

Detección de cáncer mediante deep learning

Sofía Vargas Ibarra

Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid

22 de julio de 2020

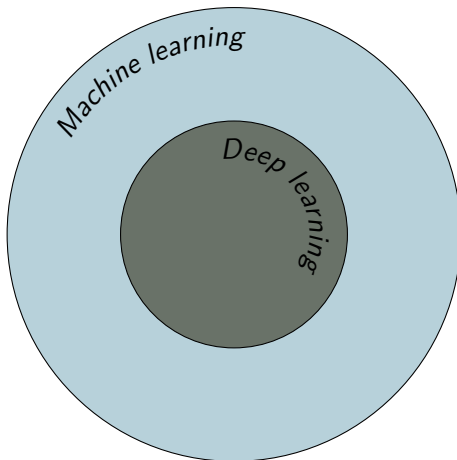
Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Introducción y objetivos



Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Estado del arte

■ Detección de objetos [1].

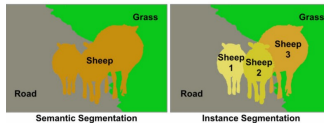


Figura: Segmentación semántica y de instancias

Estado del arte

■ Detección de objetos [1].

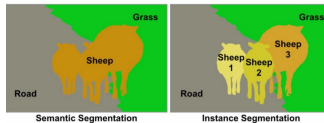


Figura: Segmentación semántica y de instancias

■ *Generative adversarial networks* (GANs)[2].

Estado del arte

- Detección de objetos [1].

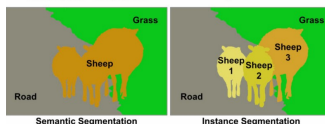


Figura: Segmentación semántica y de instancias

- *Generative adversarial networks* (GANs)[2].
- *Transfer learning* [3].

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 **Conceptos previos**
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 **Conceptos previos**
 - **Conceptos de imágenes**
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Definición

Una imagen digital es una disposición de distintos valores en distintos lugares (píxeles).

Una imagen digital es una disposición de distintos valores en distintos lugares (píxeles).

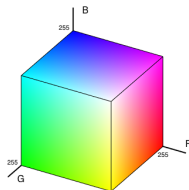


Figura: Estructura de imagen (RGB)

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 **Conceptos previos**
 - Conceptos de imágenes
 - **Red neuronal *feed-forward***
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Red neuronal *feed-forward*

Neuronal: su estructura está inspirada en el funcionamiento de las redes neuronales biológicas.

Red neuronal *feed-forward*

Neuronal: su estructura está inspirada en el funcionamiento de las redes neuronales biológicas.

Feed-forward: la información fluye únicamente hacia delante.

Red neuronal *feed-forward*

- Aprendizaje automático y supervisado.

Red neuronal *feed-forward*

- Aprendizaje automático y supervisado.
- Se trata de encontrar unos pesos W para aproximar una función $y = f(x; W)$.

Red neuronal *feed-forward*

- Aprendizaje automático y supervisado.
- Se trata de encontrar unos pesos W para aproximar una función $y = f(x; W)$.
- Las capas son vectores formados por neuronas (denotamos por a_j^L a la neurona j en la capa L).

Red neuronal *feed-forward*

- Aprendizaje automático y supervisado.
- Se trata de encontrar unos pesos W para aproximar una función $y = f(x; W)$.
- Las capas son vectores formados por neuronas (denotamos por a_j^L a la neurona j en la capa L).

