

DETECCIÓN DE CÁNCER MEDIANTE REDES NEURONALES CONVOLUCIONALES

Por Otto F. Wagner

www.ottofwagner.com

ofwagner@ottofwagner.com

Proyecto Fin de Máster dirigido por:

Dr. Fernando Corbacho

Dr. Manuel Sánchez-Montañés

MÁSTER EN DEEP LEARNING 2018/2019



Índice

1. Introducción y Contexto
2. Modelo y Resultados
3. Conclusiones

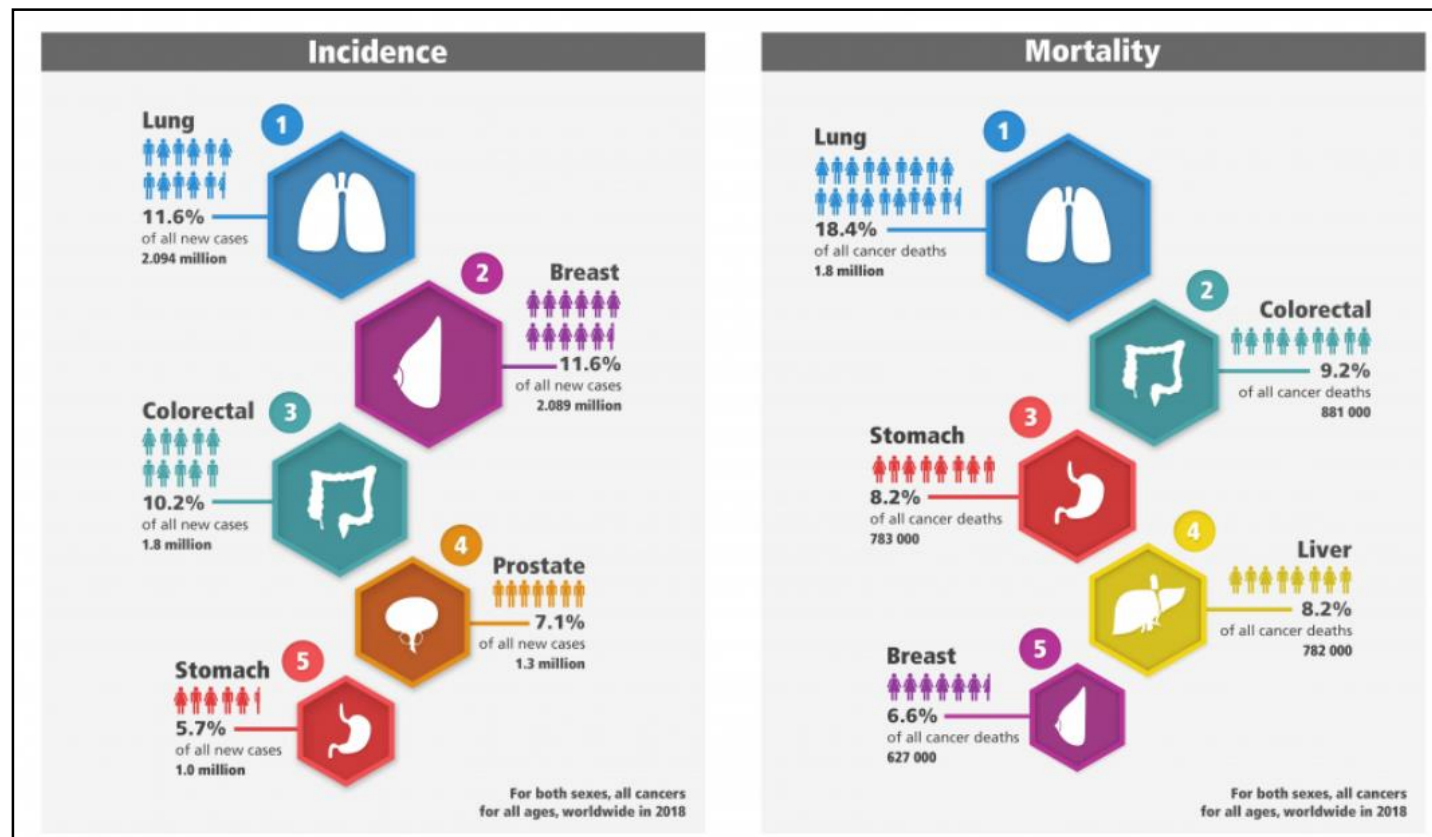


Introducción y Contexto



El cáncer es una de las mayores causas de mortalidad y morbilidad

- 18,1 millones de casos que causan 9,6 millones de muertes al año.



