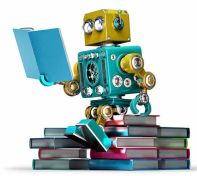
Clasificación Lineal y Evaluación del Desempeño

Inteligencia Artificial



Marco Teran

2021 - Bogotá

Contenido

Entrenamiento

Optimización de la perdida

Queremos encontrar los pesos de la red que **logren la menor pérdida**

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

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

Optimización de la perdida

Queremos encontrar los pesos de la red que **logren la menor pérdida**

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

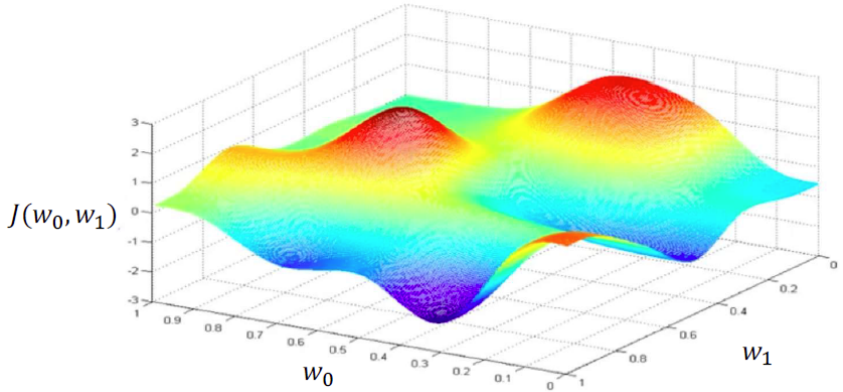
$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}} J(\mathbf{W})$$

Recuerda:

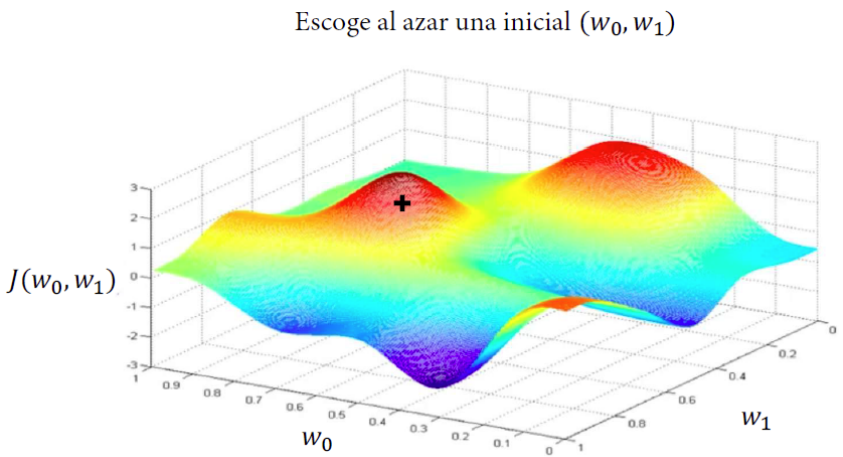
$$\mathbf{W} = \{\mathbf{W}^{(0)}, \mathbf{W}^{(1)}, ...\}$$