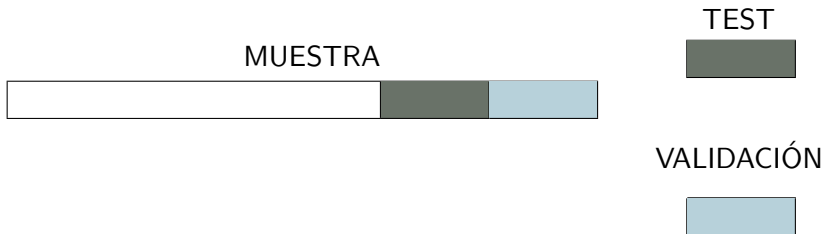
Definición

Dados unos pesos W_j , una constante b^{L-1} y una función de activación g . Se define la neurona a_j^L como:

$$a_j^L = g(W_j^T a^{L-1} + b^{L-1}) \quad (1)$$

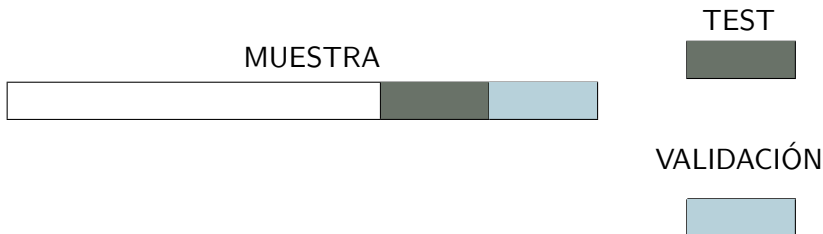
Red neuronal *feed-forward*

Aprendizaje



Red neuronal *feed-forward*

Aprendizaje

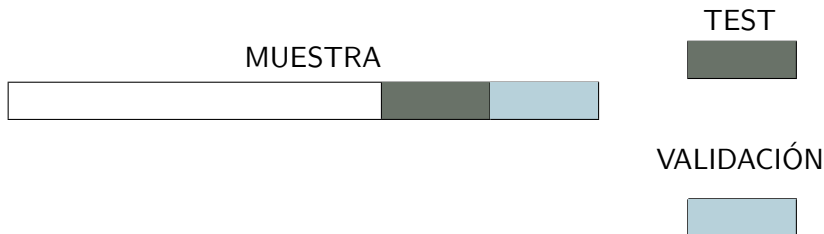


Hiperparámetro

Parámetro que se fija antes de comenzar el proceso de aprendizaje.

Red neuronal *feed-forward*

Aprendizaje



Hiperparámetro

Parámetro que se fija antes de comenzar el proceso de aprendizaje.

Tamaño del *batch*

Número de observaciones que se toman en cada paso.

Red neuronal *feed-forward*

Estandarización

Consiste en ajustar las distintas variables observadas para que estén en la misma escala.

Red neuronal *feed-forward*

Estandarización

Consiste en ajustar las distintas variables observadas para que estén en la misma escala.

- Por el rango de variable.

Red neuronal *feed-forward*

Estandarización

Consiste en ajustar las distintas variables observadas para que estén en la misma escala.

Regularización

Se utiliza para prevenir el sobreajuste para que se tengan buenos resultados para nuevas observaciones, no solo para el conjunto de entrenamiento.

- Por el rango de variable.

Red neuronal *feed-forward*

Estandarización

Consiste en ajustar las distintas variables observadas para que estén en la misma escala.

- Por el rango de variable.

Regularización

Se utiliza para prevenir el sobreajuste para que se tengan buenos resultados para nuevas observaciones, no solo para el conjunto de entrenamiento.

- Aumentar la muestra.
- Penalización de parámetros.
- *Dropout*.

Red neuronal *feed-forward*

Arquitectura

- Hiperparámetros.
- Función de activación.
- Inicialización.
- Error.
- Método de optimización.

Red neuronal *feed-forward*

Arquitectura

Hiperparámetros: número de capas, número de neuronas.

Red neuronal *feed-forward*

Arquitectura

Hiperparámetros: número de capas, número de neuronas.

Función de activación:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$\text{Relu}(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases} \quad (2)$$

$$\text{LeakyRelu}(x) = \begin{cases} 0,01x & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases}$$

Red neuronal *feed-forward*

Arquitectura

Inicialización: Xavier Uniform.

$$W \sim U\left(\frac{-\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right) \text{ donde } n_j \text{ es el tamaño de la capa } j. \quad (3)$$

Red neuronal *feed-forward*

Arquitectura

Inicialización: Xavier Uniform.

$$W \sim U\left(\frac{-\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right) \text{ donde } n_j \text{ es el tamaño de la capa } j. \quad (3)$$

Observación

“Utilizar técnicas concretas para inicializar los pesos se puede considerar como una técnica de regularización al llegar a mínimos locales que generalizan mejor” [4].