La IA una ayuda eficaz

- Una primera exploración suele estar acompañada por radiografías.
- Esta tarea consume mucho tiempo, ya que, requiere que radiólogos expertos estudien las imágenes.
- Dos problemas que pueden provocar un error de diagnóstico:
 - Fatiga del profesional.
 - Falta de experiencia diagnóstica en áreas del mundo donde los radiólogos no están disponibles (Rajpurkar *et al.*, 2018).
- La Inteligencia Artificial puede ayudar al diagnóstico precoz de estas enfermedades.



Un caso concreto, la radiografía de tórax

- Para enfocar el uso de esta tecnología nos centraremos en las radiografías de tórax.
- La base de datos del estudio está formada por una muestra aleatoria de 5,606 imágenes de la base de datos del *National Institute of Health: Chest X-ray Dataset Sample (2017)*, y su correspondiente etiquetado.
- Nos centraremos en la localización de masas y nódulos. 553 (9.86%) radiografías presentan alguno de estos hallazgos.



Un caso concreto, la radiografía de tórax

Edad Media	46.7 años
Edad Mínima	1 día
Edad Máxima	94 años
Mujeres	44%
Hombres	56%
Imágenes Anteroposterior	39%
Imágenes Posteroanterior	61%
Nº Radiografías media por paciente	2.7
Tiempo medio entre radiografías	1.7 años

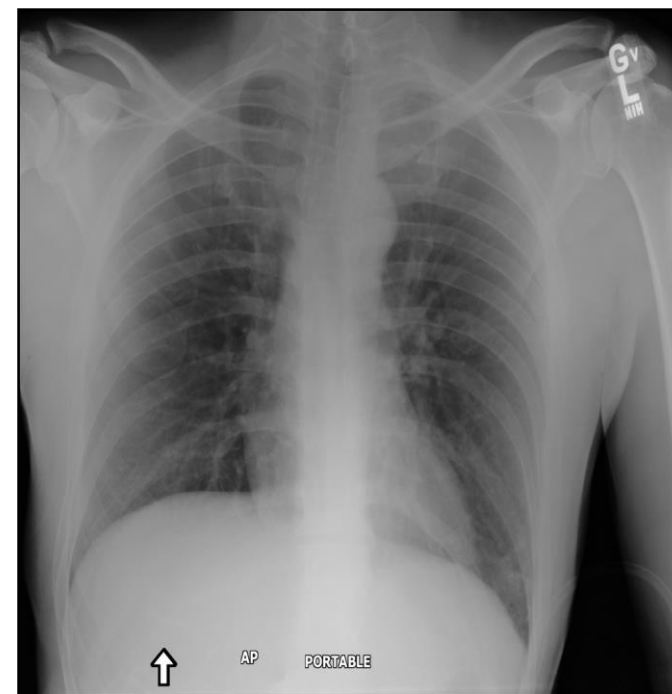


Imagen de Radiografía 1024 x 1024

Objetivos

- Probar varias arquitecturas de redes neuronales convolucionales usando *transfer learning*.
- Intentar encontrar los mejores hiper-parámetros que configuren las redes.
- Realizar un *benchmarking* de los modelos de redes neuronales frente a otros modelos de Machine Learning evaluando el poder discriminante mediante la curva ROC y el área bajo ésta (AUC).
- Seleccionar un modelo de Deep Learning que constituya nuestra prueba de concepto.



Modelo y Resultados



Fases del modelo

1. Conversión de las radiografías en tensores para su tratamiento a través de la red.
2. Utilización de las capas de una red convolucional pre-entrenada (*transfer learning*) para la extracción de las características principales de las imágenes.
3. Entrenamiento de las últimas capas del modelo con una red neuronal artificial profunda (*Multilayer Perceptron*) versus modelos *benchmarking*. Tratamiento del balanceo.
4. De manera complementaria se ha sustituido la red neuronal artificial profunda por tres modelos: regresión logística, *random forest* y *gradient boosting classifier*.