Optimización de la perdida

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

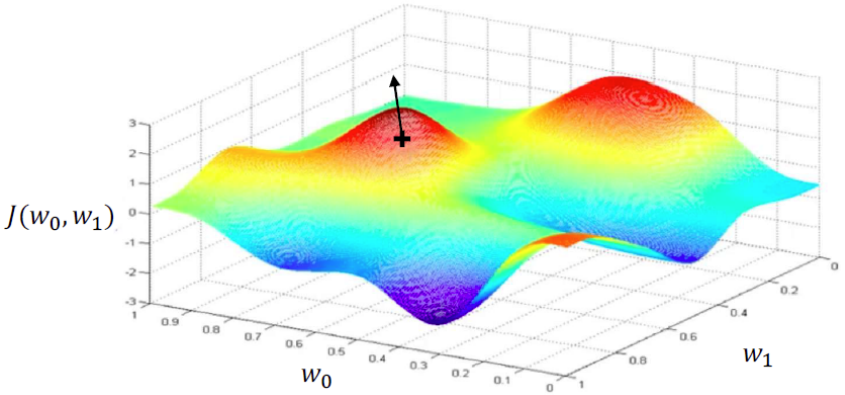


Optimización de la perdida



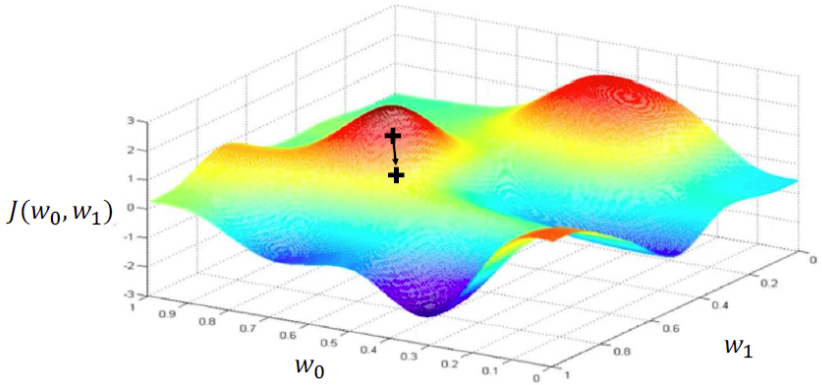
Optimización de la perdida

Calculo del gradiente, $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$



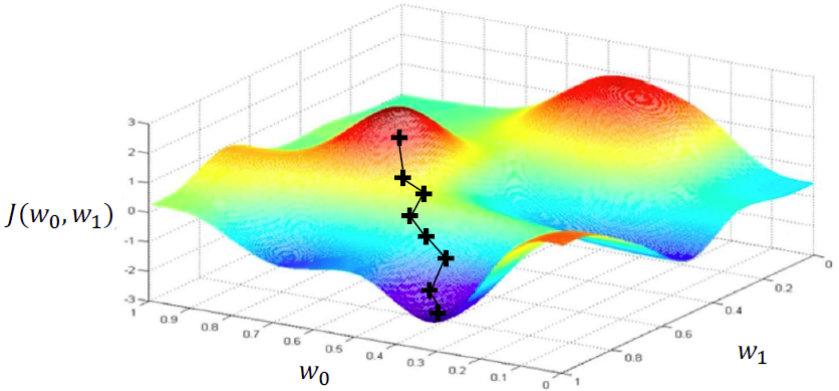
Optimización de la perdida

De un pequeño paso en dirección opuesta al gradiente



Gradiente descendente

Repita hasta la convergencia



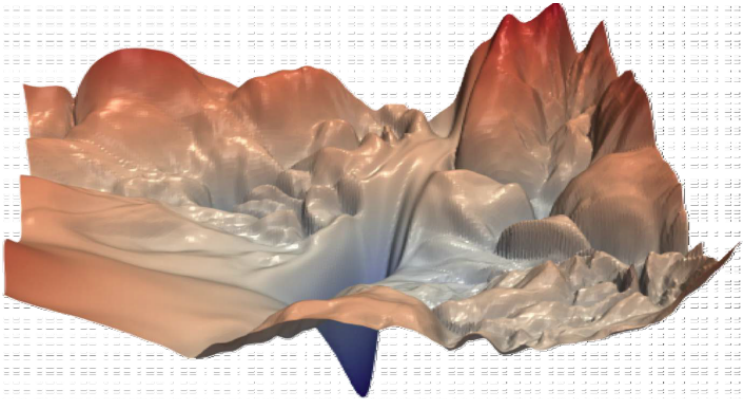
Gradiente descendente

Algoritmo:

- Iniciar los pesos al azar $\sim \mathcal{N}(0, \sigma^2)$
- Bucle hasta la convergencia:
  - Calcular el gradiente $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
  - Actualizar los pesos $\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
- Devuelve los pesos

Optimización

El entrenamiento es complejo



"Visualizando el paisaje de pérdida de las redes neuronales". Diciembre de 2017.

