

Estudio de Redes Neuronales Convolucionales

Caso de estudio: Predicción de líneas espectrales

Claudio Araya E.

^aEscuela de Ingeniería Civil Informática, Facultad de Ingeniería, Universidad de Valparaíso, Valparaíso, Chile

Abstract

Este trabajo presenta un estudio realizado para analizar el comportamiento de una red neuronal convolucional cuando se cambian algunos de los parámetros y valores. Se utilizan 5425 líneas espectrales sintéticas con rango de longitud de onda de [4460, 4480], y 201 puntos cada una, como datos de entrada. Adicionalmente, se utiliza una línea espectral observada para analizar las predicciones del modelo. Se utilizó Jupyter Notebook con la librería Keras, principalmente, para la implementación de un modelo de múltiples capas conv1D y capas Dense. Se modificaron los parámetros del modelo en diferentes configuraciones para comparar los resultados. Se pudo comprobar que los resultados del modelo cambian sustancialmente al variar la configuración de las capas y los valores de los hiper-parámetros. También se encontró que, dadas las características de la línea espectral observada, hay mucha variación en las predicciones que realiza el modelo aun cuando el error, pérdida y precisión se vean adecuados.

Keywords: Redes Neuronales Convolucionales, CNN, Líneas Espectrales

1. Introducción

Gracias a la llegada de la mecánica cuántica se pudo comprender con mayor profundidad la existencia de las líneas espectrales. Las líneas espectrales están formadas por "puntos" que representan la cantidad de energía recibida (en una unidad de superficie, en una unidad de tiempo, en intervalo de una frecuencia de onda) en una frecuencia de onda, conocido como flujo.[1]

La cantidad de puntos que definen a una línea espectral (en este caso se usan 201 puntos en un pequeño rango de la frecuencia de onda) convierten en una tarea lenta y costosa la comparación o clasificación manual, por lo que el Machine Learning y - particularmente - las redes neuronales se convierten en un efectivo método para esta tarea.

Las redes neuronales convolucionales han sido desarrolladas como respuesta al surgimiento de problemas mucho mas complejos, demostrando ser mucho mas eficientes que las redes neuronales clásicas en el reconocimiento de imágenes [2]. Ello debido al surgimiento de las capas convolucionales, las cuales a través del uso de filtros realizan operaciones convolucionales que permiten generar una representación reducida de la misma imagen, transformándolas en mapas de características, que luego son traspasadas a la siguiente capa, hasta que finalmente son procesadas por las capas totalmente conectadas (Fully-connected) [3]

1.1. Problema

El disponer de una gran cantidad de información no garantiza que esta se pueda utilizar, ya que precisamente, el volumen

de ésta hace muy costoso, en términos de recursos, su procesamiento. Este es el caso de los datos astronómicos, cuyo volumen ha ido incrementándose con el paso del tiempo.

Por tanto, para procesar esta información se ha convertido, casi en imprescindible, el uso de tecnologías informáticas que apoyen el manejo de datos tal como lo hacen las herramientas de BigData y Machine Learning.

Existen herramientas que van desde librerías hasta frameworks, pasando por herramientas que hacen uso de las anteriores. Es el caso de Keras, que es un API para deep learning sobre TensorFlow, desarrollado sobre Python y que permite su uso sobre este mismo lenguaje.[4]

Una red neuronal convolucional se puede modelar fácilmente en un Jupyter Notebook. Utilizando Keras, se define mediante la creación del modelo. Este modelo - en particular - será de tipo secuencial, es decir, los datos pasan de una capa a la siguiente en orden. La efectividad del modelo sobre los datos utilizados depende de la forma en que este sea configurado: la cantidad y tipo de capas, los parámetros para cada capa y los valores que se definan para los hiper-parámetros.

Cuántas capas convolucionales y cuántas capas totalmente conectadas se usarán, cuales son los valores adecuados para cada capa y que hiper-parámetro utilizar es lo que se debe descubrir para encontrar un modelo adecuado a los datos utilizados.