Las redes convolucionales funcionan adecuadamente para este problema

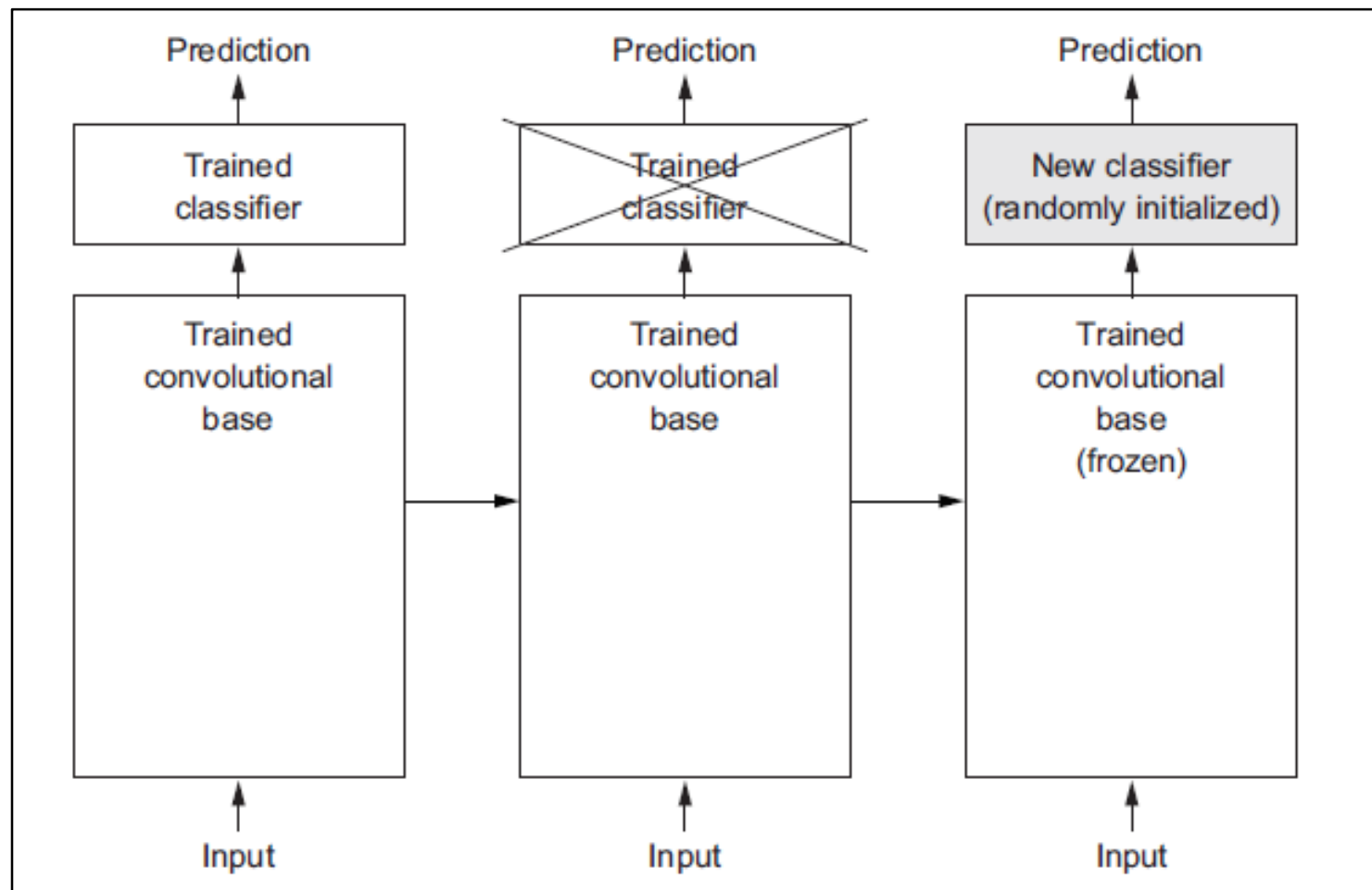
- Nos ayudarán a extraer características generales de las imágenes.
- Usaremos *transfer learning* en las capas convolucionales y de *pooling*.
- Entrenaremos las capas densas.