Red neuronal *feed-forward*

Arquitectura

Error: entropía cruzada binaria [3] :

p_i : output.

y_i : etiqueta conocida.

g : función de activación.

Red neuronal *feed-forward*

Arquitectura

Error: entropía cruzada binaria [3] :

p_i : output.

y_i : etiqueta conocida.

g : función de activación.

$$J = \frac{-1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (4)$$

Red neuronal *feed-forward*

Arquitectura

Error: entropía cruzada binaria [3] :

p_i : output.

y_i : etiqueta conocida.

g : función de activación.

$$J = \frac{-1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (4)$$

Propagación del error:

$$J^L = J^{L+1} \cdot W^L \cdot \frac{\partial g(x)}{\partial x} \quad (5)$$

Red neuronal *feed-forward*

Arquitectura

Optimización: Mini-Batch Descenso del Gradiente.

α : tasa de aprendizaje.

$J(x)$: error en un punto.

m : tamaño del *batch*.

Red neuronal *feed-forward*

Arquitectura

Optimización: Mini-Batch Descenso del Gradiente.

α : tasa de aprendizaje.

$J(x)$: error en un punto.

m : tamaño del *batch*.

$$w_{k+1} = w_k - \frac{\alpha}{m} \sum_{i=1}^m \nabla J_i(x) \quad (6)$$

Red neuronal *feed-forward*

Arquitectura

α → salta mínimos locales.

α → gran coste computacional.

Red neuronal *feed-forward*

Arquitectura

$\alpha \rightarrow$ salta mínimos locales.

$\alpha \rightarrow$ gran coste computacional.

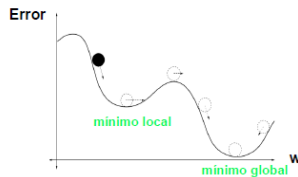


Figura: Tasa de aprendizaje

Observación

Existen métodos adaptativos donde α varía a lo largo de las iteraciones, como puede ser el método Adam.

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 **Conceptos previos**
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - **Red neuronal de convolución**
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Red neuronal de convolución

Motivación

| Interacciones sparse. | Compartir parámetros. | Equivarianza. |

Red neuronal de convolución

Motivación

| Interacciones sparse. | Compartir parámetros. | Equivarianza. |

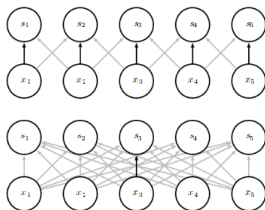
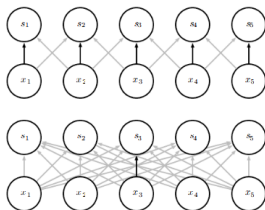


Figura: CNN (superior) y NN (inferior)

Red neuronal de convolución

Motivación

| Interacciones sparse. | Compartir parámetros. | Equivarianza. |



Traslación +
Convolución

=

Convolución +
Traslación

Figura: CNN (superior) y NN (inferior)

Red neuronal de convolución

Conceptos básicos

Input (X) de tamaño ($m \times m \times c$).

Kernel o filtro (K) de tamaño ($n \times n \times c \times f$).

c: número de canales.

f: número de filtros.

Red neuronal de convolución

Conceptos básicos

Input (X) de tamaño $(m \times m \times c)$.

Kernel o filtro (K) de tamaño $(n \times n \times c \times f)$.

c: número de canales.

f: número de filtros.

$$Z_{i,j,k} = \sum_i^m \sum_j^n \sum_o^l X_{i+m-1,j+n-1,o} K_{i,j,o,k} \quad (7)$$

Red neuronal de convolución

Conceptos básicos

Input (X) de tamaño $(m \times m \times c)$.