1.2. Métodos

Machine Learning es un campo de investigación, parte de la Inteligencia Artificial, que permite descubrir información, conocimiento, desde los datos de manera automatizada. Las aplicaciones de Machine Learning han ido en incremento, y a su vez ha ido incrementándose la demanda. [5]

Email address: claudio.arayae@postgrado.uv.cl (Claudio Araya E.)

Dentro de Machine Learning están las redes neuronales - también conocidas como redes neuronales artificiales ANN (por sus siglas en inglés) - son un subconjunto de estas y son parte importante de los algoritmos de Deep Learning. Están compuestas por múltiples capas de nodos, conectadas entre si (capa a capa). Se componen por una capa de entrada de datos, varias capas ocultas y una capa de salida.[6] La figura 1 representa una red neuronal artificial.

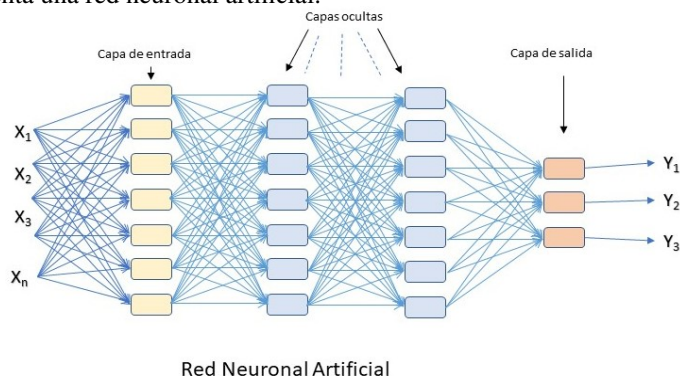


Figura 1. Red Neuronal Artificial

En el siguiente escalón se encuentran las redes neuronales convolucionales. Estas serán las utilizadas para llevar a cabo el desarrollo de esta investigación.

1.3. Contribuciones

La inteligencia artificial aparece a mediados del siglo pasado en la Conferencia de Dartmouth, organizada por Marvin Minsky, John McCarthy y Claude Shannon. La lógica teórica presentada se considera como el primer programa de inteligencia artificial. Pero ciertamente, en el resto del siglo, los avances obtenidos en esta materia fueron muy discretos. [7]

Es a principios del siglo 21, luego del gran suceso de 1997 en que la máquina Deep Blue, desarrollada por IBM, logró derrotar en ajedrez al gran maestro Kasparov. Este fué un acontecimiento de gran notoriedad mediática ya que representaba la victoria de la inteligencia artificial sobre la humana.

Esto no quiere decir que entre el año 1956, Conferencia de Dartmouth, y finales del siglo pasado no hayan existido aportes a la materia. Se pueden mencionar, entre otros, en 1974 se presenta el desarrollo, por Paul Werbos, del algoritmo de aprendizaje de propagación hacia atrás (back propagation). Stephen Grossberg desarrolla en 1977 la Teoría de Resonancia Adaptada, que incluye memoria a largo y corto plazo. En 1980 Kunihiko Fukushima desarrolla un modelo neuronal para el reconocimiento de patrones visuales. John Hopfield, publica el libro "Computación neuronal de decisiones en problemas de optimización" lo que provoca el renacimiento de las redes neuronales[8]

La primera red neuronal convolucional fue desarrollada en 1998 por Yann LeCun, inspirado en el trabajo de Fukushima sobre Neocognitron.

El año 2009 representa un gran salto en el desempeño de las redes neuronales. En dicho año Rajat, Madhavan y Andrew, presentan su trabajo relacionado al uso de las tarjetas graficas en el procesamiento de los modelos de machine learning en el en-

trenamiento, probando que se aceleraba el proceso en mas de 70 veces respecto al uso de las tradicionales CPU multi-núcleos.[9]

1.4. Objetivos

El objetivo principal de este estudio es: Encontrar los parámetros que mejor ajustan el modelo de una Red Neuronal Convolucional a los datos.

Objetivos secundarios:

- Encontrar el modelo que brinde una mejor "clasificación" de los datos observados.
- Evaluar el impacto, de la transformación aleatoria sobre la línea observada, en la "predicción" que hace el modelo.