Las funciones de pérdida pueden ser difíciles de optimizar

Recuerde: Optimización a través del descenso del gradiente

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

Las funciones de pérdida pueden ser difíciles de optimizar

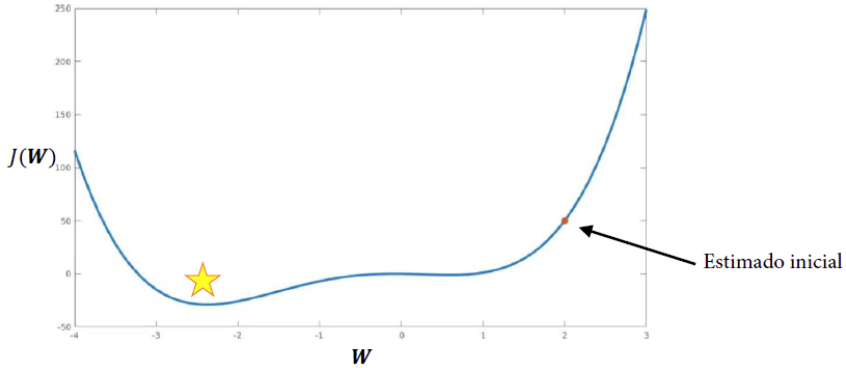
Recuerde: Optimización a través del descenso del gradiente

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

¿Cómo podemos establecer la tasa de aprendizaje? *(learning rate)*

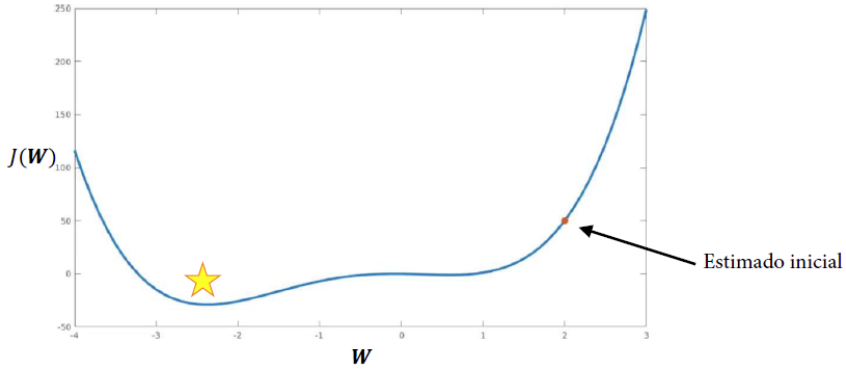
Ajuste de la tasa de aprendizaje

Recuerde: Una pequeña tasa de aprendizaje converge lentamente y se atasca en falsos mínimos locales



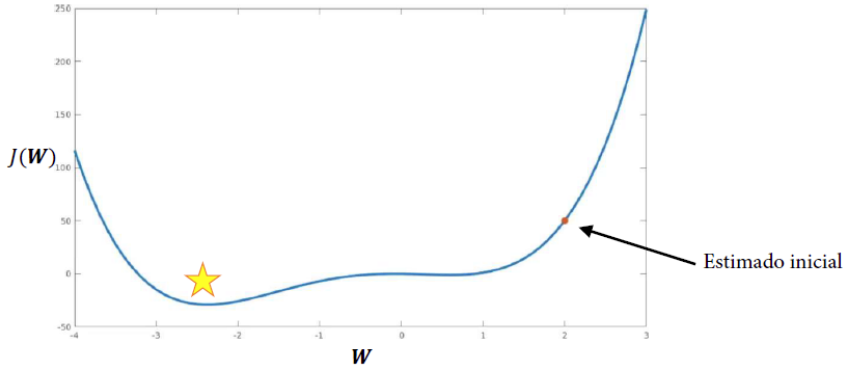
Ajuste de la tasa de aprendizaje

Recuerde: Las grandes tasas de aprendizaje se sobrepasan, se vuelven inestables y divergen



