 November 2022
This is a hybrid meeting
Buenos Aires, Argentina

Further information:
https://indico.ictp.it/event/9840/
smr3750@ictp.it
+39-040-2240284
Elizabeth Brancaccio

Dra María Graciela Molina
FACET-UNT / CONICET
Tucumán Space Weather Center - TSWC

https://spaceweather.facet.unt.edu.ar/
IG -> @spaceweatherargentina

gmolina@herrera.unt.edu.ar