MBA USP ESALQ

Deep Learning

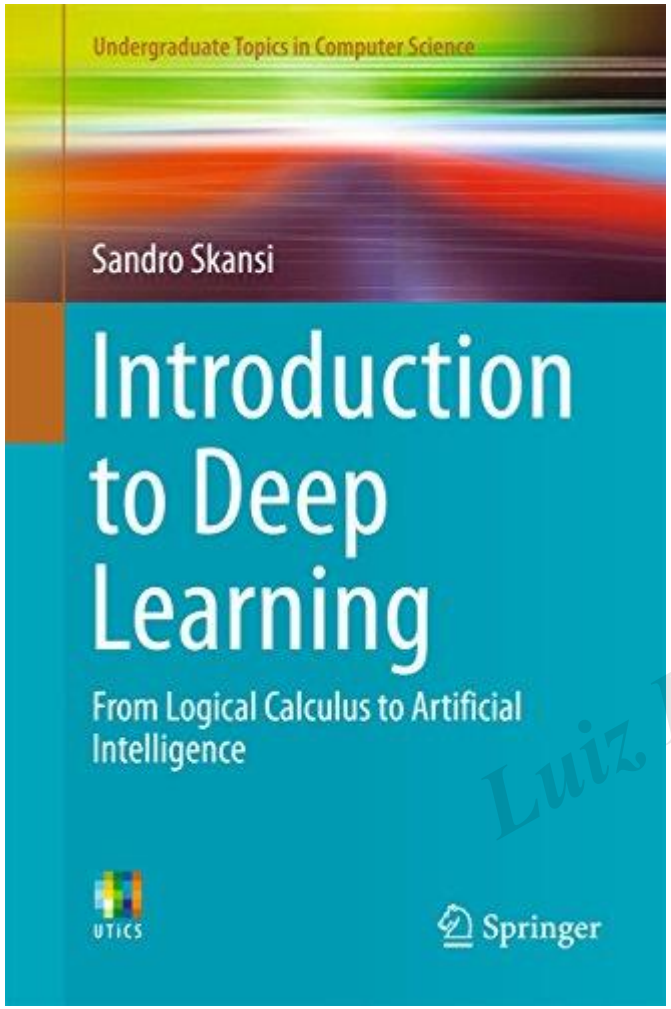
Prof. Dr. Jeronymo Marcondes

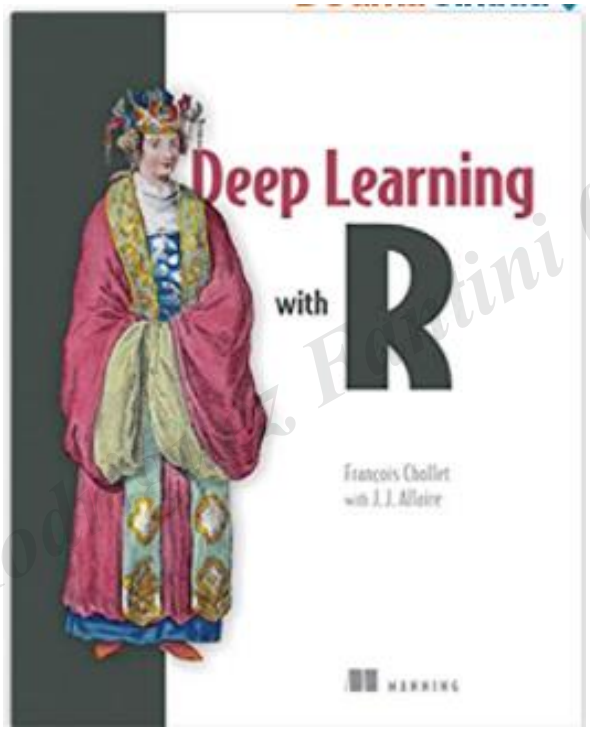
Introducción

- Plan de ataque:

1. El papel de las redes neuronales multicapa
2. El problema del overfitting.
3. El problema de definición de hiperparámetros
4. El problema de encontrar solución óptima

