



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Una red neuronal para la detección de exoplanetas en series temporales de velocidad radial

Tesis presentada para optar al título de Licenciado en Ciencias de la Computación

Luis Agustín Nieto

Director: Dr. Rodrigo F. Díaz

Codirector: Dr. Enrique C. Segura

Buenos Aires, 2020

UNA RED NEURONAL PARA LA DETECCIÓN DE EXOPLANETAS EN SERIES TEMPORALES DE VELOCIDAD RADIAL

El estudio de planetas extrasolares es un campo de investigación relativamente nuevo. Hace apenas 25 años se confirmaba el descubrimiento del primer exoplaneta en torno a una estrella de tipo solar y, gracias a las mejoras en instrumentos y técnicas, este número de cuerpos celestes fue creciendo rápidamente utilizando, principalmente, los métodos de velocidad radial y tránsito.

Misiones como GAIA¹ y TESS², junto a otros proyectos, como el relevo VVV³ o el LSST⁴, están aportando una cantidad cada vez más grande de información astronómica y la comunidad está mirando hacia la ciencia de datos y a las diferentes técnicas de inteligencia artificial como un apoyo importante ante esta avalancha de información.

Ya en los últimos tiempos han comenzado a aparecer trabajos en el campo de planetas extrasolares que hacen uso de estas herramientas, algunos con resultados que incluso superan los obtenidos con las últimas técnicas en el campo.

El método de velocidad radial busca detectar la presencia planetaria mediante caracterizaciones de los movimientos de su estrella central. Los diferentes ruidos provocados por la variabilidad intrínseca de la estrella, sumados al error instrumental y variabilidad temporal en la toma de datos, pueden dificultar la interpretación de los mismos e incluso generar falsas detecciones.

En este trabajo, se propone una red neuronal que reemplaza un cálculo crucial de este método, se generan señales sintéticas de estrellas de tipo solar y se comparan las aplicaciones de ambas implementaciones. La red alcanza un 28% menos de falsos positivos en la detección de planetas con una mejora sustancial en la velocidad de ejecución de cinco órdenes de magnitud, haciéndola ideal para su aplicación en grandes volúmenes de datos.

Los prometedores resultados obtenidos son la base para trabajos futuros cuyo objetivo final es la aplicación sobre señales reales, y esperamos que sirvan como base para reforzar la importancia de los trabajos interdisciplinarios dentro de las ciencias de la computación.

Palabras claves: Astronomía, Velocidad radial, Exoplanetas, Análisis de señales, Redes neuronales, Aprendizaje profundo.

¹ <https://sci.esa.int/gaia>

² Transiting Exoplanet Survey Satellite. <https://tess.mit.edu/>

³ <https://vvvsurvey.org/>

⁴ <https://www.lsst.org/>

A NEURAL NETWORK FOR EXOPLANET DETECTION IN RADIAL VELOCITY TIME SERIES

The study of extrasolar planets is a relatively new field of research. Just over 25 years ago, the discovery of the first exoplanet around a solar-type star was confirmed and, thanks to technical and instrumental improvements, the number of known exoplanets has been increasing rapidly, with most of them being detected using the radial velocity and transit methods.

Missions such as GAIA⁵ and TESS⁶, along with other projects, such as the VVV survey⁷ or the LSST⁸, are contributing an ever greater amount of astronomical information, and the community is looking towards data science and different artificial intelligence techniques as an important support in the face of this avalanche of information.

In recent times, some works began to emerge in the field of extrasolar planets that use these techniques, some with results that even exceed those obtained with the latest techniques in the field.

The radial velocity method seeks to detect the presence of a planet through characterizations of the movements of its central star. The different noises caused by the intrinsic variability of the star, added to the instrumental error and temporal variability in data collection, can make data difficult to interpret and even lead to false detections.

In this work, a neural network is proposed that replaces a crucial calculation of this method, synthetic stellar signals from solar-type stars are generated and the applications of both implementations are compared. The network achieves 28% fewer false positives in planet detection and has a substantial improvement in execution speed of five orders of magnitude, making it ideal for application in large volumes of data.

The promising results obtained are the basis for future work whose final objective is the application on real signals, and we hope that they will serve as a basis to reinforce the importance of interdisciplinary work within the field of computer science.

Keywords: Astronomy, Radial velocity, Exoplanets, Signal analysis, Neural networks, Deep learning.

⁵ <https://sci.esa.int/gaia>

⁶ Transiting Exoplanet Survey Satellite.<https://tess.mit.edu/>

⁷ <https://vvvsurvey.org/>

⁸ <https://www.lsst.org/>

AGRADECIMIENTOS

A mis directores, Rodrigo y Enrique, por las largas revisiones y la paciencia.

A mis padres y hermanos, amigos de Bariloche y de la facultad. Que siempre estuvieron.

A Cande, Mateo, Trini, Fran y Nacho. Que fueron apareciendo.

A mis abuelos y nonos, Pablito y a mi viejo. Que ya no están (pero en realidad sí).

A la CNEA, que como institución me dio poder de cómputo y como lugar de trabajo un montón de buena gente.

A Damien Ségransan, Jean-Baptiste Delisle y Nathan Hara, del observatorio de Ginebra, por las ideas y fragmentos de código.

A la educación pública, que con sus vaivenes sigue siendo la principal herramienta para la inclusión y el desarrollo.

*Down from the door where it began.
Now far ahead the Road has gone,
And I must follow, if I can,
Pursuing it with eager (weary) feet,
Until it joins some larger way
Where many paths and errands meet.
And whither then? I cannot say.*

– J.R.R. Tolkien, *The Fellowship of the Ring*

A mi familia.

Índice general

1.. Introducción	1
1.1. Detección y caracterización de exoplanetas	3
1.1.1. Velocidad radial	3
1.1.2. Periodogramas	5
1.1.3. False alarm probability (FAP)	6
1.1.4. Astrónomo virtual	8
1.2. Redes neuronales	12
1.2.1. Redes convolucionales	13
2.. Materiales y métodos	19
2.1. Generación de datos	19
2.1.1. Series temporales	20
2.1.2. Periodogramas	25
2.1.3. FAP	25
2.2. Red neuronal	26
3.. Resultados y análisis	29
3.1. Calidad de las predicciones	29
3.2. Tiempo de ejecución	33
3.3. Características de las detecciones	35
3.4. Método completo	37
4.. Discusión y conclusiones	41
4.1. Dificultades y posibles mejoras	41
4.2. Trabajo futuro	42
Apéndice	49
A.. Redes neuronales, breve historia y evolución	51
A.1. Neuronas artificiales y redes neuronales	52
A.2. Redes profundas	55
B.. Modelos descartados	59

1. INTRODUCCIÓN

“The cosmos is within us. We are made of star-stuff. We are a way for the universe to know itself.”

– Carl Sagan, Cosmos

El estudio de planetas extrasolares (planetas que orbitan estrellas diferentes a nuestro Sol, también llamados exoplanetas) es un campo de investigación relativamente nuevo. Si bien la primera evidencia de la existencia de este tipo de cuerpos data de 1917¹ no fue hasta la década de los noventa que tuvieron lugar las primeras detecciones confirmadas. En 1992, Wolszczan y Frail [WF92] publicaron un trabajo en el que analizaban las variaciones del período rotacional del pulsar PSR1257+12 y concluyeron que esas variaciones eran el producto de la presencia de, al menos, dos planetas de tamaño similar a la tierra. Tres años después, Mayor y Queloz [MQ95], descubrieron el primer exoplaneta orbitando una estrella del tipo solar, **51 Peg**. El entusiasmo que siguió a este primer hallazgo precipitó el descubrimiento de otros mundos. Sin embargo, para el final del siglo veinte, solo se conocía una docena de planetas orbitando estrellas similares al Sol.

Las detecciones de los primeros planetas en tránsito² fueron solamente cinco años después de **51 Peg** [HMBV00] y [CBLM00]. En este segundo trabajo no solo pudieron derivar el radio de un exoplaneta, sino que también confirmaron la existencia de los extraños *Hot Jupiters*³ (Júpiter calientes).

Los años que siguieron fueron de mucho éxito, y se fueron descubriendo planetas con masas cada vez más pequeñas con los métodos de velocidad radial y tránsito. Sin embargo, solamente en los últimos tiempos se pudo empezar a sondear con detalle los planetas de masa y radio más pequeños de la población planetaria. Esto se debió en gran medida al lanzamiento de la misión Kepler [BKB⁺11] y a la mejora de los métodos utilizados para extraer información de los datos de velocidad radial (por ejemplo [FAEA⁺16]). En las figuras 1.1 y 1.2 se puede ver un resumen de estos avances.