1.5. Organización del documento

El documento se presenta de manera que la próxima sección, "2. Métodos", describe el método o la forma en que trabajan las redes neuronales convolucionales. Seguida a esta, la sección 3, se presenta el experimento en donde se describen los datos utilizados y el proceso realizado. Posteriormente, en la sección 4, se exponen los resultados obtenidos y finalmente, las conclusiones de este trabajo se presentan en la sección 5.

2. Métodos

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son similares a las redes neuronales artificiales pero, con la variación de incluir capas convolucionales. Por ello entran en la clasificación de Deep Neural Networks (DNN), es decir, tiene mas de una capa oculta.

Las capas convolucionales son aquellas en que los nodos realizan operaciones de convolución, donde una convolución es una operación de dos funciones para generar una tercera. Particularmente, asociado a las redes neuronales, el concepto de convolución se refiere a la aplicación de una matriz (Kernel) sobre otra matriz (usualmente una imagen) para su tratamiento como filtro. La forma que tiene el filtro dependerá de lo que se busca destacar. De forma simplificada se puede decir que el filtro o Kernel se aplica sobre una zona de la imagen para generar un nuevo valor en la posición del píxel. En la figura 2 se muestra un ejemplo aplicado a un píxel de una imagen, en donde el nuevo píxel será el promedio de la del producto de los elementos de cada matriz :

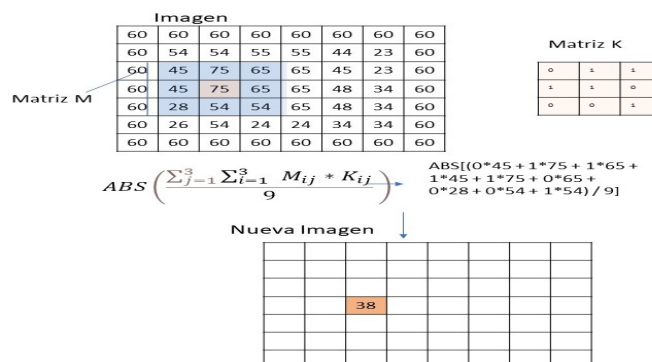


Figura 2. Ejemplo de convolución

Se puede apreciar que el valor del píxel nuevo es resultante de los valores circundantes al píxel en esa posición. Las operaciones y configuración del Kernel dependerán del objetivo buscado. (Se puede aplicar para desenfocar, buscar bordes, enfocar, resaltar colores, etc.)

El resultado de esta capa, es una nueva imagen con características resaltadas (o disminuidas, dependiendo de lo que se busque) y servirá de entrada a una siguiente capa, de esa manera se extraen características de la imagen que luego de pasar por todas las capas convolucionales son entregadas a las capas de ANN.

3. Experimentos

3.1. Datos

Para este estudio se cuenta con un conjunto de datos de 5425 líneas espectrales, originadas de manera artificial, que sirven como datos de entrada. Estas líneas espectrales representan flujos en longitudes de onda que están en el rango de 4460 y 4480 Å, presentes en cuerpos estelares de temperaturas entre 15000 y 39000 grados Kelvin, log g entre 2 y 5, y velocidad de rotación entre 100 y 400 Km/s. Cada línea espectral, de 201 puntos, se asocia a un conjunto de datos compuesto por temperatura, log g y velocidad de rotación.

Adicionalmente, se presenta una línea observada de 34243 puntos de flujo, asociados a longitud de onda que varía entre 4248 y 7308 Å. De esta línea observada se desconoce su temperatura, log g y velocidad de rotación, por lo que se utilizará el modelo para determinar estos valores.

3.2. Proceso

Se procederá a generar diversos modelos en los cuales se variará la cantidad de capas, el tipo de error, la cantidad de "épocas", las funciones de activación, etc.

Previamente, el conjunto de datos disponibles será separado en tres grupos: datos de entrenamiento, de validación y de test, en una proporción de 80, 10 y 10 por ciento respectivamente.

El modelo busca determinar los valores de temperatura, log g y velocidad de rotación de la estrella cuya línea espectral sea evaluada.

Para los efectos de evaluación o comparación de los modelos, se considerarán dos etapas: la primera etapa se realiza el aprendizaje en diversas iteraciones, durante las cuales se evalúa el rendimiento en función de un conjunto de datos de validación. En la segunda etapa se evalúa el rendimiento del modelo en base al conjunto de datos para "test".