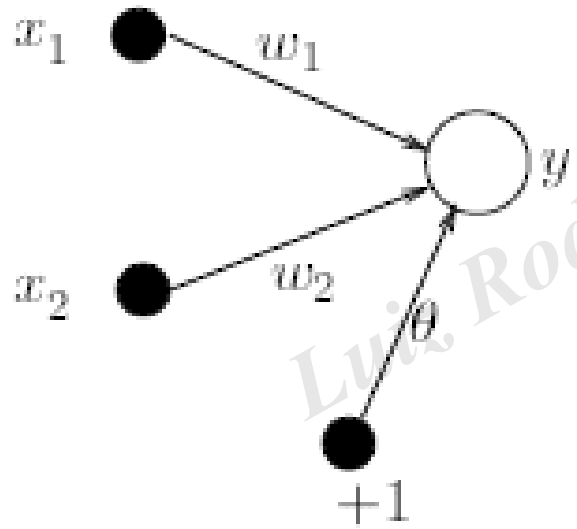
Introducción



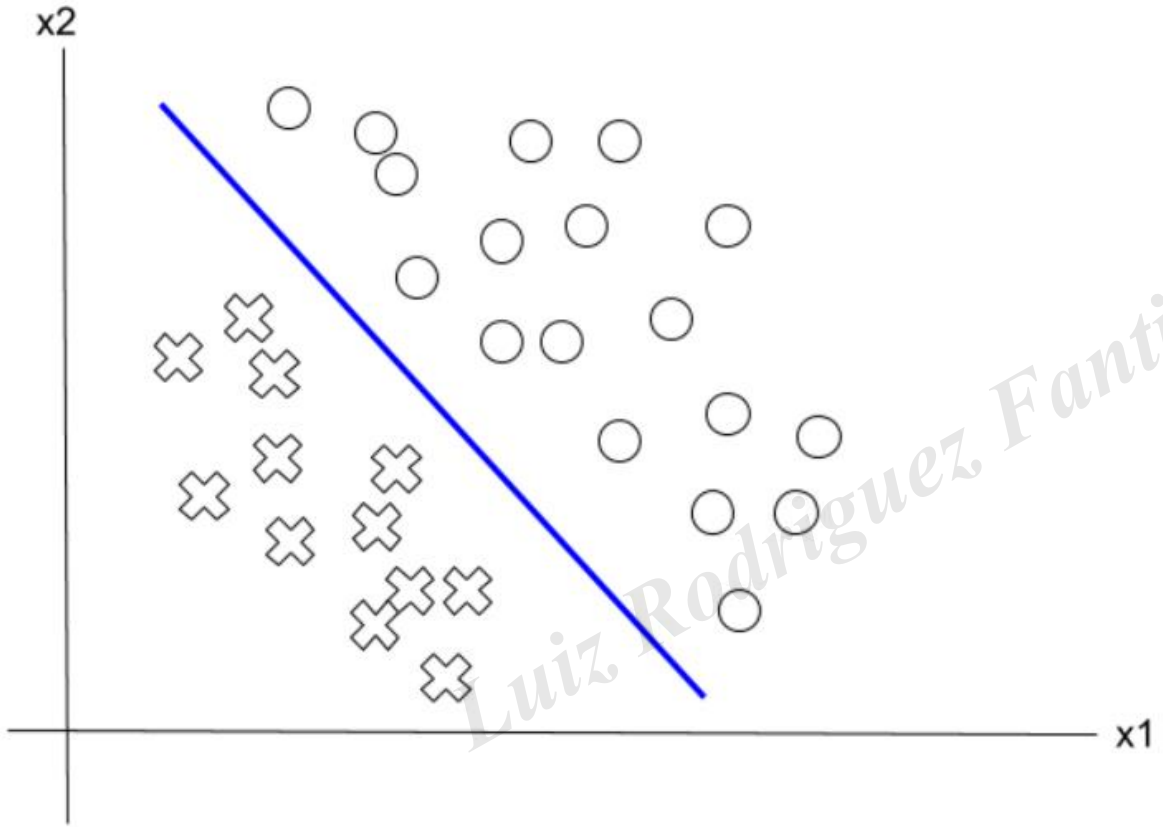


Introducción

- Era negra de las redes neuronales artificiales.