¹ Landau, Elizabeth (12 November 2017). “Overlooked Treasure: The First Evidence of Exoplanets”. NASA. Retrieved 1 November 2017. <https://www.jpl.nasa.gov/news/news.php?feature=6991>

² El planeta se cruza en la línea de visión entre el observador y la estrella disminuyendo su luminosidad, estos cambios son medidos a lo largo del tiempo y se puede inferir su presencia.

³ Planetas gigantes similares a Júpiter pero con períodos cortos que, basados en los modelos históricos de formación de planetas, no se esperaba que existiesen.

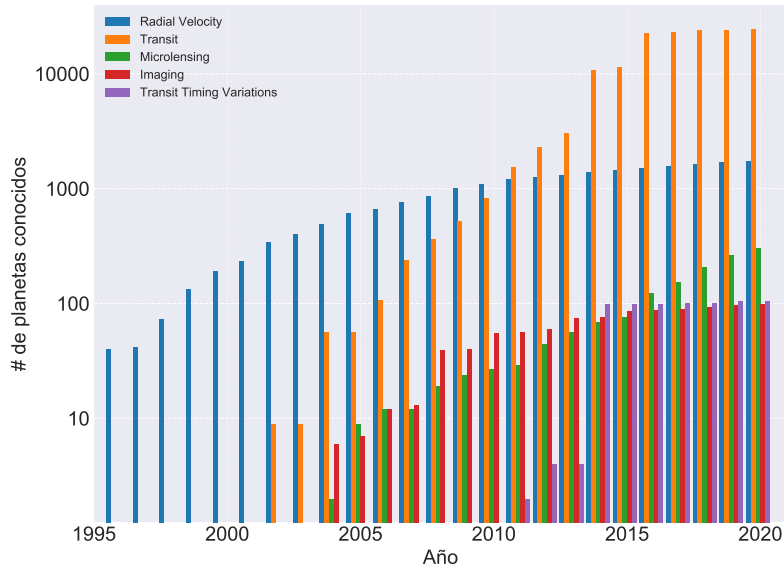


Fig. 1.1: Cantidad de exoplanetas detectados a lo largo de los años, detallado por método utilizado.

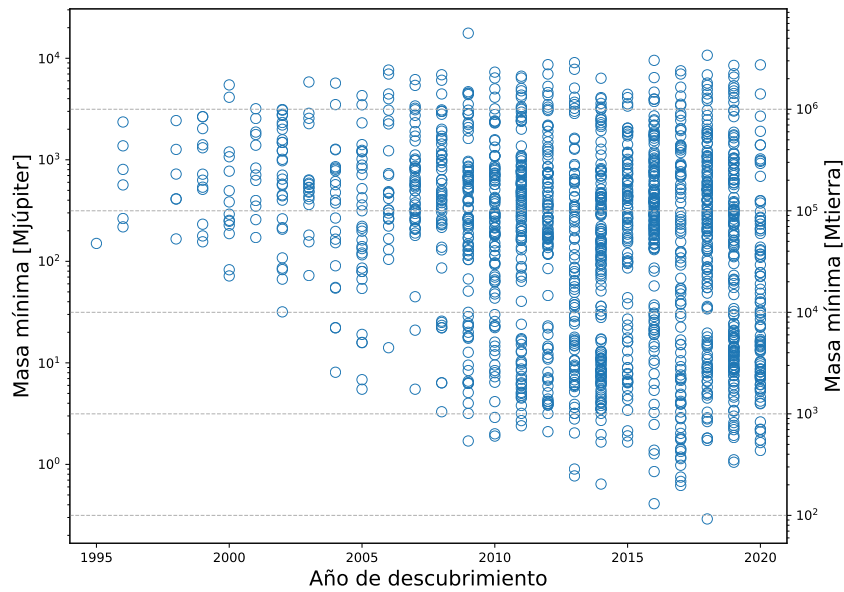


Fig. 1.2: Evolución de las masas de los exoplanetas detectados a lo largo del tiempo. Las mejoras técnicas permitieron encontrar exoplanetas con masas cada vez más pequeñas.

Misiones como GAIA⁴ y TESS⁵ están aportando una cantidad cada vez más grande de datos astronómicos, y la comunidad está mirando hacia la ciencia de datos y a las diferentes técnicas de inteligencia artificial como un apoyo importante ante esta avalancha de información. Ya en los últimos tiempos han comenzado a aparecer trabajos en el campo

⁴ <https://sci.esa.int/gaia>

⁵ Transiting Exoplanet Survey Satellite. <https://tess.mit.edu/>

de planetas extrasolares que hacen uso de estas herramientas, por ejemplo: la clasificación y veto de candidatos con tránsitos [APS16], [AGM⁺18], [CRG⁺19], [MJC⁺15], [SV18]; el estudio de atmósferas planetarias [MNFSH18], [Wal16], e incluso la clasificación de planetas según su potencial habitabilidad [SMB⁺18].

Cuando se comenzó esta tesis, no había trabajos que aborden, de alguna manera, la detección de planetas mediante la técnica de velocidad radial. Recién en octubre de 2020 apareció el primero [dBVS⁺20], el tiempo transcurrido entre los trabajos utilizando el método de tránsito y el de velocidad radial da cuenta de la complejidad de este método y de los varios enfoques posibles.

El objetivo de esta tesis es atacar esa problemática y explorar los alcances de las redes neuronales en el análisis de señales, buscando mecanismos para complementar los actuales⁶.

1.1. Detección y caracterización de exoplanetas

La mayoría de los métodos usados para detectar planetas extrasolares siguen caminos más o menos sinuosos y no consiguen detectarlos de manera directa, sino tan sólo mediante los efectos causados sobre características de su estrella central: pequeños cambios en su velocidad, posición o brillo; nos dan indicios de la presencia planetaria. A continuación, se hará una pequeña descripción del método de velocidad radial que será el utilizado a lo largo de este trabajo.

1.1.1. Velocidad radial

En un sistema de dos cuerpos y siguiendo la tercera ley de Newton y la ley de gravitación universal, cada cuerpo ejerce una atracción gravitatoria sobre el otro. Cuando uno de ellos es mucho más masivo, se suele suponer que el pequeño orbita al grande; pero, en realidad, ambos cuerpos orbitan el centro de masa, o baricentro, del sistema. La masividad de la estrella hace que este baricentro se encuentre mucho más cerca de ella (o incluso dentro) y experimente una pequeña órbita, mientras que la órbita del planeta es mucho más pronunciada.

El observador mira en línea recta a la estrella, es por esto que (dependiendo de su posición con respecto a ella y a la órbita del planeta) puede ver este “bamboleo” como si la estrella se acercase y alejase a intervalos regulares. Esto está ejemplificado en las figuras 1.3 y 1.4.

⁶ Parte de este trabajo fue realizado en el observatorio de Ginebra y financiado por el proyecto Seed Money Grant 2018 “Towards the Detection of Earth Analogues (TDEA)” de la STATE SECRETARIAT FOR EDUCATION, RESEARCH AND INNOVATION, Suiza.

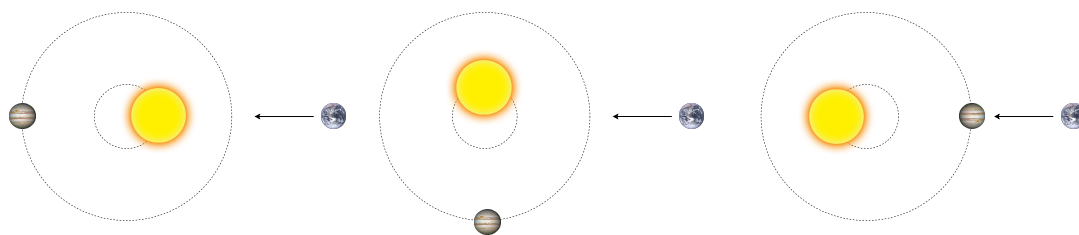


Fig. 1.3: Visto desde la tierra, la estrella se aleja.

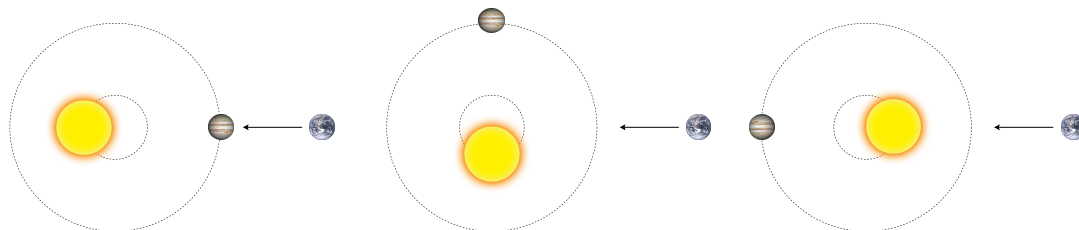


Fig. 1.4: Visto desde la tierra, la estrella se acerca.