Kernel o filtro (K) de tamaño $(n \times n \times c \times f)$.

c: número de canales.

f: número de filtros.

$$Z_{i,j,k} = \sum_i^m \sum_j^n \sum_o^l X_{i+m-1,j+n-1,o} K_{i,j,o,k} \quad (7)$$

Por ejemplo :

3	1	1	3
1	2	0	4
3	2	4	0
1	1	3	1

*

1	0
1	0

=

4

Red neuronal de convolución

Conceptos básicos

- Capa convolucional.
- Capa *pooling*.
- Capa totalmente conectada

Red neuronal de convolución

Capa convolucional

Definición

El *stride* (s) indica cómo el filtro convoluciona a lo largo del input.

Red neuronal de convolución

Capa convolucional

Definición

El *stride* (s) indica cómo el filtro convoluciona a lo largo del input.

$$\begin{bmatrix} 3 & 1 & 1 & 3 \\ 1 & 2 & 0 & 4 \\ 3 & 2 & 4 & 0 \\ 1 & 1 & 3 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 3 \end{bmatrix} \quad s = 1.$$

Red neuronal de convolución

Capa convolucional

Definición

El *stride* (s) indica cómo el filtro convoluciona a lo largo del input.

$$\begin{bmatrix} 3 & 1 & 1 & 3 \\ 1 & 2 & 0 & 4 \\ 3 & 2 & 4 & 0 \\ 1 & 1 & 3 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 3 \end{bmatrix} \quad s = 1.$$

$$\begin{bmatrix} 3 & 1 & 1 & 3 \\ 1 & 2 & 0 & 4 \\ 3 & 2 & 4 & 0 \\ 1 & 1 & 3 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 1 \end{bmatrix} \quad s = 2.$$

Red neuronal de convolución

Capa convolucional

Definición

El *padding* (p) indica la cantidad de celdas con ceros que se añaden en el alto y largo del input.

Red neuronal de convolución

Capa convolucional

Definición

El *padding* (p) indica la cantidad de celdas con ceros que se añaden en el alto y largo del input.

- *Valid padding*: no añade nada.
- *Same padding*: lo justo para que el input y el output tengan el mismo tamaño.

Red neuronal de convolución

Capa convolucional

Output de tamaño:

$$m^L = \frac{m^{L-1} - n^L + 2 \cdot p^L}{s^L} + 1 \quad (8)$$

Output: $(m^L \times m^L \times f^L)$.

Red neuronal de convolución

Capa convolucional

Output de tamaño:

$$m^L = \frac{m^{L-1} - n^L + 2 \cdot p^L}{s^L} + 1 \quad (8)$$

Output: $(m^L \times m^L \times f^L)$.

Operación (7) + Activación (2)

Red neuronal de convolución

Capa convolucional

Output de tamaño:

$$m^L = \frac{m^{L-1} - n^L + 2 \cdot p^L}{s^L} + 1 \quad (8)$$

Output: $(m^L \times m^L \times f^L)$.

Operación (7) + Activación (2)

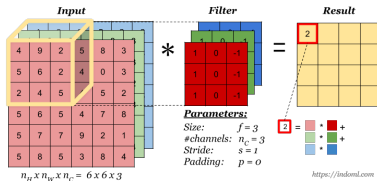


Figura: Convolución con 3 canales

Red neuronal de convolución

Capa de *pooling*

- *Maxpool.*
- *Averagepool.*

Red neuronal de convolución

Capa de *pooling*

- *Maxpool.*
- *Averagepool.*

Observación

- Terera dimensión constante $c^{L-1} = c^L$.
- Primeras dos dimensiones como en (12).

Red neuronal de convolución

Capa de *pooling*

- *Maxpool*.
- *Averagepool*.

Observación

- Terera dimensión constante $c^{L-1} = c^L$.
- Primeras dos dimensiones como en (12).