Introducción

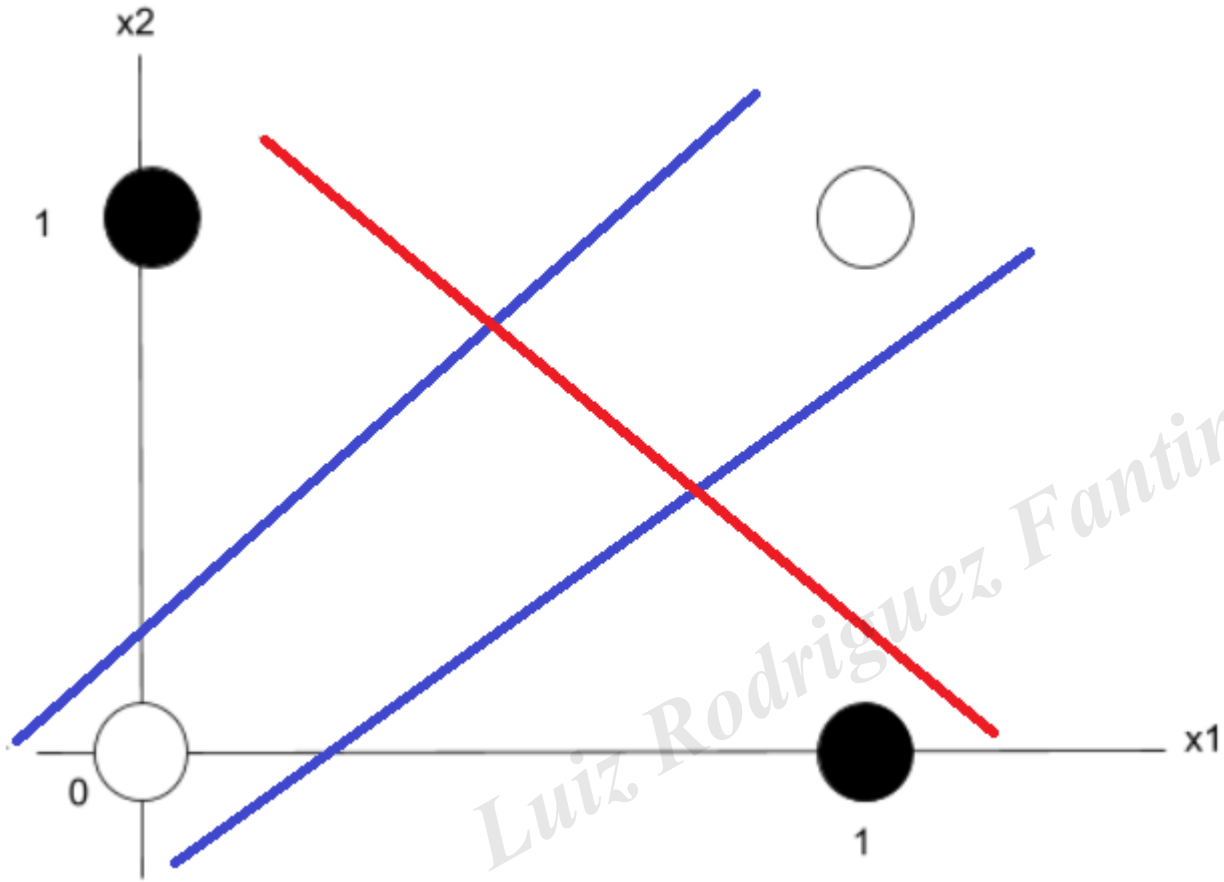


https://automaticaddison.com/linear-separability-and-the-xor-problem/

XoR

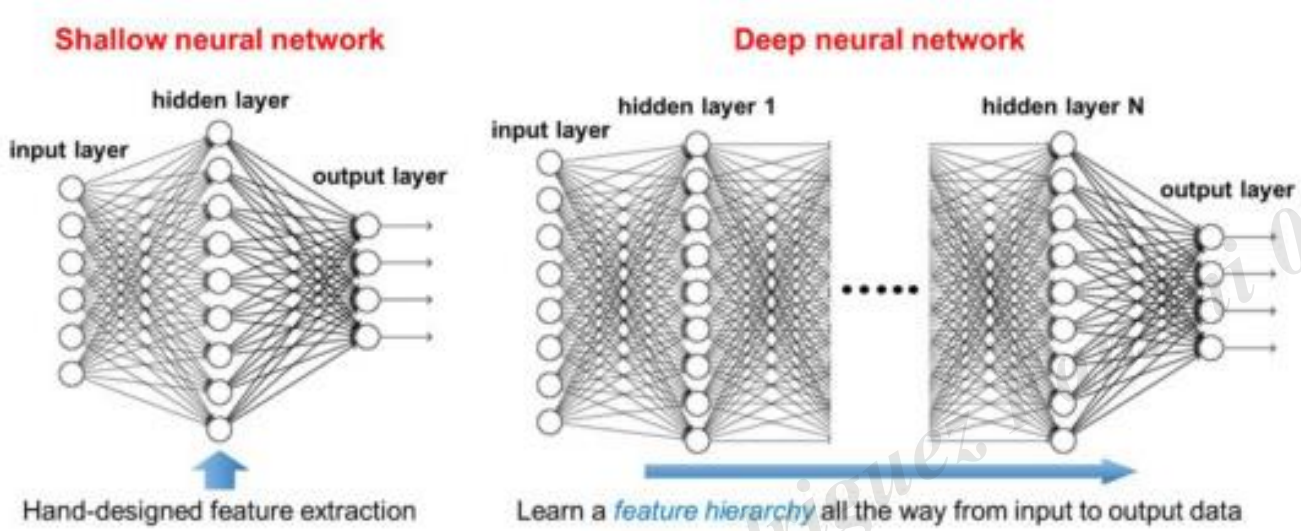
- Exclusive Or

x_0	x_1	d
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1



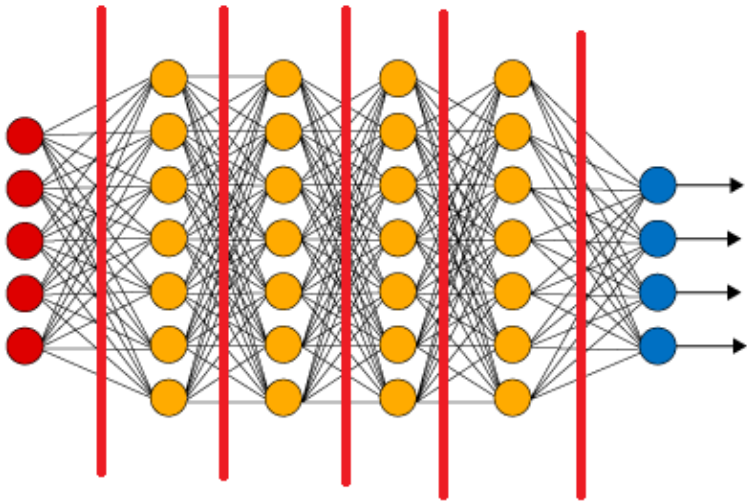
https://automaticaddison.com/linear-separability-and-the-xor-problem/

Red Neuronal Multicapa



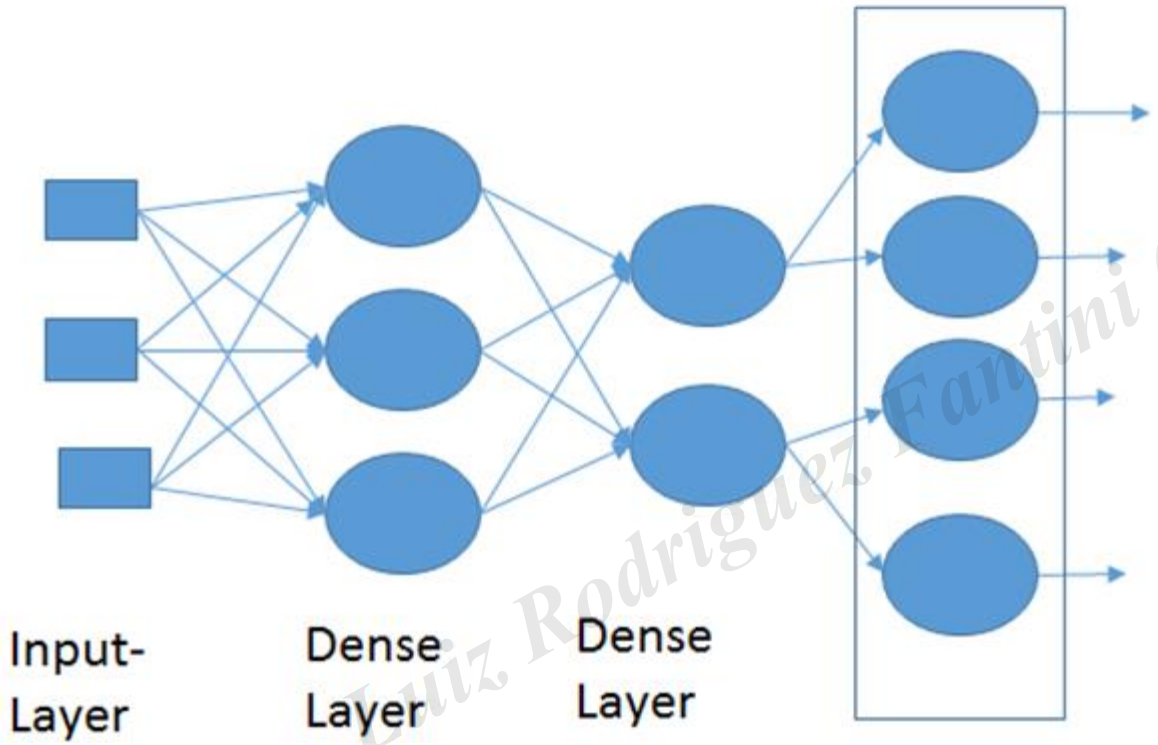
- Muchas capas permiten identificar relaciones no lineales.
- En el caso de que tengamos varías capas intermediarias se obtiene lo que es llamado aprendizaje profundo (deep learning).

Estructura de la Red



- Capas – grupo de neuronas en una etapa del proceso.
- Capa Densa – Conecta cada neurona en una capa con todas las neuronas de la capa anterior. Por ejemplo, si la capa actual tiene 5 neuronas y la capa anterior tiene 3, el total de conexiones es de 15.