La velocidad radial es, justamente, la velocidad con la que la estrella se mueve sobre esta línea visual. Será negativa cuando la estrella se acerque y positiva cuando se aleje⁷. Esta velocidad puede medirse con gran precisión utilizando el efecto Doppler de la luz⁸. A lo largo del tiempo, se toman mediciones de la velocidad de la estrella y se registran en lo que se llaman *series temporales*. El objetivo es caracterizar el movimiento estelar buscando comportamientos periódicos que indiquen la presencia de un planeta. En la figura 1.5. se muestra un ejemplo de una serie temporal. Los datos son los de [MQ95] y, al superponerlos a los puntos de medición, se puede ver la señal periódica inducida por el planeta.

⁷ Esta es una sencilla aproximación que alcanza para presentar el problema, para más detalles sobre la metodología e implementación se recomienda [Día18].

⁸ El efecto Doppler es un cambio aparente en la frecuencia de una señal cuando la fuente se mueve con respecto al observador. El ejemplo clásico es el de la sirena de la ambulancia: el sonido que emite siempre es el mismo; pero, si se acerca, lo escuchamos más agudo y si se aleja, más grave. Con la luz pasa algo similar, por lo que podemos medir cuando una fuente se está acercando o alejando si se “corre” al azul o al rojo.

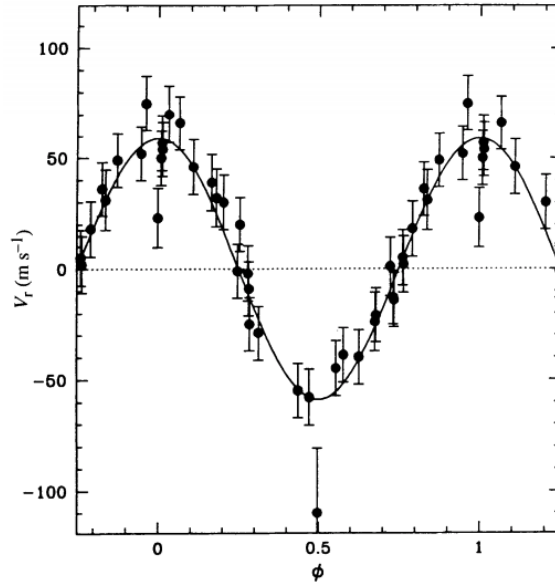


Fig. 1.5: Movimiento orbital de 51 peg. La línea sólida representa el movimiento orbital calculado mostrado sobre las mediciones.

1.1.2. Periodogramas

El periodograma de Lomb-Scargle ([ZK09],[KJY17]) para detección de señales se ha utilizado extensivamente en el campo de la búsqueda de exoplanetas, especialmente en combinación con la técnica de velocidad radial. En su forma más simple, funciona en una forma similar a Fourier: el usuario define una serie de frecuencias candidatas y el método ajusta los datos de velocidad radial, $x(t)$, usando una base de senos y cosenos de la frecuencia candidata, buscando generar un modelo que ajuste a los datos observados. Cuanto mayor ganancia tenga este modelo contra el modelo constante (sin variabilidad), mayor potencia tendrá la frecuencia candidata.

La bondad de cada ajuste (el sinusoidal y el plano) se analiza a través del estadístico $\tilde{\chi}^2$. La mejora aportada por el modelo sinusoidal se cuantifica con la diferencia de $\tilde{\chi}^2$ (normalizada de alguna manera, por ejemplo usando $\tilde{\chi}^2$ del modelo constante). De esto, se deducen las siguientes fórmulas.

$$P_x(\omega) = \frac{1}{2} \left\{ \frac{\left[\sum_{t=1}^T x(t) \cos \omega(t - \tau) \right]^2}{\sum_{t=1}^T \cos^2 \omega(t - \tau)} + \frac{\left[\sum_{t=1}^T x(t) \sin \omega(t - \tau) \right]^2}{\sum_{t=1}^T \sin^2 \omega(t - \tau)} \right\}$$

Donde P_x es la potencia en función de la frecuencia ω , y τ está definido como:

$$\tan(2\omega\tau) = \frac{\left(\sum_{t=1}^T \sin 2\omega t \right)}{\left(\sum_{t=1}^T \cos 2\omega t \right)}.$$

Luego se genera un vector con la potencia de cada una de las frecuencias candidatas y el resultado es que, en este vector, el valor más alto indica la frecuencia que tiene mayores posibilidades de ser la buscada.

Como su nombre lo indica, en el periodograma en vez de utilizar la frecuencia de la señal se usa el período. La conversión es sencilla ya que el período es la inversa de la frecuencia. En la figura 1.6. se puede ver una serie temporal (con una clara señal periódica) y en 1.7 el periodograma resultante de haber aplicado el método.

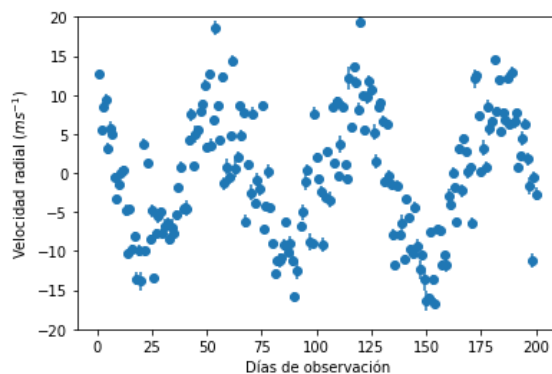


Fig. 1.6: Mediciones de velocidad radial.

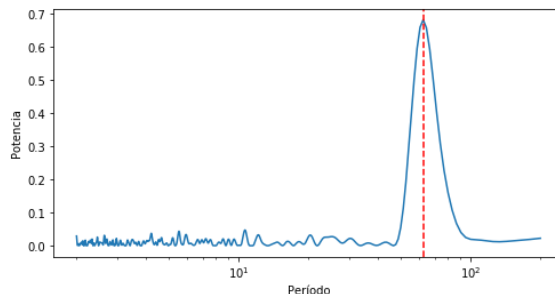


Fig. 1.7: Periodograma. La línea roja marca el pico que representa el período con mayor potencia.

1.1.3. False alarm probability (FAP)

Una vez construido el periodograma, se necesita una herramienta que ayude a decidir si el pico máximo tiene la suficiente significancia estadística como para ser considerado una señal periódica procedente de la influencia de un planeta⁹ o si es solo ruido.

Esto puede asociarse con un *likelihood-ratio test* (test de razón de verosimilitud) donde se busca contrastar una hipótesis base o *nula* (H_0) evaluando la bondad de ajuste de dos modelos estadísticos en competencia, basándose en la razón de sus probabilidades. Para

⁹ Para simplificar, a veces se dirá que un pico “es un planeta” o que “no es un planeta”.

los periodogramas de Lomb-Scargle la hipótesis base es que las observaciones provienen de ruido blanco y errores Gaussianos (es decir, que “no hay planeta”).

Como en todo test, estamos expuestos a los errores de tipo I y II. Los errores de tipo I, o de “falsa alarma”, se producen cuando se rechaza incorrectamente H_0 (es decir, se decide que un pico es un planeta cuando no lo es). Los errores de tipo II son cuando incorrectamente no rechazamos H_0 , en este caso sería decidir que un pico no es planeta cuando en realidad lo era.¹⁰

Las falsas alarmas son consideradas las más peligrosas, de ahí la importancia de estimar la “probabilidad de falsa alarma” o *false alarm probability* (de ahora en adelante FAP), asociada con una señal candidata.

Dado un pequeño valor crítico FAP_* , usualmente entre 10^{-3} y 0.1, podemos decir que la señal candidata tiene la suficiente significancia estadística si su valor FAP cumple con $FAP < FAP_*$. Estos valores suelen expresarse en \log_{10} y en este trabajo se tomará FAP_* como 0.05, que es el valor más usado, y se redondeará $\log_{10}(FAP_*)$ a -1.3.

Tenemos dos formas de calcular este número: una es analítica y otra es mediante un algoritmo tipo Monte Carlo.

La fórmula analítica utilizada es la descrita por Baluev en [Bal08] y tiene como ventaja que es mucho más rápida de calcular que el Monte Carlo, Sin embargo, para cierto rango de resultados, deja de ser confiable. En particular, siguiendo los comentarios del paper, se considerará que, si este resultado es mayor o igual $\log_{10}(0.01)$, deja de ser confiable y se tiene que usar el otro método.