Por ejemplo, $\max(2,2)$

$$\text{Maxpool} \begin{bmatrix} 1 & 1 & 2 & 5 \\ 2 & 2 & 2 & 3 \\ 1 & 7 & 0 & 1 \\ 1 & 2 & 4 & 1 \end{bmatrix} = \begin{bmatrix} \max(1,1,2,2) = 2 & 2 & 5 \\ \max(2,2,1,7) = 7 & 7 & 3 \\ \max(1,7,1,2) = 7 & 7 & 4 \end{bmatrix}$$

Red neuronal de convolución

Capa totalmente conectada

Definición

Se disponen los valores en un vector unidimensional y se tiene una red neuronal *feed-forward*.

Red neuronal de convolución

Capa totalmente conectada

Definición

Se disponen los valores en un vector unidimensional y se tiene una red neuronal *feed-forward*.

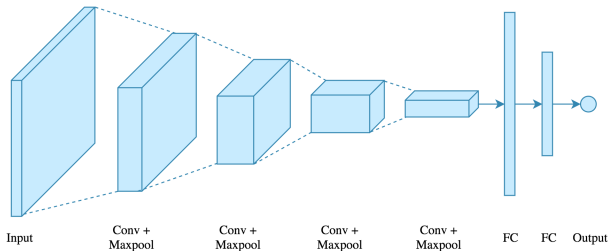


Figura: Arquitectura completa de red neuronal convolucional

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - **Matriz de confusión**
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Matriz de confusión

Problema de **clasificación binaria**



Umbral

Matriz de confusión

Problema de **clasificación binaria** → Umbral

	P	V
P	VP	FP
N	FN	VN

Figura: Matriz de confusión

Matriz de confusión

Problema de **clasificación binaria** → Umbral

	P	V
P	VP	FP
N	FN	VN

Figura: Matriz de confusión

$$VPR = \frac{VP}{P} \quad P: \text{total de positivos} \quad (9)$$
$$SPC = \frac{VN}{N} \quad N: \text{total de negativos}$$

Matriz de confusión

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (10)$$

Matriz de confusión

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (10)$$

Problemas

Matriz de confusión

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (10)$$

Problemas

- Díficil comparar modelos.

Matriz de confusión

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (10)$$

Problemas

- Díficil comparar modelos.
- Objetivo distinto a mejorar precisión.

Matriz de confusión

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (10)$$

Problemas

- Díficil comparar modelos.
- Objetivo distinto a mejorar precisión.
- Umbral óptimo cercano al valor nulo.

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - **Curva ROC**
- 5 Caso de estudio
- 6 Conclusiones

Curva ROC

Eje x: 1-SPC.

Eje y: VPR.

Curva ROC

Eje x: 1-SPC.

Eje y: VPR.

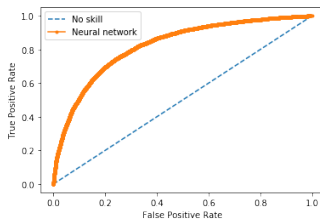


Figura: Curva ROC

Curva ROC

Eje x: 1-SPC.

Eje y: VPR.

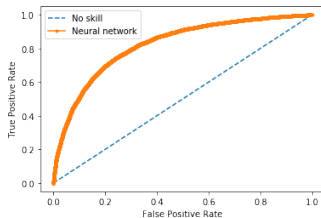


Figura: Curva ROC

Área bajo la curva (AUC)

Curva ROC

Eje x: 1-SPC.

Eje y: VPR.

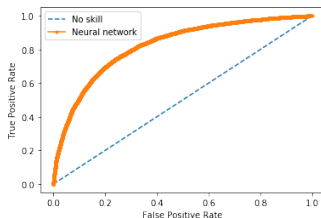


Figura: Curva ROC

→ 1 clasificación perfecta.

Área bajo la curva (AUC)

→ 0.5 clasificación aleatoria.

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio**
- 6 Conclusiones

Caso de estudio

Dataset

Entrenamiento: 220024 imágenes. Test: 57457.