Capa Densa



Función Pérdida

- Función que permite verificar lo asertiva que es determinada previsión.
- Predicted x Actual
- Son diferentes para variables continuas y categóricas

Función Pérdida

- Las variables categóricas contienen un número finito de categorías o grupos distintos. Los datos categóricos pueden no tener un orden lógico. Por ejemplo, los predictores categóricos incluyen género, tipo de material y método de pago.
- Variables continuas son variables numéricas que tienen un número infinito de valores entre dos valores cualquiera. Una variable continua puede ser numérica o de fecha/hora. Por ejemplo, la longitud de una pieza o la fecha y hora en que un pago es recibido.
- Variables discretas son variables numéricas que tienen un número contable de valores entre cualquiera de los valores. Una variable discreta siempre es numérica. Por ejemplo, el número de reclamaciones de clientes o el número de fallas o defectos.

https://support.minitab.com/

Algunas funciones de error

- Error cuadrático Medio:

$$EQM = \sum_{n=1}^{k} (Actual - Predicted)^2$$

	Nota	
	Actual	**Predicted**
A	7	8
B	10	9
C	5	10
D	8	8

Algunas funciones de error

- Error Absoluto Medio

$$EQM = \sum_{n=1}^{k} |Actual - Predicted|$$

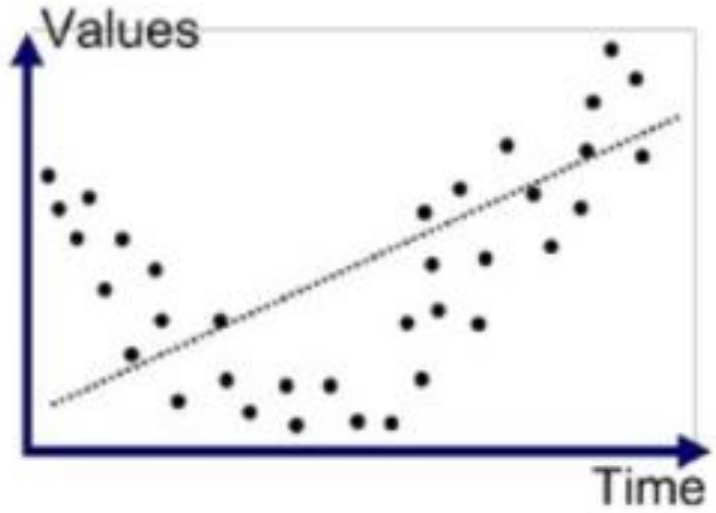
	Nota	
	Actual	**Predicted**
A	7	8
B	10	9
C	5	10
D	8	8

Algunas funciones de error

- Funciones para output continuo.
- Hay necesidad de otras métricas para clasificación, como 1 y 0.
- Binary cross-entropy
- Categorical cross-entropy

Tendencia x Varianza

- ¿Qué estamos buscando?
- ¿Qué puede ocurrir?
- Importancia de la Generalización – teoría del mapa



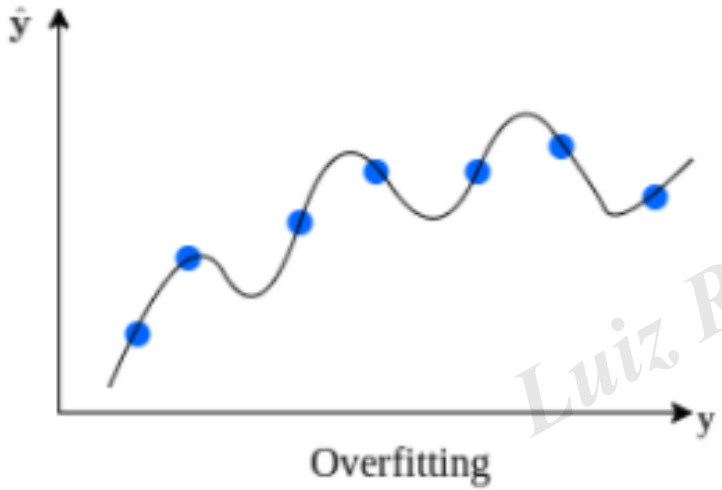
Underfitted

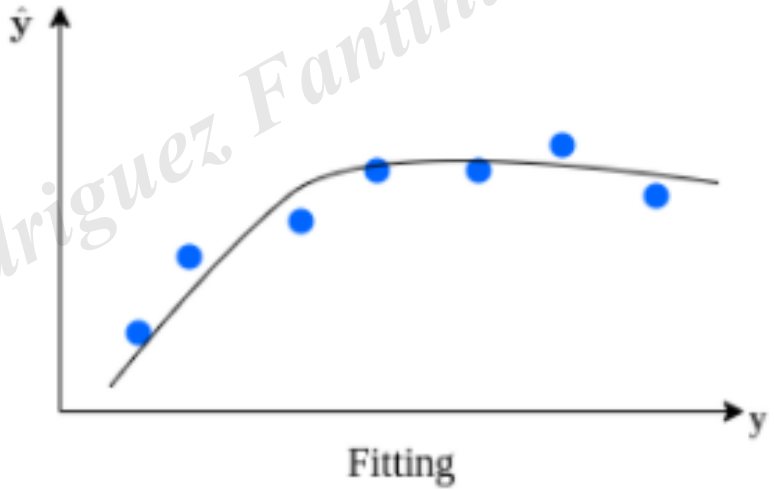
Sesgo x Varianza

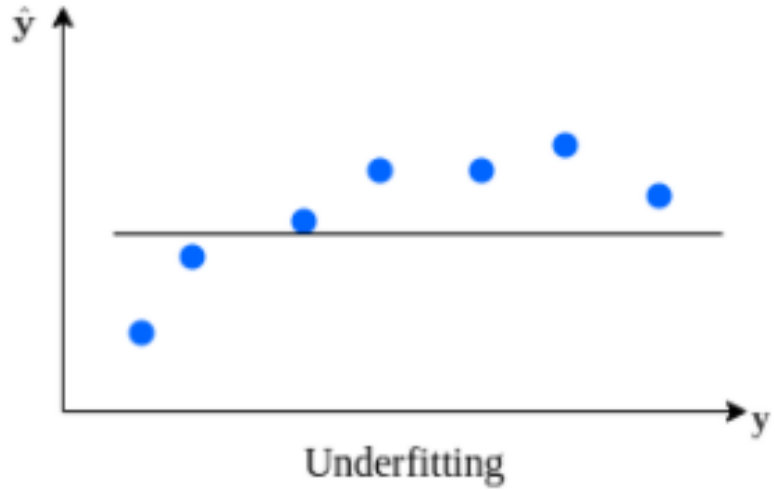
- Ausencia de Sesgo - en la media usted acierta.
- Reducción de Varianza – tiro al blanco.