Para realizar el Monte Carlo, se genera un número de periodogramas bajo la hipótesis nula (es decir, periodogramas sin la señal que generaría el pico al que se le quiere calcular el FAP) y se ve cual es la probabilidad de que, por azar, el ruido genere otro pico de esa potencia, independientemente de en que período ocurra. La necesidad de generar esa cantidad de series temporales (con sus correspondientes periodogramas) lo convierte en un método mucho más lento que el anterior, especialmente cuando se tiene en cuenta que se está apuntando a analizar gran cantidad de datos.

¹⁰ También se suele referir a estos errores como falsos positivos y falsos negativos.

Su pseudocódigo es el siguiente:

Algorithm 1 FAP Monte Carlo

```

1: procedure CALCULAR_FAP(params_vel_radial, pot_pico_max)
2:    $N \leftarrow 150$ 
3:    $p\_max = []$ 
4:   loop  $N$  veces
5:      $vr\_ruido \leftarrow \text{gen\_vel\_radiales}(params\_vel\_radial)$ 
6:      $pg \leftarrow \text{gen\_periodograma}(vr\_ruido)$ 
7:     if  $\max(pg) > pot\_pico\_max$  then
8:        $p\_max.append(1)$ 
9:    $sum\_pg \leftarrow \text{sumar}(p\_max)$ 
10:  return  $\log_{10}(sum\_pg/N)$ 

```

Una vez que se calcule el valor FAP de un pico, se considerará que el mismo es un planeta si, como ya se mencionó, $FAP < FAP_*$.

1.1.4. Astrónomo virtual

Ya se explicó como, partiendo de las mediciones de velocidad radial de una estrella, se produce un periodograma y lo que puede representar el pico más alto. Además se presentó un método para decidir si ese pico es significativo o no. Pero puede haber otros planetas “escondidos” en la serie.

Una estrategia posible para tratar de encontrarlos es remover de la serie la primera señal detectada usando los parámetros ajustados, y generar un nuevo periodograma a partir de los residuos. El razonamiento es que, si hay otra señal planetaria, esta emergerá como un pico máximo en este nuevo periodograma.

Todo esto se utiliza en un mecanismo simple para la búsqueda de planetas que suele ser el primer paso para analizar una serie temporal de velocidades radiales. Aunque existen técnicas mucho más sofisticadas (p. ej., [DSU⁺16]), por la extensión de su uso nos concentramos en ella como punto de comparación de nuestro modelo de aprendizaje automático. En nuestra implementación lo llamaremos *astrónomo virtual*.

Este es su pseudocódigo:

Algorithm 2 Astrónomo Virtual

```

1: procedure ASTRONOMO_VIRTUAL(vel_radiales)
2:   pg ← gen_periodograma(vel_radiales)
3:   pico_max ← dame_pico_max(pg)
4:   planetas = []
5:   while es_planeta(pico_max, pg) do
6:     planetas.append(pico_max)
7:     vr_nueva ← eliminar(pico_max, vel_radiales)
8:     pg ← gen_periodograma(vr_nueva)
9:     pico_max ← dame_pico_max(pg)
10:  return planetas
  
```

En las figuras 1.8, 1.9 y 1.10, hay un ejemplo de aplicación sucesiva en una estrella con dos planetas.

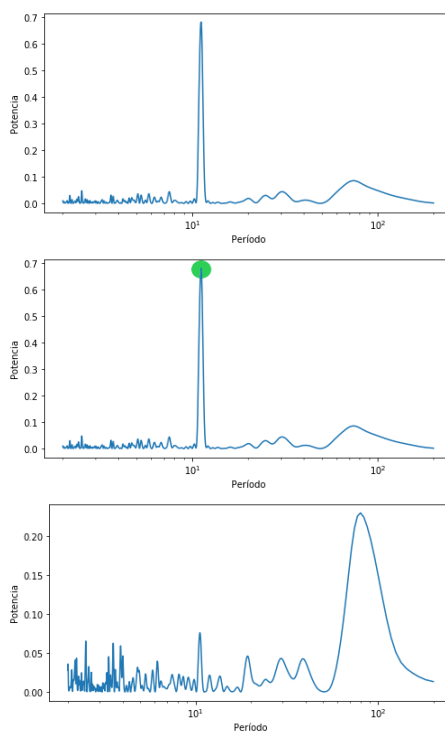


Fig. 1.8: Detección del primer planeta y generación del nuevo periodograma. El segundo planeta que estaba “oculto” emerge como un nuevo pico máximo.

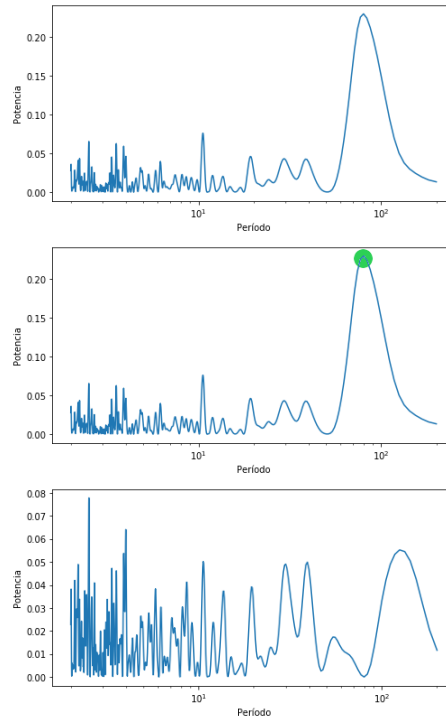


Fig. 1.9: Detección del segundo planeta y generación del nuevo periodograma.

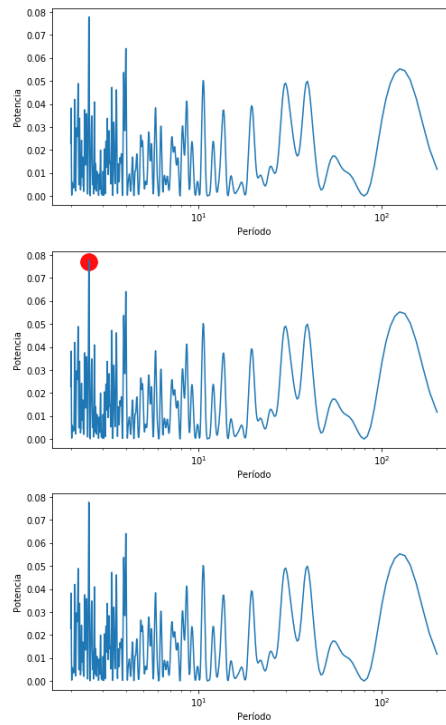


Fig. 1.10: El pico más alto no es planeta. Se finaliza la ejecución.

Esta es una ejecución ideal pero, por la naturaleza del método (y del mismo FAP), se puede incurrir en los errores de tipo I y II ya mencionados y, además, muchas veces no se pueden encontrar todas las señales. Estos son algunos de los problemas que pueden llegar a surgir al usar este algoritmo:

- **El periodograma solo es sensible a señales sinusoidales:** Si la órbita del planeta es excéntrica, la potencia se divide entre un pico a la frecuencia del planeta y sus armónicos, por lo que pierde fuerza y puede pasar desapercibido.
- **La confianza en los resultados puede ir variando:** Al sustraer una señal de las velocidades radiales, la suposición implícita es que los parámetros de esa señal son conocidos de forma exacta. Esto es solo aproximadamente cierto con los planetas de gran amplitud (gigantes), pero para nada con los planetas de baja masa. Esto puede dar menos (o más) confianza en la detección de los planetas sucesivos de lo que corresponde a partir de un análisis más robusto.
- **No encontrar todos los planetas:** Si hay perturbaciones planetarias en la serie temporal, pero ninguna de ellas está en el máximo del periodograma, el algoritmo se detiene (suponiendo que la clasifica correctamente como ruido), por lo que esos planetas no serán encontrados¹¹. Dentro de este punto también se puede mencionar a los **falsos negativos**, donde el pico más grande corresponde a un planeta y el FAP dictamina que no lo es. No se considerará tan grave como los falsos positivos, pero es un error a considerar.
- **Falsos positivos:** El FAP dictamina que un pico máximo corresponde a un planeta cuando en realidad no lo es. Este es el error que se considerará más grave y, paradójicamente, puede llevar a encontrar más planetas ya que el algoritmo continuará su ejecución.

A lo largo del trabajo se prestará especial atención a los últimos dos puntos.

¹¹ En las simulaciones realizadas más adelante, este caso resultó ser el más común, tener “planetas inalcanzables”.