Caso de estudio

Dataset

Entrenamiento: 220024 imágenes. Test: 57457.

Tamaño: $(96 \times 96 \times 3)$ píxeles, central $(32 \times 32 \times 3)$.

Caso de estudio

Dataset

Entrenamiento: 220024 imágenes. Test: 57457.

Tamaño: $(96 \times 96 \times 3)$ píxeles, central $(32 \times 32 \times 3)$.

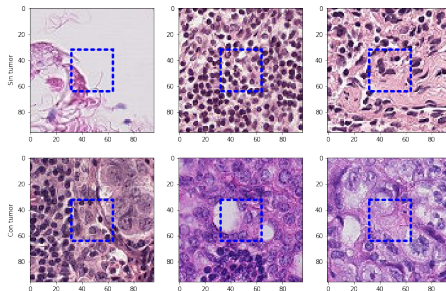


Figura: Exploraciones histopatológicas de secciones de ganglios linfáticos

Caso de estudio

Dataset

40 % de “ 1 ” y el 60 % de “ 0 ”.

Caso de estudio

Dataset

40 % de " 1 " y el 60 % de " 0 ".

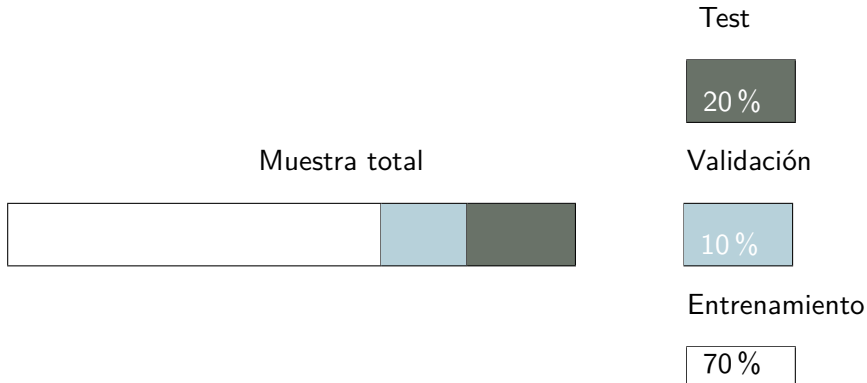


Figura: Conjunto de entrenamiento, validación y test

Caso de estudio

Preprocesamiento

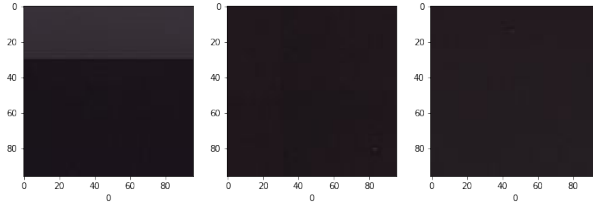


Figura: Posibles *outliers*

Caso de estudio

Preprocesamiento

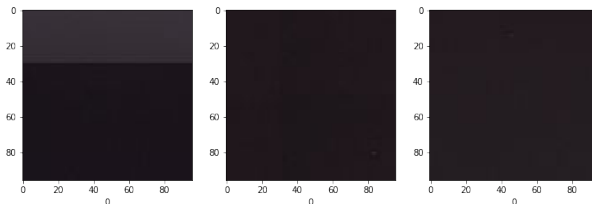


Figura: Posibles *outliers*

Se eliminan el 0.001 % del dataset.

Caso de estudio

Resultados

- NN1.
- NN2.
- CNN1.
- CNN2.

Caso de estudio

Resultados

Estandarización → Rango de variables.

Caso de estudio

Resultados

Estandarización	→	Rango de variables.
Inicialización	→	Xavier Uniform.

Caso de estudio

Resultados

Estandarización	→	Rango de variables.
Inicialización	→	Xavier Uniform.
Error	→	Entropía cruzada binaria.

Caso de estudio

Resultados

Estandarización → Rango de variables.

Inicialización → Xavier Uniform.