Ajuste de la tasa de aprendizaje

Recuerde: Las tasas de aprendizaje estables convergen sin problemas y evitan los mínimos locales



¿Cómo se puede hacer frente a esto?

- **Idea 1:** Intentar muchas tasas de aprendizaje diferentes y ver cuál funciona "bien".
- **Idea 2:** ¡Haz algo más inteligente! Diseñar una tasa de aprendizaje adaptativo que se "adapte" al paisaje

Tasas de aprendizaje adaptativas

- Las tasas de aprendizaje ya no son fijas
- Pueden hacerse más grandes o más pequeñas dependiendo de:
  - de cuán grande sea el gradiente
  - lo rápido que se está aprendiendo
  - tamaño de pesos particulares
  - etc...

Algoritmos de tasas de aprendizaje adaptativas

- Momentum
- Adagrad
- Adadelta
- RMSProp

Detaller adicionales: http://ruder.io/optimizing-gradient-descent/

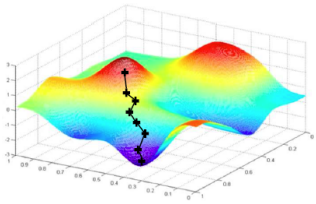
Mini-lotes

Gradiente descendente

Algoritmo:

- Iniciar los pesos al azar $\sim \mathcal{N}(0, \sigma^2)$
- Bucle hasta la convergencia:
  - Calcular el gradiente $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
  - Actualizar los pesos $\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
- Devuelve los pesos

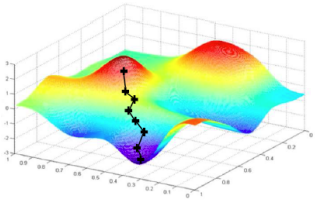
Difícil de calcular $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$



Gradiente descendente estocástico

Algoritmo:

- Iniciar los pesos al azar $\sim \mathcal{N}(0, \sigma^2)$
- Bucle hasta la convergencia:
  - Tomar un solo punto i
  - Calcular el gradiente $\dfrac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}}$
  - Actualizar los pesos $\mathbf{W} \leftarrow \mathbf{W} - \eta \dfrac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
- Devuelve los pesos

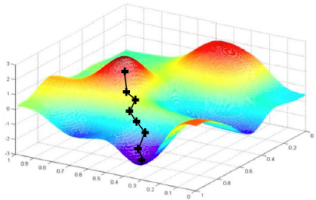


Fácil de calcular $\dfrac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$ pero muy ruidoso (estocástico)

Gradiente descendente en mini-lotes

Algoritmo:

- Iniciar los pesos al azar $\sim \mathcal{N}(0, \sigma^2)$
- Bucle hasta la convergencia:
  - Tomar un lote de puntos B
  - Calcular el gradiente $\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = \frac{1}{B} \sum_{k=1}^{B} \frac{\partial J_k(\mathbf{W})}{\partial \mathbf{W}}$
  - Actualizar los pesos $\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
- Devuelve los pesos