Sesgo x Varianza

- Preocupación con la varianza.







https://www.baeldung.com/cs/época-neural-networks

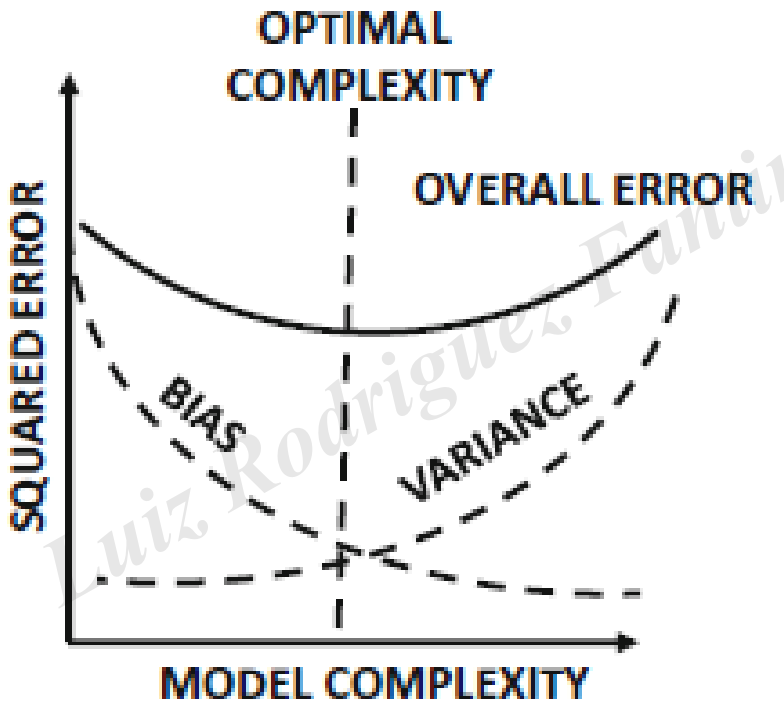
Fitting

- UnderFitting - Su modelo está subyugando los datos de entrenamiento cuando el modelo tiene un desempeño insatisfactorio en los datos de entrenamiento. Eso ocurre porque el modelo es incapaz de capturar la relación entre los ejemplos de entrada (generalmente llamados de X) y los valores de destino (generalmente llamados de Y).

Fitting

- OverFitting - Su modelo está super ajustando sus datos de entrenamiento cuando usted ve que el modelo tiene un buen desempeño en los datos de entrenamiento, pero no en los datos de evaluación. Eso ocurre porque el modelo está memorizando los datos que vio y es incapaz de generalizar para ejemplos no vistos. Muy común en Redes Neurales.

Fitting



Reducción del overfitting

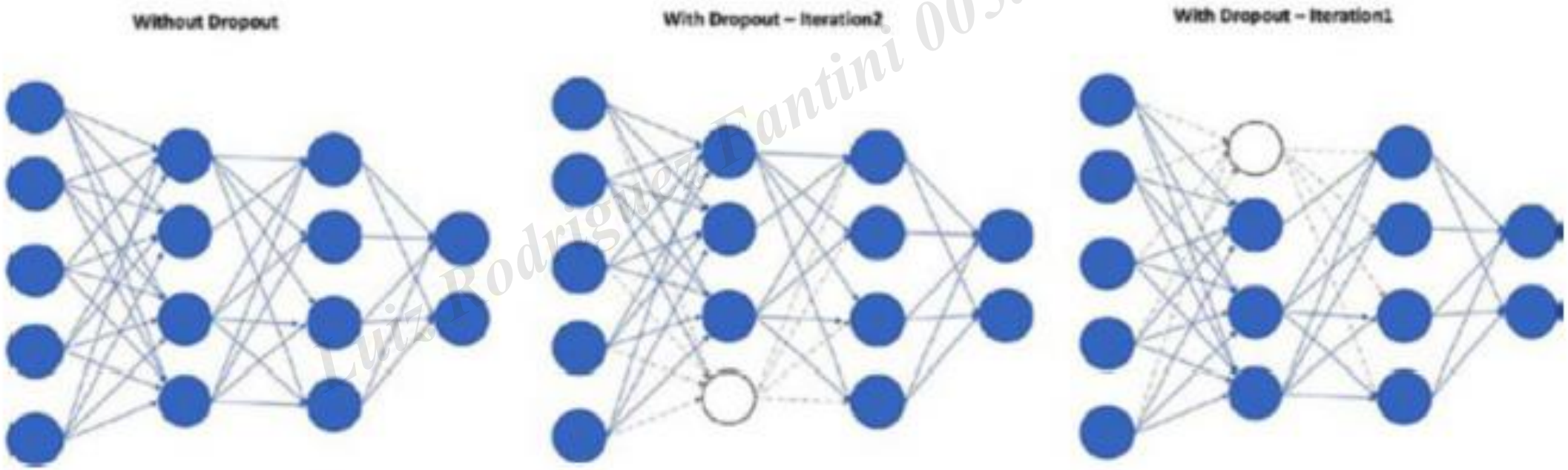
- ¿Cómo reducir overfitting?
- Reducir número de capas
- Más parámetros = + capacidad de memorización
- Regularización

Regularización

- Introducción de “ruido”.
- Dropout

El modelo descarta o desactiva arbitrariamente algunas neuronas para una capa durante cada iteración. En cada iteración, el modelo mira para una estructura ligeramente diferente de sí misma para optimizar (como una pareja de neuronas y las conexiones serían desactivadas). Diga que tenemos dos capas sucesivas, H1 y H2, con 15 y 20 neuronas, respectivamente. La aplicación de la técnica de eliminación entre esas dos capas resultaría en descartar aleatoriamente algunas neuronas (con base en un porcentaje definido) para H1, lo que, por lo tanto, reduce las conexiones entre H1 y H2. Ese proceso se repite para cada iteración con aleatoriedad, por lo tanto, si el modelo tiene que aprender para un lote y actualizó los pesos, el próximo lote puede tener un conjunto bastante diferente de pesos y conexiones para entrenar.

Dropout



L1

- Ajuste de los pesos realizado por medio de la regularización.
- El enfoque no es más en la neurona, pero sí en el valor del peso.
- El peso es ajustado hasta el valor de cero dependiendo de su importancia.

L2

- Ajuste de los pesos realizado por medio de la regularización.
- El enfoque no es más en la neurona, pero sí en el valor del peso.
- El peso es ajustado hasta un valor próximo de cero dependiendo de su importancia.