Error → Entropía cruzada binaria.

Función de activación salida → Sigmoid.

Caso de estudio

Resultados

NN1

Neuronas en capa oculta → 350.

Caso de estudio

Resultados

NN1

Neuronas en capa oculta →

350.

Función activación →

Leaky Relu.

Caso de estudio

Resultados

NN1

Neuronas en capa oculta	→	350.
Función activación	→	Leaky Relu.
Estandarización	→	Tras primera capa.

Caso de estudio

Resultados

NN1

Neuronas en capa oculta	→	350.
Función activación	→	Leaky Relu.
Estandarización	→	Tras primera capa.
Método de optimización	→	Mini-Batch Descenso del gradiente. $\alpha = 0.0001$, momento = 0.8.

Caso de estudio

Resultados

NN1

Neuronas en capa oculta	→	350.
Función activación	→	Leaky Relu.
Estandarización	→	Tras primera capa.
Método de optimización	→	Mini-Batch Descenso del gradiente. $\alpha = 0.0001$, momento = 0.8.
Tamaño del <i>batch</i>	→	174.

Caso de estudio

Resultados

NN1

AUC: 0.821

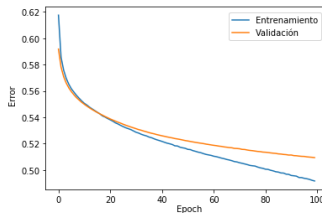


Figura: Aprendizaje NN1

Caso de estudio

Resultados

NN2

+ capa de 40 neuronas (con Leaky Relu).

Caso de estudio

Resultados

NN2

+ capa de 40 neuronas (con Leaky Relu).

AUC: 0.823.

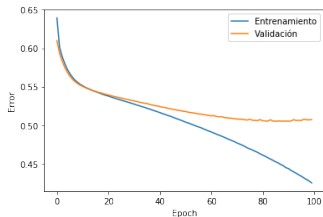


Figura: Aprendizaje NN2.

Caso de estudio

Resultados

CNN1

Tamaño del *batch* → 29.

Caso de estudio

Resultados

CNN1

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 3.

Caso de estudio

Resultados

CNN1

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 3.

Método de optimización → del gradiente.

Mini-Batch Descenso

$\alpha = 0.0001$, momento = 0.8.

Caso de estudio

Resultados

CNN1

Nº de filtros 15	Tamaño del filtro (7,7,3)	Activación Relu	<i>Pool layer</i> Max(4,4), s = 2
---------------------	------------------------------	--------------------	--------------------------------------

Primera capa.

Caso de estudio

Resultados

CNN1

Nº de filtros 15	Tamaño del filtro (7,7,3)	Activación Relu	<i>Pool layer</i> Max(4,4), s = 2
---------------------	------------------------------	--------------------	--------------------------------------

Primera capa.

Nº de filtros 20	Tamaño del filtro (5,5,15)	Activación Relu	<i>Pool layer</i> Max(4,4), s = 2
---------------------	-------------------------------	--------------------	--------------------------------------

Segunda capa.

Caso de estudio

Resultados

CNN1

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
30	(3,3,20)	Relu	Max(2,2), $s = 2$

Tercera capa.

Nº neuronas	Activación
1920	Sigmoid

Capa totalmente conectada.

Caso de estudio

Resultados

CNN1

AUC: 0.930

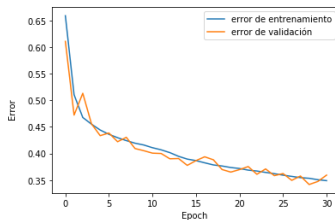


Figura: Aprendizaje CNN1

Caso de estudio

Resultados

CNN2

Tamaño del *batch* → 29.

Caso de estudio

Resultados

CNN2

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 4.

Caso de estudio

Resultados

CNN2

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 4.

Adam

Método de optimización → $\alpha = 0.001$, $\beta_1 = 0.9$,
 $\beta_2 = 0.999$ y $\epsilon = 10^{-8}$

Caso de estudio

Resultados

CNN2

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 4.