	AlexNet	GoogLeNet	VGGNet-16	ResNet-50
Atelectasia	0,65	0,63	0,63	0,71
Cardiomegalia	0,69	0,71	0,71	0,81
Derrame	0,66	0,69	0,65	0,74
Infiltración	0,60	0,61	0,59	0,61
Masa	0,56	0,54	0,51	0,56
Nódulo	0,65	0,56	0,66	0,72
Neumonía	0,55	0,60	0,51	0,63
Neumotórax	0,74	0,78	0,75	0,79

Comparativa de AUC de arquitecturas preentrenadas. (Wang *et al.*, 2017)



El *transfer learning* es nuestro gran aliado

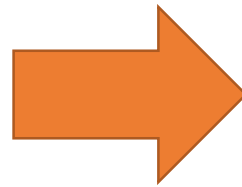


Transfer Learning para extracción de características (Chollet, 2017)



La arquitectura VGGNet es la que mejor funciona

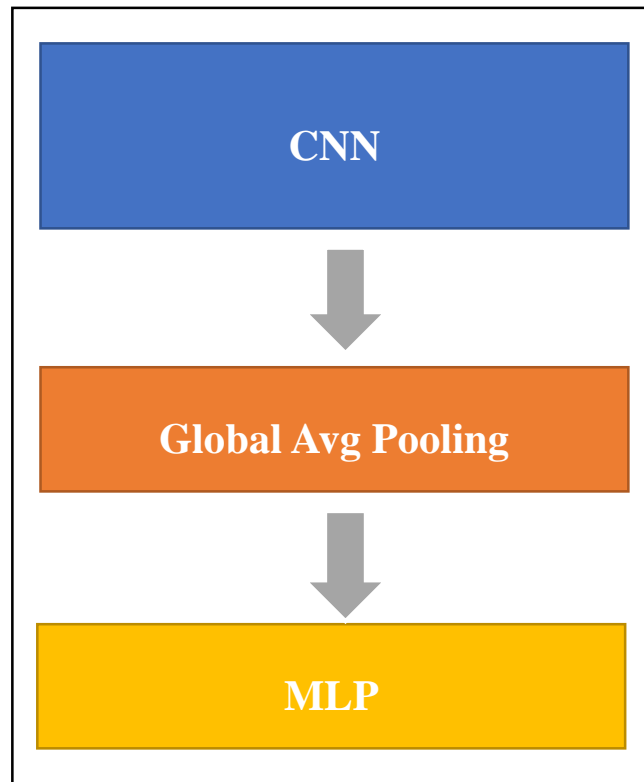
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144



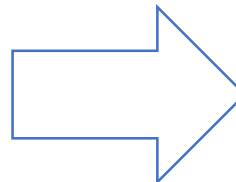
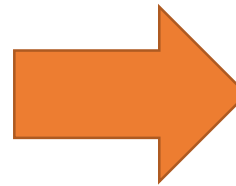
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 256, 256, 3)	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
Total params: 14,714,688		
Trainable params: 14,714,688		
Non-trainable params: 0		



Utilizamos 3 capas densas para entrenar



CCN
+
Random Forest / Regresión Logística



Layer (type)	Output Shape	Param #
=====		
dropout_17 (Dropout)	(None, 512)	0
batch_normalization_25 (Batch Normalization)	(None, 512)	2048
dense_25 (Dense)	(None, 30)	15390
dropout_18 (Dropout)	(None, 30)	0
batch_normalization_26 (Batch Normalization)	(None, 30)	120
dense_26 (Dense)	(None, 20)	620
batch_normalization_27 (Batch Normalization)	(None, 20)	80
dense_27 (Dense)	(None, 1)	21
=====		
Total params: 18,279		
Trainable params: 17,155		
Non-trainable params: 1,124		
None		