1.2. Redes neuronales

En los últimos años se produjo una explosión de trabajos en el área de la Inteligencia artificial, términos como *machine learning* (aprendizaje automático), *neural networks* (redes neuronales) o *deep learning* (aprendizaje profundo) son moneda corriente tanto en resúmenes de papers como en los medios de comunicación. Se podría llegar a pensar que son cosas muy modernas y revolucionarias, pero la realidad es que las ideas fundamentales en las que se basan las técnicas actuales (y no tanto) nacieron casi con la computación. El impresionante avance en hardware y hardware gráfico impulsado, en un inicio, por la industria de los videojuegos, junto al inmenso acceso a datos provisto por Internet; hizo viable el uso de técnicas que hace treinta o cuarenta años eran curiosidades matemáticas.

También colaboraron en esta explosión gigantes de la industria como Google, Facebook o Microsoft que, para sus plataformas, desarrollaron bibliotecas y herramientas que luego fueron liberadas al público general. Ahora no solo cualquiera podía usarlas, todos podían colaborar para modificarlas y mejorarlas. Motores como *Tensorflow*, *Pytorch* o *CNTK* implementaron la engorrosa matemática que subyace a las redes neuronales y bibliotecas como *Keras* o *Scikit-learn* permiten abstraerse aun más, facilitando enormemente su definición y uso. Ahora, para implementar una red neuronal, se necesitan unas pocas líneas de código (prácticamente descriptivo) y ya se puede acceder a todo el poder de cómputo del que los equipos modernos son capaces.

Esta “democratización” en el uso de la Inteligencia artificial (junto a la popularización de *Python* como un lenguaje de programación ameno, sencillo de aprender y con una gran cantidad de bibliotecas) permitió que no solo los matemáticos y computadores puedan sacarle provecho al área, por lo que hoy, se pueden encontrar trabajos de prácticamente cualquier ciencia con alguna implementación de *Machine learning* algunos con resultados realmente impresionantes. Sin ir más lejos, en [SV18] (uno de los trabajos ya mencionados) utilizando una red neuronal, encontraron dos planetas nuevos en datos antiguos que ya habían sido descartados en análisis manuales.

Dado que son las utilizadas en este trabajo, se hará un brevísimo resumen de conceptos generales, historia y uso de las redes convolucionales. Se presuponen conocidos los términos básicos como capa o *backpropagation* y las generalidades de las neuronas artificiales (función de activación, peso sináptico, bias, etc.). Sin embargo, para aquellos que no estén familiarizados, se incluye en el apéndice A un pequeño repaso tanto histórico como práctico que abarca desde la definición de una neurona artificial hasta las generalidades de las redes profundas. Los que deseen ahondar aún más conceptos teóricos y de implementación pueden referirse a la bibliografía, en particular [HKP91] y [GBC16].

1.2.1. Redes convolucionales

Las redes profundas han logrado muy buenos resultados prácticos utilizadas en una gran variedad de problemas; pero, para el tratamiento de imágenes, poseen una desventaja, no tienen en cuenta la estructura espacial de las mismas. No es que no se puedan usar, simplemente no son tan efectivas. Aquí es donde aparecen las redes convolucionales que son redes que utilizan *convoluciones* en vez de las operaciones matriciales clásicas. Fueron creadas para trabajar con imágenes, pero se pueden aplicar perfectamente en datos de una sola dimensión que tengan una relación entre si.

Estas redes, como la gran mayoría, tuvieron sus comienzos en estudios neurobiológicos que luego se imitaron con modelos matemáticos. Estos son algunos hitos que fueron fundamentales para su creación:

- En 1959, Hubel y Wiesel [HW59] insertaron microelectrodos en la corteza visual de un gato y analizaron la reacción de las neuronas al mostrarle diferentes patrones de formas, luz y oscuridad y encontraron que:
 - Cada neurona responde individualmente a pequeñas regiones del total del campo visual, esta región se conoce como **campo receptivo**. El tamaño y la ubicación de estos campos receptivos varia sistemáticamente a través de la corteza para formar un mapa completo del espacio visual, y no solo eso, las neuronas vecinas tienen campos receptivos similares que se superponen entre sí.
 - Algunas neuronas disparaban rápidamente cuando se les presentaban líneas en un ángulo, y otras lo hacían para líneas en otro. Llamaron a estas neuronas *simple cells* (células simples). Mientras que otro grupo respondía mejor a líneas en cierto ángulo moviéndose en una dirección, esto es, no les “importaba” la posición específica en el campo receptivo. Reconocían la línea, pero eran insensibles a la traslación. A estas las llamaron *complex cells* (células complejas).

El gran resultado de este trabajo fue mostrar cómo el sistema visual **construye una imagen partiendo de estímulos simples, yendo hacia representaciones cada vez más complejas**.

- Tomando estas ideas, en 1980 Kuniyiko Fukushima [Fuk80] presentó el **Neocognitron** e introdujo dos tipos de capas, las convolucionales y las de *downsampling* (disminución de resolución). La capa convolucional contiene unidades cuyos campos receptivos cubren una pequeña parte de la capa anterior y las capas de *downsampling* cubren partes de las capas convolucionales ayudando a clasificar correctamente objetos en escenas donde los mismos están corridos.

- Unos años después, en 1989, Yann LeCun [LBD⁺89] usó un algoritmo tipo *backpropagation* en la arquitectura de Fukushima y creó, para el correo, un sistema automático que fue capaz de interpretar correctamente los códigos postales manuscritos en las cartas que se recibían. Este trabajo se considera fundacional del área moderna de visión por computadora.

Estos y otros trabajos a lo largo de los años nos legaron las redes convolucionales modernas que, a grandes rasgos, realizan los siguientes pasos al momento de procesar una imagen:

- **Extracción de características** (*Feature extraction*): Cada neurona toma sus entradas sinápticas de un campo receptivo local en la capa anterior, esto la obliga a extraer características locales. Una vez que se ha extraído una característica su ubicación exacta se vuelve menos importante, siempre que su posición relativa a otras características se conserve aproximadamente.
- **Mapeo de características** (*Feature mapping*): Cada capa de la red se compone de múltiples mapas de características, cada uno tiene la forma de un plano dentro del cual las neuronas individuales están obligadas a compartir el mismo conjunto de pesos sinápticos. Esta segunda forma de restricción estructural tiene los siguientes efectos beneficiosos:
 - Invariancia en la traslación, lograda por realizar la operación de convolución sobre un mapa de características con un *kernel* pequeño seguido de la aplicación de la función de activación. Un *kernel* es una pequeña matriz de pesos con la que se va convolucionando la imagen y es el equivalente a los campos receptivos biológicos.
 - Reducción en el uso de parámetros, lograda a través del uso de pesos compartidos.
- **Submuestreo** (*Subsampling*): Suele seguirse cada capa convolucional por una capa que realiza un promedio local y submuestreo, por lo que se reduce la resolución del mapa de características. Esta operación tiene el efecto de reducir la sensibilidad de la salida del mapa de características a las variaciones y otras formas de distorsión.

De esta forma se obtiene una red profunda que imita a la corteza visual. En sus primeras capas detecta líneas o curvas; pero, a medida que se va avanzando, detecta formas más complejas como un rostro o una silueta.

Se mencionó que para realizar un mapa de características se aplica un *kernel* sobre una

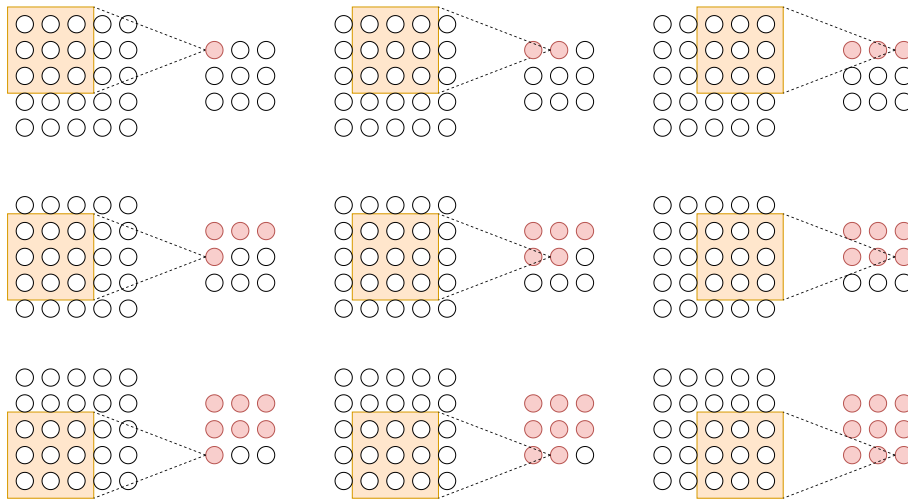


Fig. 1.11: Aplicación de un kernel generando un mapa de características.

imagen, recordemos que lo que busca es imitar al cerebro, por lo que este *kernel* sobre la imagen es el campo receptivo que “ve” una neurona. Al ir moviendo y aplicando el *kernel* se obtiene una capa que representa la mencionada reacción de las primeras células de la corteza visual.