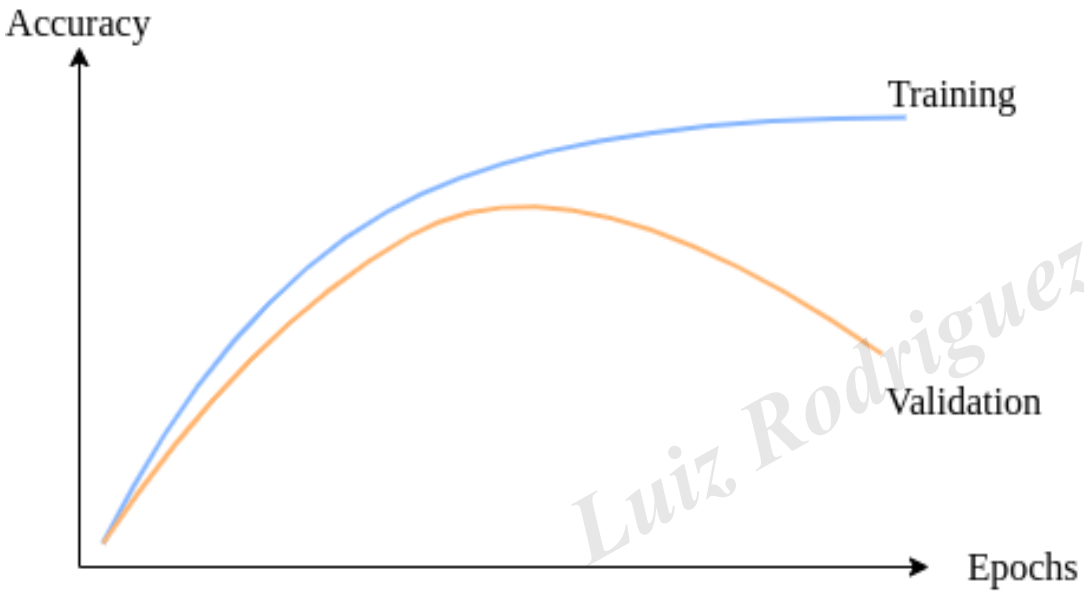
Early Stopping

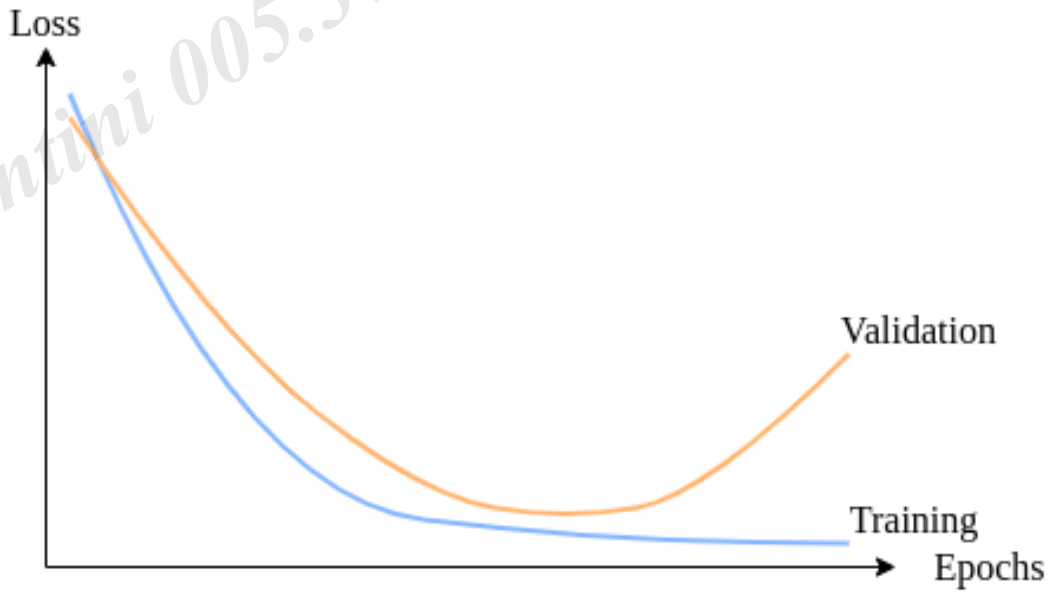
- Final de cada época – verificar mejoría.
- ¿La mejoría no está sucediendo más? ¡Alto!
- Eso evita el overfitting.

Épocas

- ¿Cómo verificar si tenemos overffitting?
- Cuántas veces los datos pasan por la red.
- Cada época es un paso en dirección al resultado “genial”.
- Convergencia.

Épocas





https://www.baeldung.com/cs/época-neural-networks

Batch

- ¿Cómo realizar el entrenamiento?
- Técnicas que pueden ayudar en el proceso de optimización.
- Uso de batch (lotes).

Batch

Normalmente, el entrenamiento sería realizado en lotes debido a restricciones de memoria en el sistema. Un lote es una colección de muestras de entrenamiento de toda la entrada. La red actualiza sus pesos después de procesar todas las muestras en un lote. Eso es llamado de iteración (o sea, un paso exitoso de todas las muestras en un lote, seguido por una actualización de peso en la red). La computación de todas las muestras de entrenamiento proporcionadas en los datos de entrada con actualizaciones de peso lote por lote es llamado de época. En cada iteración, la red aprovecha la función del optimizador para hacer un pequeño cambio para sus parámetros de peso (que fueron inicializados aleatoriamente en el inicio) para mejorar la previsión final, reduciendo la función de pérdida. Paso a paso, con varias iteraciones y, a continuación, varias épocas, la red actualiza su ponderación y aprende a hacer una previsión correcta para el entrenamiento dado muestras.

Hiperparámetros

- ¿Qué son?
- Cualquier número utilizado por la red que no es aprendido.
- ¿Cómo determinar esos valores?
- ¿Cross validation? ¿Grid Search?

Hiperparámetros

- ¿Vamos a recordar algunos?

1. Learning rate
2. Batch size
3. Epochs
4. Función de activación

AUTOML

- Vamos a discutir un poco sobre eso.