Benchmarking

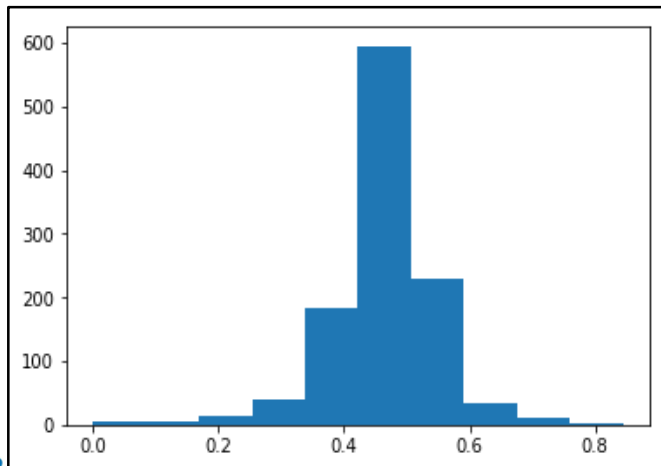
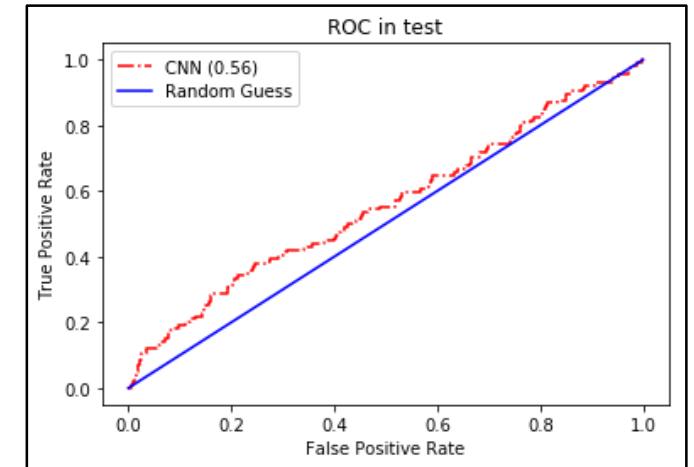
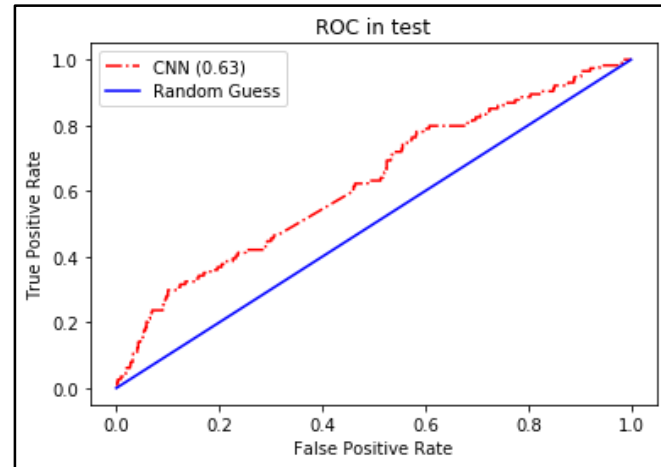
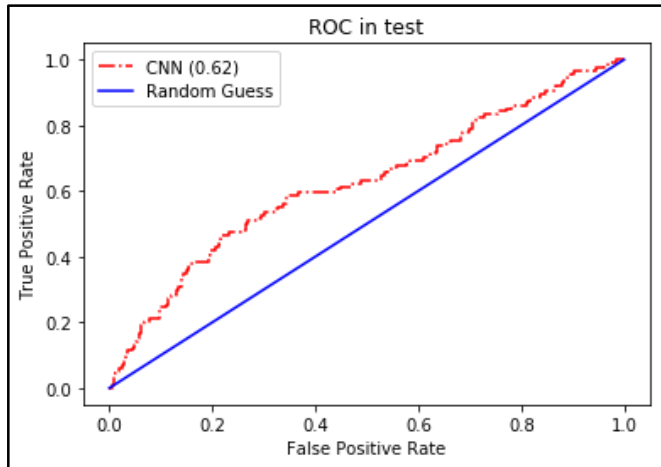