En la figura 1.11, se ve un ejemplo de como se genera un mapa de características al aplicar un *kernel* de 3×3 a una imagen de 5×5 ¹².

El tamaño del *kernel* es uno de los parámetros que se eligen al momento de crear la red. Suelen ser pequeños para extraer características locales; pero, como se quiere encontrar muchas de estas características, lo que se hace es usar muchos *kernels*. Los mismos tienen todos la misma dimensión, pero los pesos que van aprendiendo son diferentes (de esta forma buscan cosas diferentes). Este conjunto de *kernels* es lo que se llama un **filtro**.

Ahora que se tienen los conceptos de *kernel* y filtro, se pasará a algo más concreto y se verá por qué son necesarias las capas de *downsampling*. El ejemplo clásico es usar el conjunto de datos de MNIST, que consta de miles de imágenes de dígitos manuscritos, están en blanco y negro y son de 28×28 píxeles.

En la figura 1.12 se ve como al aplicar un filtro de 32 *kernels* de 3×3 se obtienen 32 mapas de activación de 26×26 cada uno¹³, por lo que la cantidad de neuronas de esta

¹² Para simplificar, los ejemplos que veremos están presentados sobre matrices en dos dimensiones presumiendo imágenes en blanco y negro, si fuesen a color se necesitaría agregar una dimensión a la entrada y al kernel por lo que tendríamos tres filtros, esos tres filtros se suman junto con el *bias* y conformarán una salida, cómo si fuese un solo canal.

¹³ La dimensión de salida o de aplicar un kernel de dimensiones k a una imagen de tamaño w se puede calcular con esta fórmula: $o = w - k + 1$. Conceptos como *padding* o *stride* modifican esta fórmula, pero son innecesarios para el ejemplo.

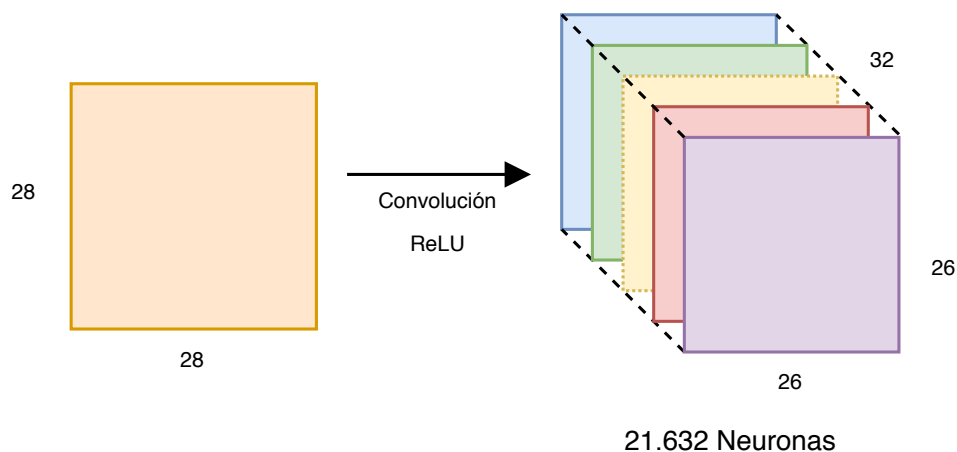


Fig. 1.12: Primera capa oculta.

primera capa oculta es de 21.632¹⁴.

Este número es demasiado grande para una imagen tan pequeña, y es solo la primera capa. Agregar más haría crecer este número de manera que se complicaría cada vez más el entrenamiento y perdería la capacidad de abstraerse, aquí es donde entra la capa de *downsampling*.

Para implementar esta capa de *downsampling* hay varias opciones, pero la más utilizada es la de *maxpooling*. En este mecanismo, ejemplificado en la figura 1.13, se busca reducir en un cuarto a los parámetros, dividiendo la matriz de entrada en partes y quedándose con el valor máximo de cada una. De esta forma se busca preservar las características más importantes y disminuir las dimensiones, y por ende, la cantidad de parámetros entrenables.

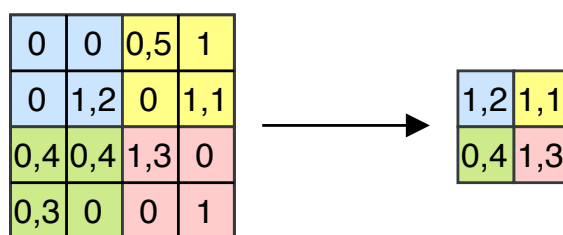


Fig. 1.13: Maxpooling.

En la figura 1.14 se puede ver el resultado de aplicarle esta capa a la salida anterior.

¹⁴ Cabe mencionar que, por el tipo de problema, la función de activación es una tipo ReLU (Rectified Linear Unit) que simplemente devuelve el máximo entre 0 y el valor de entrada.

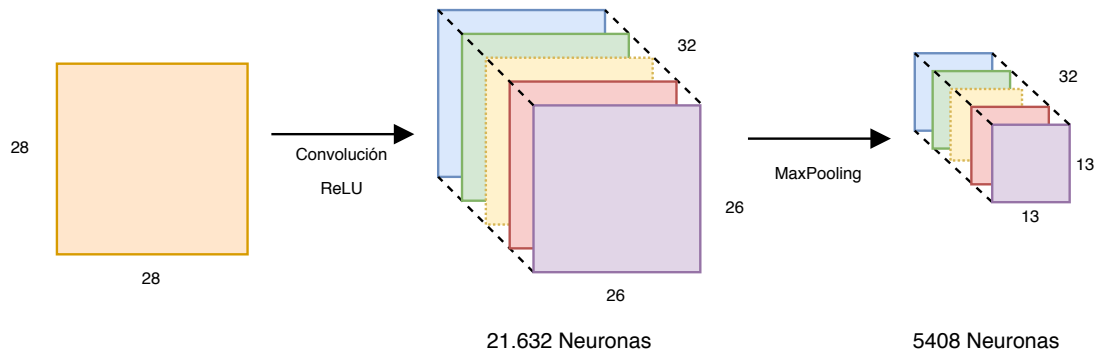


Fig. 1.14: Aplicación de *maxpooling* para reducción de dimensionalidad.

La elección del número de capas y cómo se intercalan depende del problema a resolver y no hay una regla general¹⁵.

La red convolucional extrae las características más relevantes de la imagen, luego se conecta a una o varias capas totalmente conectadas y a una clásica capa de salida para clasificación. Esta capa, nuevamente, depende del problema y en este caso particular es una neurona por cada categoría y una función de activación *softmax*. Lo que hace esta función es que, en cada salida quede reflejada la probabilidad de que esa entrada pertenezca a cada clase. Toda esta última parte es la que, usando la información extraída en la parte convolucional, “decide” a que categoría pertenece la entrada.

En la figura 1.15 se muestra un esquema general de la red de este ejemplo.

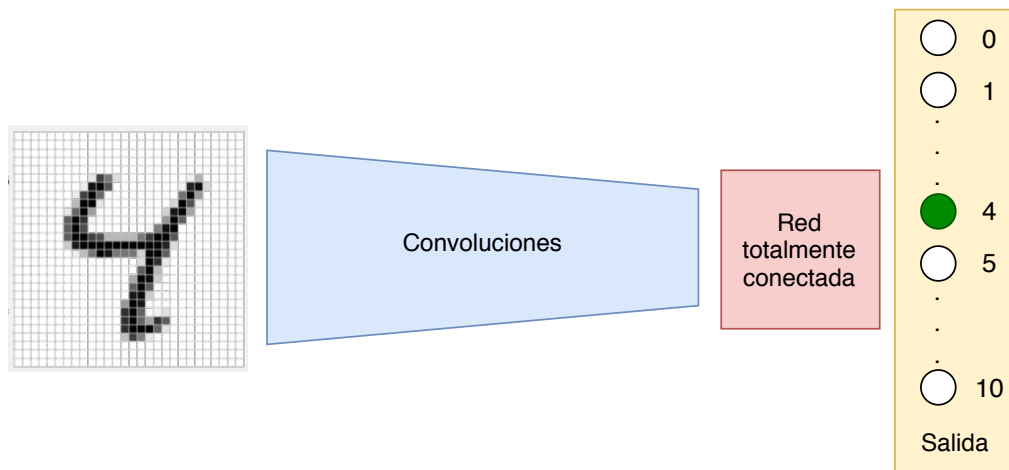


Fig. 1.15: Red completa.