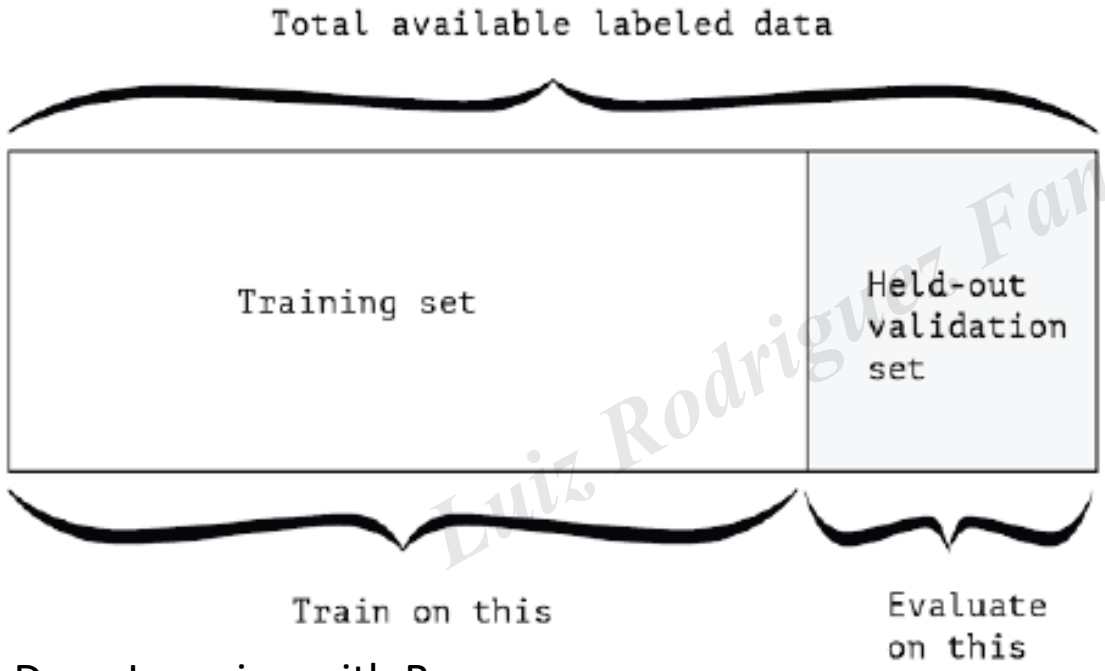
- AutoKeras, H20.

- Transfer Learning.

- Grid Search

Cross validation

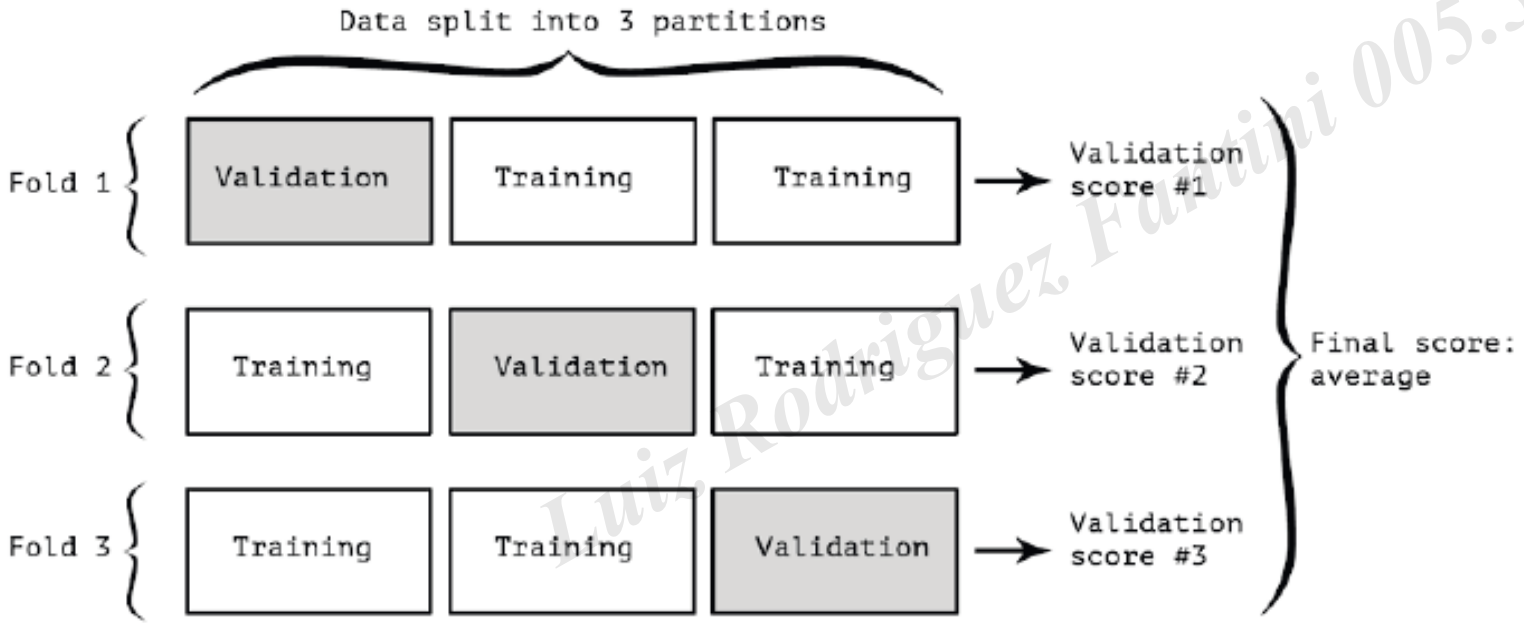
- ¿Cómo funciona?
- Conjunto de entrenamiento, prueba y validación