La arquitectura VGGNet junto al MLP da los mejores resultados

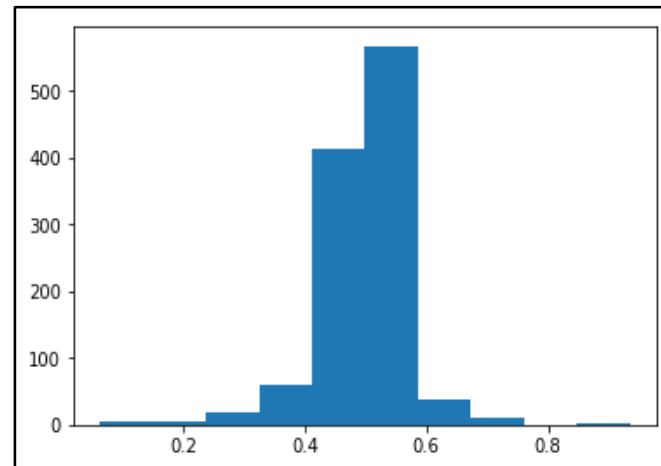
		VGGNet-16		VGGNet-19		ResNet-50	
		0	1	0	1	0	1
Regresión Logística	Score Avg.	0.74		0.74		0.7	
	Precision	0.88	0.1	0	0.1	0.9	0
	Recall	0.02	0.97	0	1	1	0
	F1-Score	0.04	0.18	0	0.18	0.95	0
	AUC	0.59		0.55		0.51	
Random Forest	Score Avg.	0.9		0.9		0.9	
	Precision	0.9	0	0.9	0	0.9	0
	Recall	1	0	1	0	1	0
	F1-Score	0.95	0	0.95	0	0.95	0
	AUC	0.54		0.51		0.52	
Red Neuronal	Accuracy Avg.	0.64		0.49		0.44	
	Precision	0.93	0.16	0.92	0.12	0.91	0.11
	Recall	0.65	0.59	0.48	0.66	0.43	0.61
	F1-Score	0.76	0.25	0.63	0.21	0.58	0.18
	AUC	0.62		0.63		0.56	



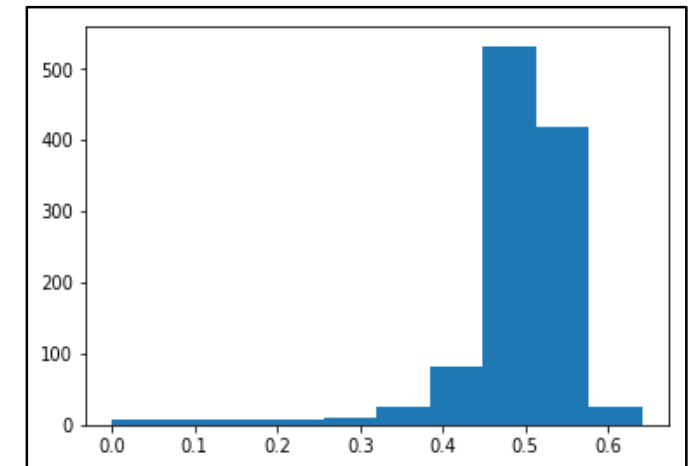
ROC: ninguno de los modelos discriminan lo suficientemente bien