Para la primera etapa se usarán como indicadores: la función de pérdida y el "accuracy". Para la segunda etapa, se usarán como indicadores el Error absoluto medio y el Error relativo en cada valor determinado (temperatura, log g y velocidad).

Posteriormente, se evaluará la línea observada en el modelo generado. Para ello, se debe ajustar la dimensión de los datos. Esta contiene 34243 puntos, en un intervalo de longitud de onda que varía entre 4248 y 7308 Å. Los datos de entrada utilizados se encuentran en un rango más acotado que está entre los 4460

y 4480 Å, por tanto fue necesario realizar un preproceso de selección de puntos que se encuentren en ese mismo rango de longitud de onda.

Finalmente, existen 224 puntos en el rango deseado y será necesario remover 23 puntos para dejar solo 201 e igualar a los datos conocidos por el modelo. Debido a esta remoción aleatoria, los resultados podrán variar entre una evaluación y otra, por lo que se realizan 1000 evaluaciones y se obtendrán los indicadores de: media, desviación estándar, mediana y moda en cada uno de los valores que arroja el modelo (temperatura, log g y moda)

4. Resultados

Se realizaron múltiples variaciones al modelo, de las cuales se presenta 9 en la tabla 1. Los resultados obtenidos indican variaciones cuando se modifican parámetros. Por supuesto, se encontraron varias configuraciones que no convergían y se estancaban en la pérdida, como el caso que se muestra a continuación (50 épocas y LearningRate= 0.01):

Layer (type)	Output Shape	Param #
Conv1D_1 (Conv1D)	(None, 192, 512)	5632
Conv1D_2 (Conv1D)	(None, 183, 512)	2621952
Conv1D_3 (Conv1D)	(None, 177, 256)	917760
Conv1D_4 (Conv1D)	(None, 171, 128)	229504
Conv1D_5 (Conv1D)	(None, 165, 64)	57408
Conv1D_6 (Conv1D)	(None, 159, 64)	28736
Conv1D_7 (Conv1D)	(None, 157, 32)	6176
Conv1D_8 (Conv1D)	(None, 155, 32)	3104
flatten (Flatten)	(None, 4960)	0
Dense_1 (Dense)	(None, 32)	158752
Dense_2 (Dense)	(None, 16)	528
Dense_3 (Dense)	(None, 8)	136
Dense_4 (Dense)	(None, 3)	27
Total params: 4,029,715		
Trainable params: 4,029,715		
Non-trainable params: 0		

4.1. Etapa 1 - Aprendizaje

En las tablas 2 y 3 se encuentra la lista de los modelos expuestos. Se puede apreciar que se varían algunos elementos entre un modelo y otro. Se pueden ver los resultados respecto a la pérdida y el "accuracy".

Se puede apreciar que el mejor resultado en cuanto a estos dos indicadores, se encuentra en el modelo 8, cuando la pérdida toma un valor de 0.0251 y el accuracy toma el valor 0.9325. El modelo 8 varía en la función de pérdida respecto al modelo 7. Sin embargo, también es notorio que los mejores resultados - que no varían sustancialmente a los del modelo 8 - se reflejan en los modelos con la misma configuración de capas (modelos 6, 7 y 8). Por lo que se puede afirmar que el modelo que mejor se comporta en la etapa de aprendizaje-validación es el modelo 8.

Por otro lado, es destacable que el modelo 9 tiene unos indicadores cercanos al modelo 8, aun cuando su configuración en cuanto a las capas es menor. Al compararlo con el modelo 3 - que tiene una capa más - se puede evidenciar que el modelo 9 tiene mejor valor en ambos índices, hay que tener en cuenta que este modelo tiene 75 épocas contra las 50 del modelo 3 y su Learning rate es 10 veces mayor que el definido en el modelo 3.

4.2. Etapa 2 - testing

En la Tabla 1 se presentan los índices de la etapa de test para cada configuración de modelo. Se destaca en esta tabla que el modelo 8 tiene el error absoluto medio mas bajo, lo mismo en cuanto al error relativo medio de la temperatura. Sin embargo, el mejor valor para error relativo medio de Log g y velocidad se alcanza en el modelo 7, aunque las variaciones entre ellas son bastante pequeñas.