Las pruebas del rendimiento de las diferentes arquitecturas se realizan entrenando, validando y testeando con entradas etiquetadas buscando disminuir el *overfitting* (sobre-

¹⁵ En la práctica, se suele recomendar usar varios “bloques” de dos capas convolucionales seguidas por una de *maxpooling* pero, nuevamente, es un resultado empírico.

ajuste) de la red, esto es, que aprenda muy bien los datos con los que fue entrenada y no pueda generalizar ante datos nuevos. Para ayudar a lidiar con esto, una de las cosas que se pueden usar son las capas de *dropout*. Son capas que aleatoriamente “apagan” algunas neuronas con el objetivo es tratar de disminuir la co-dependencia que se pueden generar entre ellas para aumentar la robustez de cada una y, con ello, el desempeño general.

2. MATERIALES Y MÉTODOS

-Man said, "AC, is this the end? Can this chaos not be reversed into the Universe once more?
Can that not be done?"
-AC said, "THERE IS AS YET INSUFFICIENT DATA FOR A MEANINGFUL ANSWER."
Man's last mind fused and only AC existed – and that in hyperspace."
– Isaac Asimov, The last question

Para realizar cualquier entrenamiento, validación y testeo en la red se necesitan datos. En principio no se sabe cuantos, pero cuanto mayor sea el conjunto con el que se entrena y valida la red, mejores resultados se espera que provea. La series temporales que pueden servir y están disponibles son escasas, muy diferentes en características y, en realidad, no se sabe con certeza cuantos planetas pueden contener.

La opción es generar series temporales artificiales. La posibilidad de controlar cada uno de los componentes del ruido, tiempos de observación, cantidad de planetas y demás aspectos, dieron muchas opciones al momento de elegir que tipo de datos usaría la red y permitieron probar diferentes modelos para atacar al problema.

Para la generación de datos se utilizó **Python 3.7** (prácticamente ya un estándar *de facto* en el área) y para la implementación de la red neuronal se le sumó la biblioteca **Keras** con el motor **Tensorflow**.

No se requirió gran poder de cómputo para las etapas de entrenamiento y testeo; pero, la generación de los datos resultó sumamente costosa¹ y se utilizó el cluster **NEURUS** del *Departamento de Computación de Alta Prestación, Gerencia de Tecnología de la Información* de la *Comisión Nacional de Energía Atómica*.

2.1. Generación de datos

Se detallará el proceso de generación del conjunto de datos, el mismo consta de dos grandes partes:

1. **Generación de series temporales:** Se simulan varios tipos de ruido presentes en estrellas de tipo solar y se le agregan las perturbaciones planetarias.
2. **Generación de periodogramas:** Una vez que se tiene la serie temporal se genera el periodograma y se calcula el FAP del pico más alto.

¹ A modo de ejemplo, generar 600 periodogramas (la entrada de la red) lleva más de tres días de cómputo en un nodo de 48 núcleos y 192 GB de memoria.

Todos los parámetros usados y datos generados se guardan en un archivo (características particulares de cada planeta, semilla utilizada para generar parámetros aleatorios, ruidos, errores, etc.) y además no solo se genera y almacena el periodograma correspondiente a la serie temporal, también se calcula y almacena el FAP del pico máximo y de cuatro pasos completos del *astrónomo virtual*, independientemente de si su valor indica que la periodicidad es significativa o no.

Esto es por una cuestión meramente práctica. Al tener ya calculados y almacenados los periodogramas y los FAP es mucho más sencillo y rápido simular una iteración del *astrónomo virtual*. Simplemente se itera la lista de periodogramas sucesivos hasta que el valor del FAP dictamine que el pico no es planeta y, como están guardados todos los valores de las señales planetarias, se puede saber exactamente los aciertos y errores de la ejecución, variar el límite de detección del FAP y ver como influye en los resultados, etc.

Como ventaja final, al hacer esto, se multiplica por cuatro la cantidad de periodogramas disponibles para entrenar la red ya que cada uno se pueden utilizar de manera independiente².

2.1.1. Series temporales

Se busca que las series temporales sean lo mas realistas posibles y que exhiban la variabilidad observada en las estrellas de tipo solar, por lo que se generaron diferentes tipos de ruido y perturbaciones planetarias que luego se muestrean en doscientos puntos. Estos puntos donde se toman las muestras simulan las mediciones tomadas en las noches de observación astronómica, por lo que no son a un intervalo regular (no todas las noches se observa a la misma hora).

Los tiempos en los que se realizan las mediciones se obtienen de valores equiespaciados a los que se les suma una componente normal para darles cierta variabilidad. En la implementación esto se consigue mediante un vector de *tiempos* con los valores obtenidos de esa distribución, para luego aplicarlo en la señal generada y obtener sus valores en esos momentos.

Los diferentes componentes de esta señal son:

- Errores intrínsecos.
- Ruido blanco.
- Ruido correlacionado.
- Planetas.

² Como se verá más adelante, los periodogramas serán una de las entradas de la red neuronal.

Errores y ruido blanco

Todas las mediciones tienen errores (pequeñas fallas del equipo, interferencias de la atmósfera, etc.) para simularlos, se investigó la estadística de los errores del relevamiento HARPS [UDL⁺19], [DSU⁺16] que cuentan con más de cuarenta mediciones. Por otro lado, mirando las estrellas características HD40307, HD1461 y HD204313, se encontró que la dispersión de los errores es de 0.24 ms^{-1} , 0.37 ms^{-1} y 0.13 ms^{-1} , respectivamente. Es decir, una dispersión bastante pequeña por estrella.

Por lo que se tomó el valor medio de los errores de una estrella al azar del programa de alta precisión (PI: M. Mayor, luego S. Udry, luego R. Díaz), y se obtuvieron doscientos números aleatorios de una distribución normal con media igual a ese valor y varianza igual a $(0.30 \text{ ms}^{-1})^2$, si algún valor era negativo se lo reemplazó por la mitad de la media. El vector obtenido constituirá el valor del error.

Para simular ruido blanco³ en la señal simplemente se toma el vector de errores, calculado anteriormente, y se multiplica cada valor por un número aleatorio proveniente de una distribución normal estándar.

Ruido rojo

Hay varias componentes de ruido estelar que pueden alterar las mediciones de velocidad radial, en particular el ruido rojo o browniano puede generar una señal con una cierta periodicidad que, durante el análisis, puede inducir a pensar que hay un planeta.

Algunos de los fenómenos estelares más importantes que pueden generar este tipo de ruido son:

1. **Pulsaciones:** Las capas exteriores de la estrella se contraen y expanden mientras la estrella busca mantener su equilibrio (la enorme gravedad de la misma *tira* para adentro y la energía liberada de la fusión nuclear *empuja* para afuera). Cuando estas fluctuaciones en el radio estelar son vistos desde nuestra perspectiva, muestran que la estrella se está acercando y alejando y, al haber cierta regularidad en este movimiento, se puede crear la ilusión que es por la influencia de un planeta.
2. **Granulación:** Diversos movimientos convectivos en la fotosfera, la superficie luminosa de la estrella, provocan este fenómeno. Estos “gránulos” que emergen del interior de la estrella a la fotosfera están más calientes que los que se enfriaron y

³ El ruido blanco es una señal aleatoria que se caracteriza por el hecho de que sus valores de señal en dos tiempos diferentes no guardan correlación estadística. Como consecuencia de ello, su densidad espectral de potencia es una constante, esto significa que la señal contiene todas las frecuencias y todas ellas muestran la misma potencia. Igual fenómeno ocurre con la luz blanca, de ahí su denominación.

descienden, esto produce un corrimiento espúreo al rojo que cambia a medida que se modifique el patrón de convección.

3. **Modulación rotacional:** La rotación de la estrella puede transportar diversas estructuras en la superficie, haciendo que las mismas aparezcan y desaparezcan a un intervalo regular creando la ilusión de un “bamboleo” estelar.

Por ejemplo: si pudiésemos mirar en detalle la luz de la estrella veríamos que, por la rotación, una mitad está corrida al rojo y la otra al azul; al aparecer una mancha en el lado azul, el promedio la luz indicaría que hay más rojo, por lo que se pensaría que la estrella se aleja. Cuando la mancha pase al lado rojo ocurriría lo opuesto, y aparentaría acercarse. Esta mancha va desapareciendo con el tiempo y este aparente corrimiento sería cada vez menor, pero estas mediciones tienen una correlación.

Para simular las **pulsaciones** y las **granulaciones** se utilizó como base a [DUL⁺11], en ese trabajo se estudia el espectro de potencias de cinco estrellas solares muy bien observadas (β Hyi, μ Ara, α Cen A, τ Ceti, α Cen B), se ajusta el espectro con una componente Lorentziana que representa las pulsaciones y tres componentes para la granulación, normal, meso y súper. Usando los valores de los parámetros definidos en el trabajo se construye el espectro de potencias como suma de estos cuatro componentes y se genera la velocidad radial.