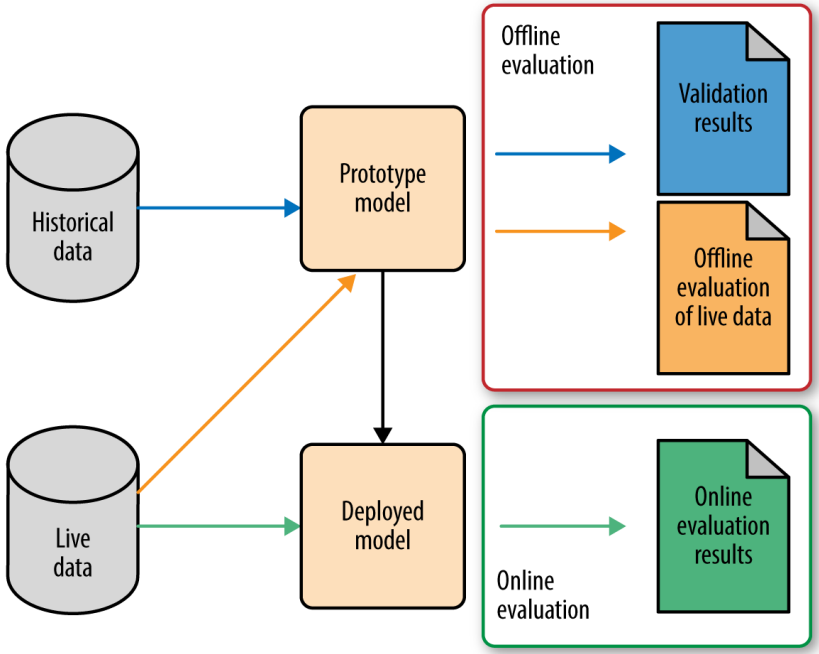


Rápido de calcular y una estimación mucho mejor del verdadero gradiente

Mini-batches durante el entrenamiento

- Estimación más precisa del gradiente:
  - Una convergencia más suave
  - Permite mayores tasas de aprendizaje
- Los mini lotes conducen a un rápido entrenamiento
  - Puede paralelizar la computación + lograr aumentos significativos de velocidad en las GPU

Métricas de evaluación


Offline evaluation
Validation results
Offline evaluation of live data
Historical data
Prototype model
Live data
Deployed model
Online evaluation
Online evaluation results

Test Image from X_test

TRAINED MODEL

DOG

Prediction on Test Image

DOG == DOG?

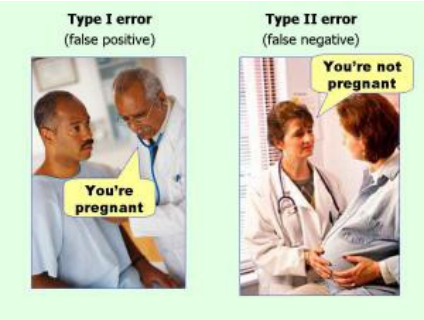
CAT == DOG?

Compare Prediction to Correct Label

DOG

Correct Label from y_test

Tipos de error



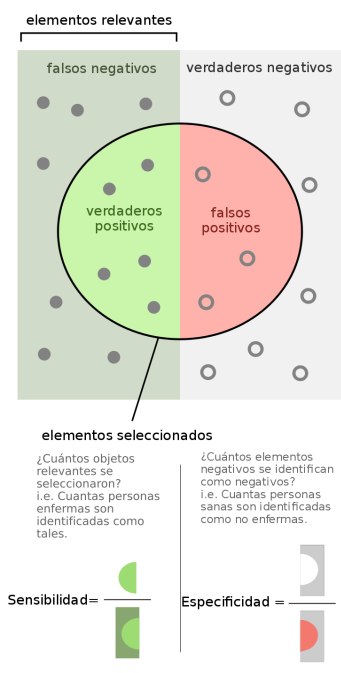
Matriz de confusión

		Actual Value: Positive	Actual Value: Negative
Predicted Value	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

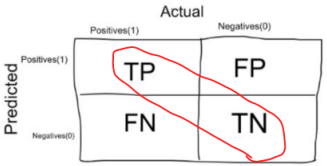
- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Tabla de métricas

Metric	Formula
True positive rate, recall	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
False positive rate	$\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
F-measure	$\frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$