VGGNet-16



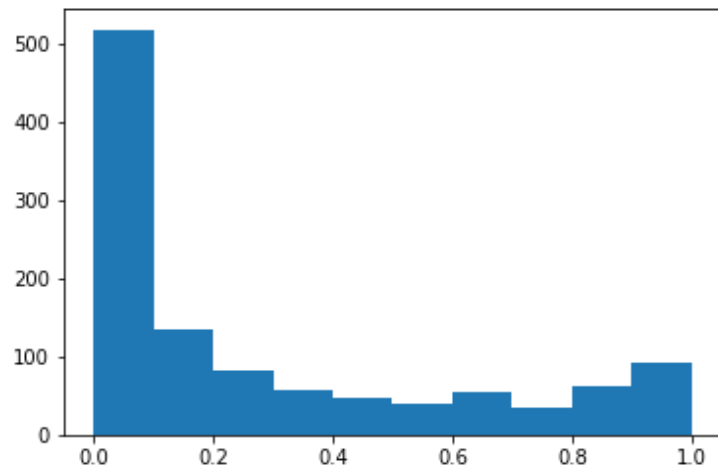
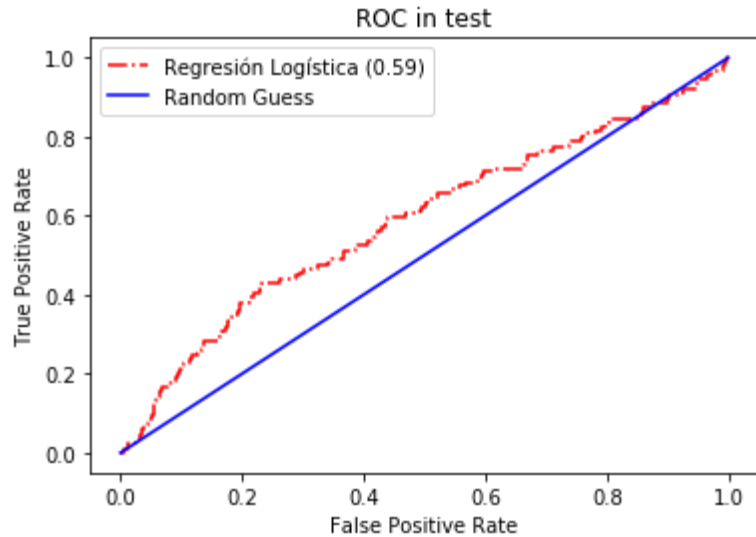
VGGNet-19



ResNet-50



CNN + Regresión Logística



		PARÁMETROS POR DEFECTO		GRID SEARCH	
		0	1	0	1
Regresión Logística	Score Avg.	0.74		0.73	
	Precision	0.88	0.1	0.91	0.11
	Recall	0.02	0.97	0.41	0.65
	F1-Score	0.04	0.18	0.57	0.19
	AUC	0.59		0.59	
MLP	Accuracy Avg.	0.64			
	Precision	0.93	0.16		
	Recall	0.65	0.59		
	F1-Score	0.76	0.25		
	AUC	0.62			



Comentarios a los resultados

- Los resultados aún se encuentran muy alejados de lo deseado
- Esto se debe a:
 - Muestra demasiado pequeña para poder entrenar con cierta fiabilidad.
 - Datos no balanceados.
 - Probables problemas de etiquetado de las radiografías.
 - Equipo poco potente, ausencia de GPUs, no pudiendo realizar un entrenamiento completo.
- La capa de MLP da los mejores resultados incluso cuando a los otros modelos de ML se les realiza un *grid search*.
- Nos decantamos por la VGGNet-16 al ser más sencilla que la VGGNet-19 y tener mejores resultados $AUC = 0.62$
- ResNet-50 discrimina peor que las arquitecturas VGGNet: $AUC = 0.56$



Conclusiones



Conclusiones

- Es una prueba de concepto de cómo se pueden predecir hallazgos patológicos a través del análisis de imágenes radiológicas.
- Este trabajo muestra como trabajar con pocos datos no balanceados.
- Los resultados, lejos de ser buenos, demuestran como las redes neuronales artificiales se pueden combinar con otros modelos (modelos híbridos).
- Los modelos que mejor han funcionado son las redes puras (CNN+MLP).



Mejoras al modelo y posibles líneas de investigación

- *Fine tuning* de todas las capas o de algunas de la VGGNet-16.
- Entrenamiento completo de la VGGNet-16.
- Trabajar con formatos de mayor calidad en la imagen: *dicom* o *nii.gz*.
- Utilización de otras bases de datos de radiografías.
- Clasificación multi-etiqueta de los 14 hallazgos del NIH.
- Construcción de mapas de calor.
- Prueba de otras arquitecturas más complejas.
- Creación de una arquitectura de red convolucional ad-hoc.
- Combinación de CNN con RNN, teniendo en cuenta el histórico de cada paciente.
- Uso de algoritmos de detección de objetos en tiempo real.