Versión	Error Absoluto Medio	Error Relativo Medio Temperatura	Error Relativo Medio Log g	Error Relativo Medio Velocidad
1	1727,8	19.89% +/- 10.63%	14.58% +/- 12.15%	7.06% +/- 8.12%
2	1515,68	17.83% +/- 16.0%	13.76% +/- 12.88%	3.92% +/- 3.5%
3	626,47	7.53% +/- 6.74%	11.98% +/- 11.33%	3.93% +/- 3.9%
4	407,18	4.67% +/- 3.8%	10.48% +/- 10.13%	2.25% +/- 2.23%
5	365,84	4.22% +/- 3.74%	10.45% +/- 11.51%	2.05% +/- 2.38%
6	367,29	4.2% +/- 3.89%	10.32% +/- 10.35%	2.64% +/- 2.58%
7	276,16	3.17% +/- 2.88%	9.76% +/- 10.11%	1.75% +/- 1.71%
8	249,83	2.89% +/- 2.67%	11.1% +/- 9.36%	2.15% +/- 1.99%
9	317,61	3.74% +/- 3.34%	10.2% +/- 10.48%	2.83% +/- 2.84%

Tabla 1.- Etapa de testing

4.3. Etapa de predicción

4.3.1. Temperatura

En cuanto a la predicción realizada a la línea espectral observada, no se puede afirmar que la predicción es buena en alguno de los modelos ya que no se conoce cual es el valor real. Sin embargo, a juzgar por el comportamiento del modelo se puede analizar los valores obtenidos como desviación estándar lo que permite identificar la estabilidad del modelo frente a los resultados.

En la tabla 4 se encuentran los valores obtenidos para la variable entregada por el modelo como predicción de la temperatura. En esta se puede apreciar que para los modelos 3, 4 y 8, el valor entregado como media se encuentra por debajo del rango factible, es de recordar que el rango de temperaturas está entre 15000 y 39000 grados Kelvin. Por tanto, se podría pensar que la media obtenida por el modelo 5 también debería considerarse fuera del rango factible ay que su desviación estándar indica que muchos valores caen fuera de esta.

Considerando la desviación estándar como indicador de la estabilidad, se podría afirmar que la predicción de la temperatura es mejor realizada por el modelo 1.

Versión	Temperatura				
	media	Desv. Estandar	%	Mediana	Moda
1	21308,79	686,78	3,22	21233,89	20298,12
2	22312,27	2797,78	12,54	22356,78	18468,71
3	10952,38	5090,51	46,48	10284,49	12275,79
4	13452,1	11226,11	83,45	16189,49	9021,97
5	38361,88	22045,21	57,47	39714,48	51833,5
6	25197,33	1599,83	6,35	251454,08	26457,59
7	22483,24	6177,35	27,48	23377,29	26645,69
8	14958,38	12003,72	80,25	14974,02	5324,77
9	26052,47	1970,68	7,56	261814,23	25873,29

Tabla 4.- predicción de temperatura

4.3.2. Log g

La zona factible para esta variable es entre el 2 y 5, por lo que se puede apreciar en la Tabla 5, todos los modelos entregan valores en la zona, con la excepción del modelo 4 en el que la media se ubica por debajo del rango de factibilidad. Nuevamente, considerando que la dispersión de los datos generados nos indica lo errático de las predicciones, se puede afirmar que la predicción de Log g, es mejor en el modelo 1.

Versión	Log g				
	media	Desv. Estandar	%	Mediana	Moda
1	4,45	0,16	3,60	4,46	4,52
2	3,82	0,19	4,97	3,82	3,57
3	3,87	0,28	7,24	3,86	3,64
4	1,85	1,34	72,43	2,02	1,09
5	3,37	1,77	52,52	3,32	4,53
6	4,51	0,81	17,96	4,5	5,46
7	3,18	0,75	23,58	3,08	4,19
8	4,33	1,18	27,25	4,42	5,19
9	4,92	0,27	5,49	4,89	4,76

Tabla 5.- predicción de Log g

4.3.3. Velocidad

Los distintos modelos entregaron predicciones para la velocidad con valores medios que van desde 68 y hasta 370 KM/s. La tabla 6 resume los valores entregados por cada modelo para la variable velocidad de rotación.