Deep Learning with R

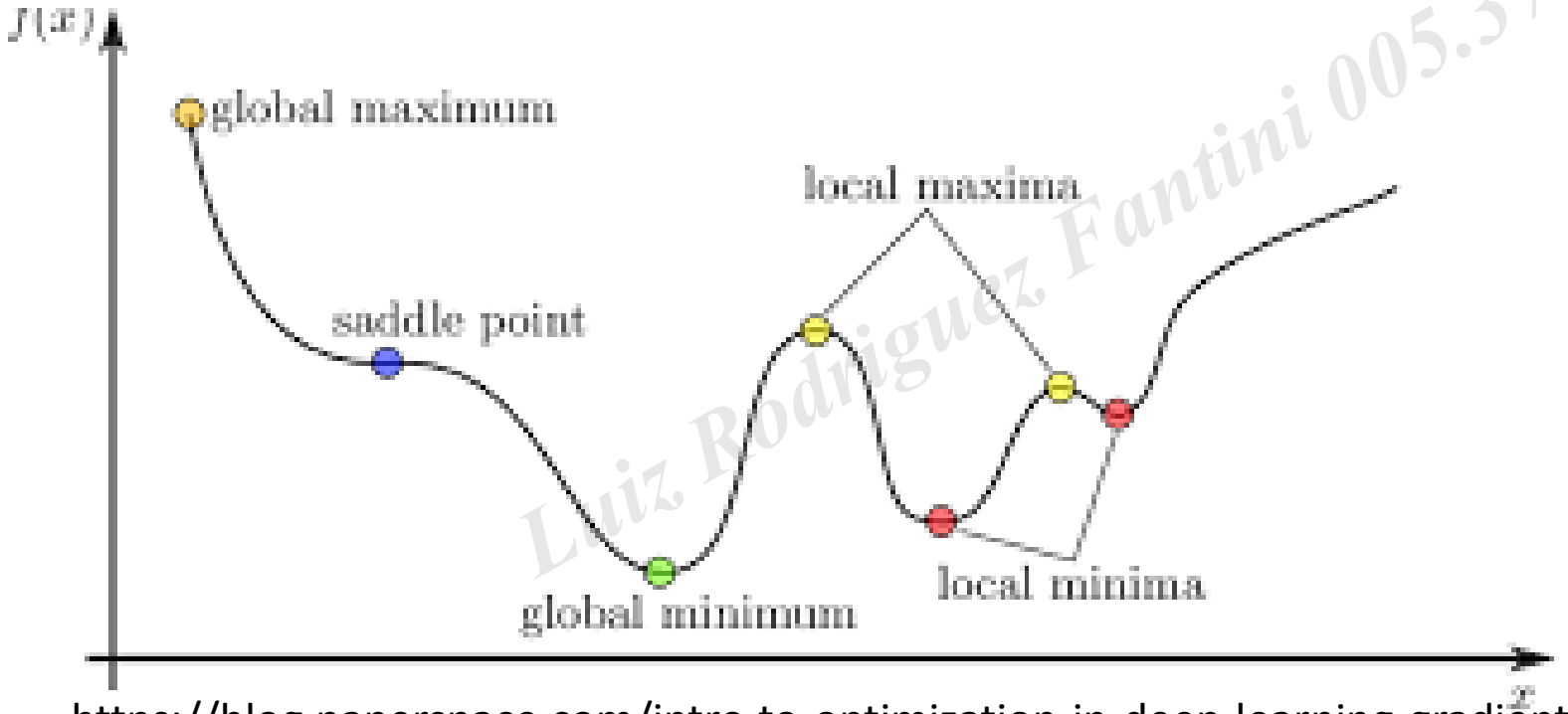
Cross validation

- K fold cross validation



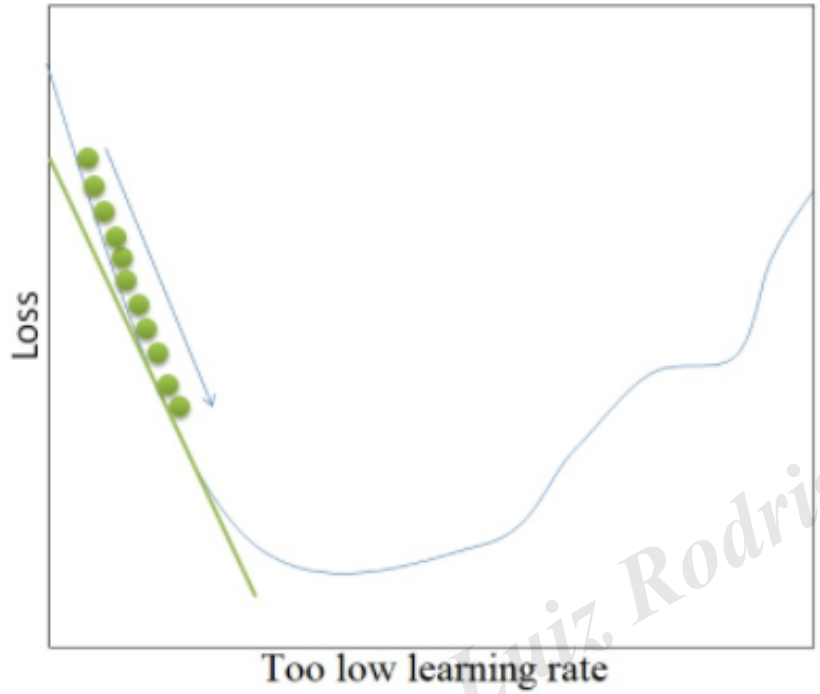
Deep Learning with R

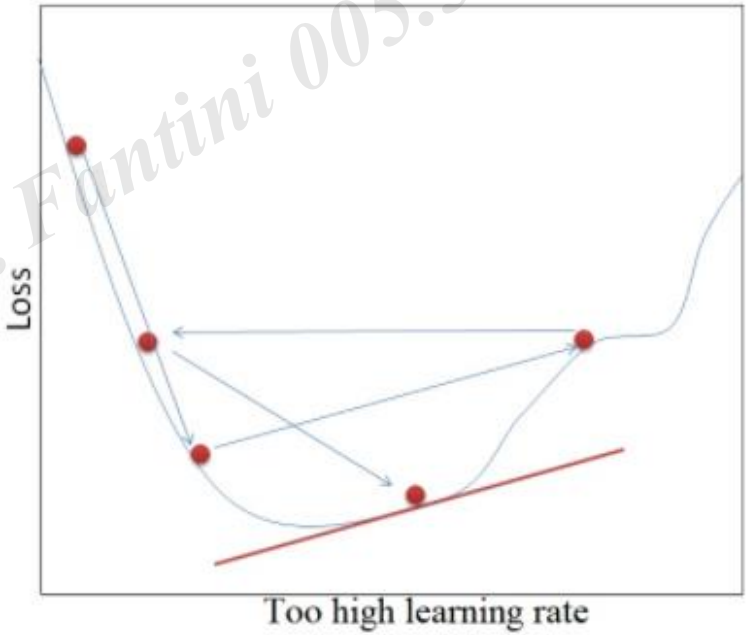
Mínimo local y punto de montaje



https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Mínimo local y punto de montaje





https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/

Optimizadores

- ¿O sea, dada la función pérdida, como actualizar los pesos?
- ¿Cuál es la regla?
- Nosotros utilizamos el modelo clásico de descenso del gradiente, pero hay otras.

Optimizadores

- ADAGRAD
- ADAM
- RMSPROP

Momentun

- También llamada de inercia
- Concepto próximo al de inercia = fricción
- Figura arriba – la bolita con velocidad podría pasar el punto de montaje y llegar en el mínimo.

Momentun

$$w_i^{new} = w_i^{old} - \eta \frac{\partial E}{\partial w_i^{old}} + \mu(|w_i^{old} - w_i^{older}|)$$

Introduction to deep Learning

https://www.linkedin.com/in/jeronymo-marcondes-585a26186