Adam

Método de optimización → $\alpha = 0.001$, $\beta_1 = 0.9$,
 $\beta_2 = 0.999$ y $\epsilon = 10^{-8}$

Dropout → $p = 0.25$

Caso de estudio

Resultados

CNN2

Tamaño del *batch* → 29.

Capas convolucionales + *pool* → 4.

Adam

Método de optimización → $\alpha = 0.001$, $\beta_1 = 0.9$,
 $\beta_2 = 0.999$ y $\epsilon = 10^{-8}$

Dropout → $p = 0.25$

Pool → *max y avg*

Caso de estudio

Resultados

CNN2

Nº de filtros 15	Tamaño del filtro (6,6,3), $s = 2$	Activación Relu	<i>Pool layer</i> Avg(4,4)
---------------------	---------------------------------------	--------------------	-------------------------------

Primera capa.

Caso de estudio

Resultados

CNN2

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
15	(6,6,3), $s = 2$	Relu	Avg(4,4)

Primera capa.

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
20	(5,5,15), $s = 2$	Relu	Avg(4,4), $s = 2$

Segunda capa.

Caso de estudio

Resultados

CNN2

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
15	(6,6,3), $s = 2$	Relu	Avg(4,4)

Primera capa.

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
20	(5,5,15), $s = 2$	Relu	Avg(4,4), $s = 2$

Segunda capa.

Nº de filtros	Tamaño del filtro	Activación	<i>Pool layer</i>
30	(3,3,20)	Relu	Max(2,2)

Tercera capa

Caso de estudio

Resultados

CNN2

Nº de filtros 40	Tamaño del filtro (3,3,30)	Activación Relu	<i>Pool layer</i> Max(2,2)
---------------------	-------------------------------	--------------------	-------------------------------

Cuarta capa

Caso de estudio

Resultados

CNN2

Nº de filtros 40	Tamaño del filtro (3,3,30)	Activación Relu	<i>Pool layer</i> Max(2,2)
---------------------	-------------------------------	--------------------	-------------------------------

Cuarta capa

Nº neuronas 160	Activación Sigmoid
--------------------	-----------------------

Capa totalmente conectada

Caso de estudio

Resultados

CNN2

AUC: 0.948

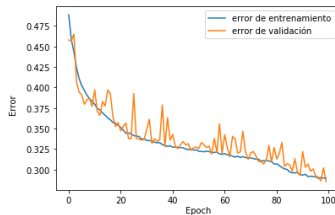


Figura: Aprendizaje CNN2

Caso de estudio

Resultados

Discusión

- Número de parámetros.

Caso de estudio

Resultados

Discusión

- Número de parámetros.
- Curva de aprendizaje.

Caso de estudio

Resultados

Discusión

- Número de parámetros.
- Curva de aprendizaje.
- Inputs tomados.

Caso de estudio

Resultados

Discusión

- Número de parámetros.
- Curva de aprendizaje.
- Inputs tomados.

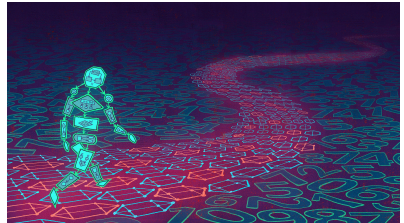
CNN2

AUC sobre test: 0.9224

Índice

- 1 Introducción y objetivos
- 2 Estado del arte
- 3 Conceptos previos
 - Conceptos de imágenes
 - Red neuronal *feed-forward*
 - Red neuronal de convolución
- 4 Métricas de validación
 - Matriz de confusión
 - Curva ROC
- 5 Caso de estudio
- 6 Conclusiones

Conclusiones



¡Muchas gracias!

Bibliografía I



Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele.

In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 876–885, 2017.



Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.

Bibliografía II



Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

MIT Press, 2016.

<http://www.deeplearningbook.org>.



Xavier Glorot and Yoshua Bengio.

In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.