Se puede advertir que los modelos 4, 7 y 8, entregan un valor medio que está fuera del rango de factibilidad. mientras que, en esta oportunidad, es el modelo 9 el que tiene una dispersión mas controlada, aunque el modelo 1 también presenta una desviación estándar que representa menos del 10% de la media.

Versión	Velocidad				
	media	Desv. Estandar	%	Mediana	Moda
1	189,79	15,59	8,21	186,39	177,15
2	194,16	17,85	9,19	192,52	210,04
3	255,79	45,38	17,74	262,51	245,04
4	88,73	40,09	45,18	83,79	112,29
5	370,43	98,17	26,50	387,5	379,18
6	138,45	49,09	35,46	134,03	161,79
7	68,66	61,12	89,02	72,93	29,13
8	74,56	91,18	122,29	82,08	0,13
9	161,56	12,25	7,58	158,92	149,83

Tabla 6.- predicción de velocidad

Versión	Pérdida	Epochs	Capa	Filtros	Kernels	Activacion	Salida	LerningRate	Loss	Accuracy
1	mse	50	Conv1d	32	7	relu		0,0001	0,3346	0,8198
			Conv1d	16	3	relu				
			Dense			selu	32			
			Dense			Linear	3			
2	mse	50	Conv1d	128	14	relu		0,0001	0,2389	0,8398
			Conv1d	64	7	relu				
			Conv1d	32	7	relu				
			Conv1d	16	3	relu				
			Dense			selu	32			
			Dense			selu	16			
			Dense			selu	8			
			Dense			Linear	3			
3	mse	50	Conv1d	128	14	swish		0,0001	0,1425	0,8766
			Conv1d	64	7	swish				
			Conv1d	32	7	swish				
			Conv1d	16	3	swish				
			Dense			selu	32			
			Dense			selu	16			
			Dense			selu	8			
			Dense			Linear	3			
4	mse	50	Conv1d	128	14	swish		0,001	0,0797	0,9116
			Conv1d	64	7	swish				
			Conv1d	32	7	swish				
			Conv1d	32	3	swish				
			Conv1d	16	3	swish				
			Dense			selu	32			
			Dense			selu	16			
			Dense			selu	8			
5	mse	50	Dense			Linear	3	0,001	0,0745	0,9154
			Conv1d	256	14	swish				
			Conv1d	128	7	swish				
			Conv1d	64	7	swish				
			Conv1d	32	3	swish				
			Conv1d	16	3	swish				
			Dense			selu	32			
			Dense			selu	16			
			Dense			selu	8			
			Dense			Linear	3			

Tabla 2.- Configuración de los modelos, pérdida y accuracy (Parte 1)

Versión	Pérdida	Epochs	Capa	Filtros	Kernels	Activacion	Salida	LerningRate	Loss	Accuracy
6	mse	50	Conv1d	256	14	swish		0,001	0,0708	0,9228
			Conv1d	512	14	swish				
			Conv1d	256	14	swish				
			Conv1d	128	7	swish				
			Conv1d	64	7	swish				
			Conv1d	64	7	swish				
			Conv1d	32	3	swish				
			Conv1d	32	3	swish				
			Dense			selu	32			
			Dense			selu	16			
			Dense			selu	8			
7	mse	60	Dense			Linear	3	0,001	0,0292	0,9173
			Conv1d	256	14	swish				
			Conv1d	512	7	swish				
			Conv1d	256	7	swish				
			Conv1d	128	7	swish				
			Conv1d	64	7	swish				
			Conv1d	64	7	swish				
			Conv1d	32	3	swish				
			Conv1d	32	3	swish				
			Dense			selu	32			
			Dense			selu	16			
8	log_cosh	75	Dense			selu	8	0,001	0,0251	0,9325
			Dense			Linear	3			
			Conv1d	256	14	swish				
			Conv1d	512	7	swish				
			Conv1d	256	7	swish				
			Conv1d	128	7	swish				
			Conv1d	64	7	swish				
			Conv1d	64	7	swish				
			Conv1d	32	3	swish				
			Conv1d	32	3	swish				
			Dense			selu	32			
9	mse	75	Dense			selu	16	0,001	0,0839	0,9131
			Dense			selu	8			
			Dense			Linear	3			
			Conv1d	64	7	swish				
			Conv1d	32	3	swish				
			Conv1d	32	3	swish				