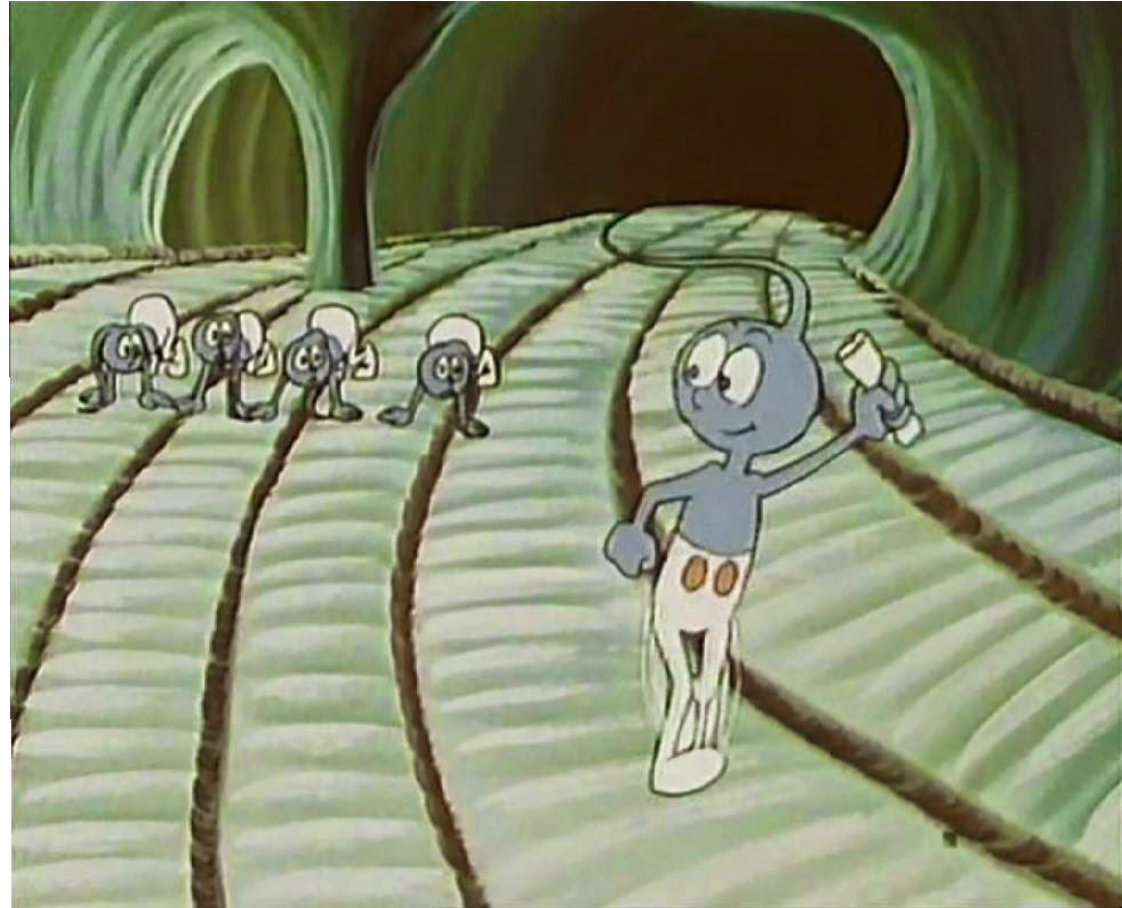


GRACIAS!!!

Otto F. Wagner

ofwagner@ottofwagner.com

www.ottofwagner.com



Anexo: etiquetado

	MetaMap			MetaMap + Dnorm		
Patología	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Atelectasia	0.95	0.95	0.95	0.99	0.85	0.91
Cardiomegalia	0.99	0.83	0.90	1.00	0.79	0.88
Derrame	0.74	0.90	0.81	0.93	0.82	0.87
Infiltración	0.25	0.98	0.39	0.74	0.87	0.80
Masa	0.59	0.67	0.62	0.75	0.40	0.52
Nódulo	0.95	0.65	0.77	0.96	0.62	0.75
Normal	0.93	0.90	0.91	0.87	0.99	0.93
Neumonía	0.58	0.93	0.71	0.66	0.93	0.77
Neumotórax	0.32	0.82	0.46	0.90	0.82	0.86
Total	0.84	0.88	0.86	0.90	0.91	0.90



Anexo: formulas *performance*

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

$$Precision = \frac{TP}{TP + FP}$$

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

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall}$$