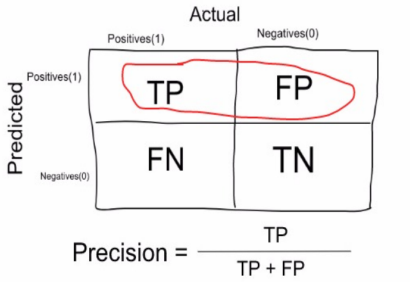

elementos relevantes
falsos negativos
verdaderos negativos
verdaderos positivos
falsos positivos
elementos seleccionados
¿Cuántos objetos relevantes se seleccionaron?
i.e. Cuantas personas enfermas son identificadas como tales.
¿Cuántos elementos negativos se identifican como negativos?
i.e. Cuantas personas sanas son identificadas como no enfermas.
Sensibilidad=
Especificidad =

Accuracy

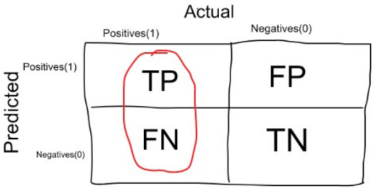


$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision


Actual
Positives(1)
Negatives(0)
Predicted
Positives(1)
Negatives(0)
TP
FP
FN
TN
Precision = TP / TP + FP

Recall o sensibilidad



$$\text{Recall} = \frac{TP}{TP + FN}$$

Tabla de métricas

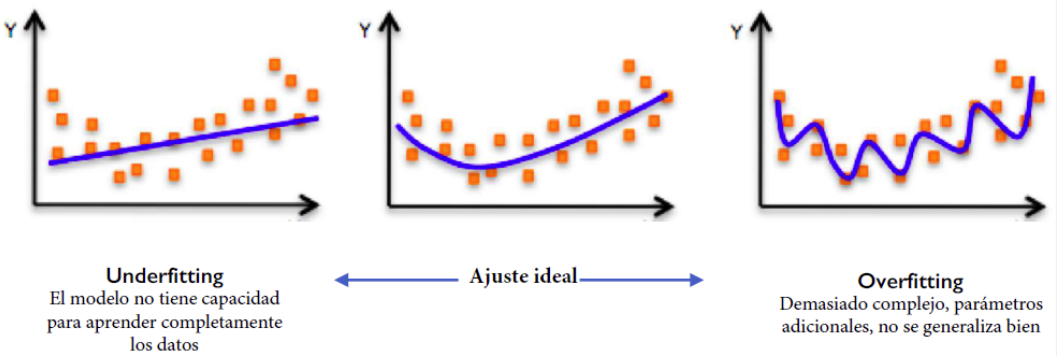
Metric	Formula
True positive rate, recall	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
False positive rate	$\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
F-measure	$\frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$

Ejemplo

		True condition	
	Total population = 1000	Cancer Negative (N) 0	Cancer Positive (P) 1
Predicted Condition	Cancer Negative (N) - 0	850 TN	6 FN
	Cancer Positive(P) - 1	50 FP	94 TP

Sobreajuste (overfitting)

El problema del sobreajuste



Regularización

Regularización

- **¿Qué es?** Técnica que limita nuestro problema de optimización para no incentivar la generación de modelos complejos
- **¿Por qué lo necesitamos?** Mejorar la generalización de nuestro modelo sobre datos no vistos

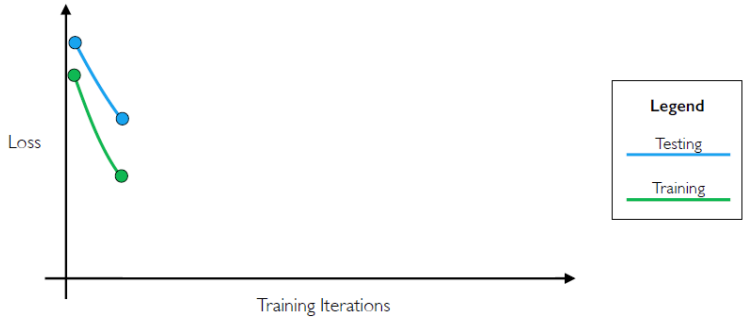
Regularización: Parada temprana

Early Stopping

- Detener el entrenamiento antes de empezar a sobreajustar...

Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...

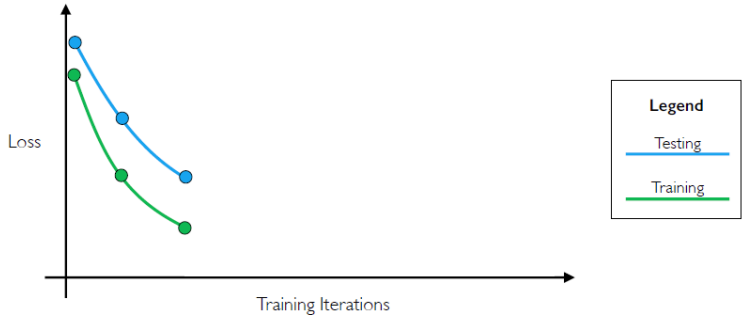


Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...

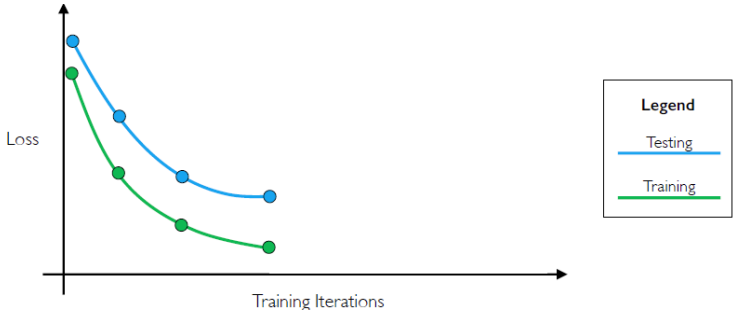
Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...



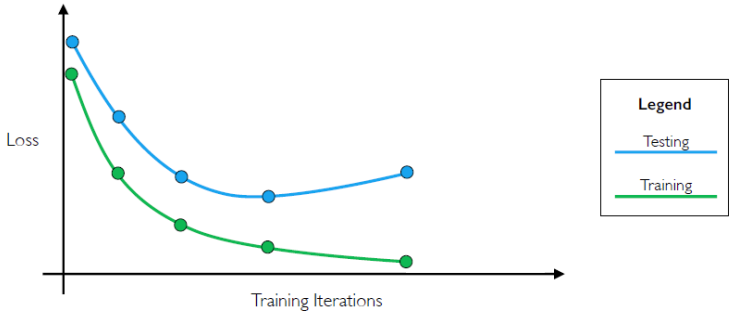
Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...



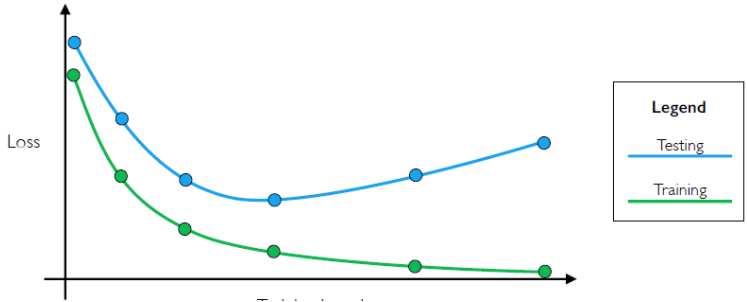
Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...



Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...



Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...

Regularización: Parada temprana

- Detener el entrenamiento antes de empezar a sobreajustar...

Muchas gracias por su atención

¿Preguntas?

Contacto: Marco Teran
webpage: marcoteran.github.io/