Tabla 3.- Configuración de los modelos, pérdida y accuracy (Parte 2)

5. Conclusiones y Recomendaciones

En vista de los resultados obtenidos en las diferentes etapas evaluadas, no es posible llegar a concluir cual modelo es mejor, ya que nos tienen un buen comportamiento en las etapas de validación y testing pero muy cuestionable desempeño en la predicción. Así es el caso del modelo 8, que presenta el mejor desempeño en la etapa de validación con una accuracy de 0.9325, el mas bajo error relativo medio en el testing para la variable temperatura, el segundo mejor índice para la variable log G y el tercer mejor resultado para el error relativo medio de la velocidad. Pero, resultados muy poco confiables en la predicción, entregando una media para la temperatura y la velocidad por debajo del rango esperado.

Por otro lado, los resultados mas estables, con desviaciones estándar menores, para la etapa de predicción resultan entregados por los modelos cuyos índices en las etapas de validación y testing son peores. Es el caso del modelo 1 que presenta el peor comportamiento en la etapa de validación con un accuracy de 0.8138 y una perdida del 0.3346 y con los errores relativos medios y absoluto mas altos.

Lo descrito anteriormente hace pensar que queda mucho espacio para seguir explorando este caso y que los valores ideales aun están ocultos. Nadase puede concluir respecto a la aleatoriedad de la selección de puntos de la línea observada ya que los modelos no ofrecen ninguna garantía de cumplir adecuadamente con su labor.

Pero no todo es pérdida, este trabajo ha permitido descubrir que las variables, parámetros e hiper-parámetros existentes para los modelos de Redes Neuronales Convolucionales, implementados mediante la librería Keras, son muchos e importantes, todos a la vez, por lo que para llevar a cabo la labor de descubrir la mejor configuración frente a un problema determinado, requiere de tiempo, buena planificación y conocimiento profundo sobre la afectación e impacto en el modelo.

Inteligencia%20Artificial/Apuntes/tareas_alumnos/RNA/Redes%20Neuronales2.pdf

- [9] R. Raina, A. Madhavan, A. Y. Ng, Largescale deep unsupervised learning using graphics processors, in: International Conf. on Machine Learning, 2009.

References

- [1] U. C. de Madrid, Espectros estelares en color real [cited 02 june 2022].
URL <https://webs.ucm.es/info/Astrof/users/jaz/TRABAJOS/COLOR/info.html>
- [2] A. del Barrio, P. Fernández-Miranda, P. Bellón, L. Iglesias, D. González, Inteligencia artificial en radiología: introducción a los conceptos más importantes, Radiología 64 (3) (2022). doi:<https://doi.org/10.1016/j.rx.2022.03.003>.
- [3] A. Iglesias-Puzas, P. Boixeda, Deep learning and mathematical models in dermatology, Actas Dermo-Sifiliográficas 111 (3) (2019). doi:<https://doi.org/10.1016/j.ad.2019.01.014>.
- [4] About kerass (nd) [cited 02 june 2022].
URL <https://keras.io/about/>
- [5] A. Müller, S. Guido, Introduction to Machine Learning with Python, O'Reilly Media, 2016.
- [6] I. C. Education, Redes neuronales (2020) [cited 02 june 2022].
URL <https://www.ibm.com/cl-es/cloud/learn/neural-networks>
- [7] W. B. Services, Cómo se levanta uno e inventa la ia en 1956 (2018) [cited 02 june 2022].
URL <https://ecosistemahuawei.xataka.com/como-se-levanta-uno-e-inventa-la-ia-en-1956/>
- [8] I. Nuevo-Laredo, Historia de las redes neuronales (np) [cited 02 june 2022].
URL <http://www.itnuevolaredo.edu.mx/takeyas/apuntes/>