Para la **modulación rotacional** se usó un proceso gaussiano con una función de covarianza generada con un *kernel* pseudoperiódico calculado con:

$$k_{QP}(t_i, t_j) = A^2 \exp \left(-\frac{(t_i - t_j)^2}{2\tau^2} - \frac{2}{\epsilon} \sin^2 \left(\frac{\pi(t_i - t_j)}{\mathcal{P}} \right) \right)$$

Este *kernel* tiene 4 hiperparámetros:

- **A:** Amplitud de la covarianza.
- **\mathcal{P} :** Tiempo de recurrencia, similar a un período de rotación
- **τ :** Tiempo de decaimiento, se puede asociar con el tiempo de vida medio de una región activa.
- **ϵ :** Factor de estructura, es un parámetro asociado con la cantidad de regiones activas que se presentan simultáneamente.

La elección de estos hiperparámetros se realiza con la ayuda de la estadística del relevo HARPS mencionado anteriormente, todas las estrellas cuentan con una estimación de su período de rotación obtenida a partir de una medición de su actividad magnética usando la calibración de Mamajek [MH08].

Con esta información la elección de los parámetros fue la siguiente:

- **A**: Al azar de una distribución gamma, $\Gamma(2.0, 0.5)$.
- **P**: Se elige al azar un período de rotación de HARPS que sea mayor o igual a un día.
- **τ** : Obtenido al azar de una distribución normal, $\mathcal{N}(3 * \mathcal{P}, 0.1 * \mathcal{P})$.
- **ϵ** : Obtenido al azar de una distribución uniforme, $\mathcal{U}(0.5, 1.0)$.

Sumando todas estas contribuciones, se llega a la serie temporal de velocidades radiales sin planeta rv_{sp} que se muestra en la figura 2.1.

Planetas

En este modelo simplificado los planetas van a tener órbitas circulares, no se van a pisar en período y no van a interactuar entre sí.

Para simular estas alteraciones se necesitan determinar tres parámetros:

- Período de la variación: P_{pl}
- Amplitud de la variación: K
- Tiempo en el que ocurre el mínimo de la variación: T_0

La elección de estos valores fue de acuerdo a los siguientes criterios:

- P_{pl} : Elegir un número al azar entre 10 y 100. Más generalmente, entre $10 * \delta$ y $\frac{\Delta}{2}$. Donde δ es la mínima distancia entre dos puntos de la serie temporal, y Δ es la duración total de la serie.
- K : Elegir un número distribuido uniformemente en log entre 0.1ms^{-1} y 10ms^{-1} .
- T_0 : Al azar dentro de la serie temporal.

La variación toma entonces la forma:

$$rv_{pl} = K \sin\left(\frac{2\pi(t - T_0)}{P_{pl}}\right)$$

Una vez obtenido el vector rv_{pl} , ejemplificado en la figura 2.2, puede sumarse directamente a rv_{sp} para obtener la serie temporal con planeta de la figura 2.3, ya antes mostrada.

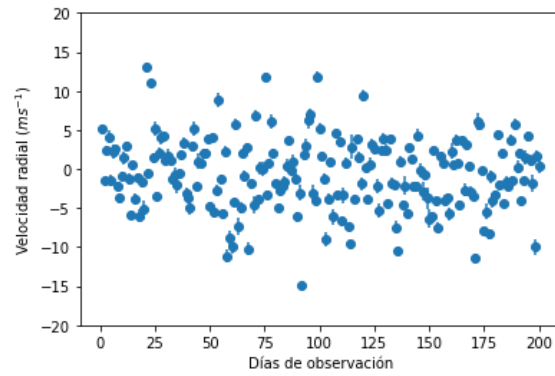


Fig. 2.1: Ruido estelar con sus barras de error.

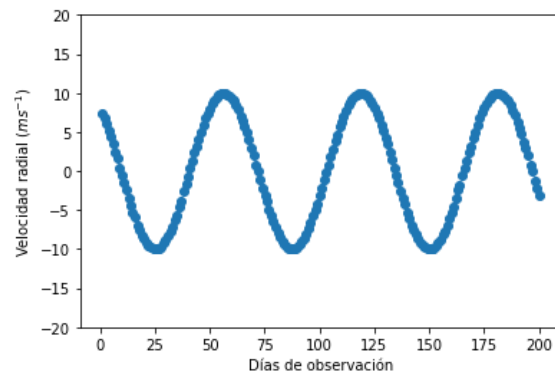


Fig. 2.2: Una señal planetaria con período de 62 días y una amplitud de 10ms^{-1} .

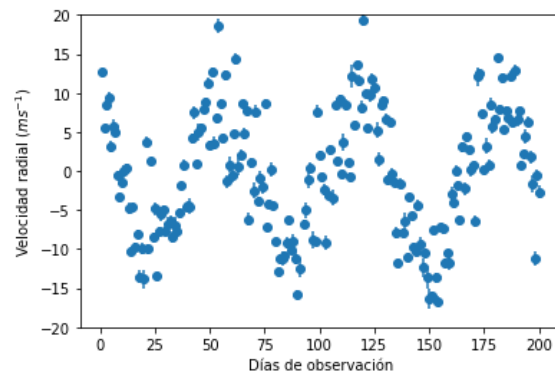


Fig. 2.3: Serie temporal resultante de sumar el ruido estelar a la señal planetaria.

Para agregar más de un planeta se repite el procedimiento y se suman las contribuciones, suponiendo que el modelo es lineal en los efectos de los planetas, es decir, se desprecia la interacción entre planetas.

2.1.2. Periodogramas

Se usa una implementación propia de [ZK09], que toma cuatro argumentos:

- **t**: El vector de tiempos calculado anteriormente.
- **rv**: El vector de las velocidades radiales.
- **erv**: El vector de errores.
- **prange**: Vector con los períodos donde se quiere evaluar el periodograma. Este parámetro es optativo, si no se lo proporciona se calcula un arreglo automáticamente; pero, se optó por tener todos los periodograma evaluados en los mismos valores, así que se usó el mismo para todos.

Para asegurarse que no se pierde ningún pico importante hay que *samplear* las frecuencias con un paso máximo dnu igual a:

$$dnu = \frac{1}{\max(t) - \min(t)}$$

Los límites del rango de frecuencias evaluado (pnu) vienen dados; por un lado, por la duración de las observaciones, y por el otro por la frecuencia de Nyquist.

Aquí, como el muestreo es de aproximadamente $día^{-1}$, la frecuencia de Nyquist es de $0.5 día^{-1}$. Entonces el vector pnu se genera con:

$$pnu = [dnu, dnu + \left(\frac{dnu}{10}\right) * 1, dnu + \left(\frac{dnu}{10}\right) * 2, \dots, 0.5]$$

Este el vector de frecuencias a evaluar, y su inversa, es el vector de períodos $prange$ con el que finalmente se genera el periodograma.

$$prange = \frac{1}{pnu}$$

La salida de la función que calcula el periodograma es un vector con la potencia de cada uno de los períodos evaluados.

2.1.3. FAP

El cálculo del FAP es una parte crucial del análisis, por lo que se implementaron ambas variantes, la analítica y el Monte carlo. Al momento de calcularlo primero se hace con la fórmula de Baluev, si el resultado es mayor o igual que $\log_{10}(0.01)$ ⁴ se considera como un resultado dudoso y se vuelve a calcular utilizando el Monte carlo.

⁴ O lo que es lo mismo, -2.

2.2. Red neuronal

Al momento de encarar este trabajo, una de las cosas más importantes fue decidir el tipo de preguntas que se querían hacer a la red y, en base a esto, evaluar que tipo de la misma se podía utilizar. El problema es bastante amplio y, al haber generado cada componente de los datos de entrada, se cuenta con una gran variedad de opciones al momento del modelado⁵. La gran ventaja de la solución que se va a proponer es que permite una comparación directa con el método tradicional, y se trata de ver si la red es capaz de distinguir variabilidad estelar planetaria, o dicho de otra manera, si puede reemplazar al cálculo del FAP.

Entonces, la pregunta que se le va a hacer a esta red es:

“Dado este periodograma y este pico máximo, ¿es un planeta?”.

El problema es un problema de decisión, dado que la entrada va a ser un periodograma (que está representado en un arreglo unidimensional de datos, con relación entre sí y características locales y generales) la solución más natural es usar redes convolucionales en una dimensión.

Va a recibir, entonces, dos entradas:

- El **periodograma**, que consta de un arreglo de 990 posiciones.
- La **posición** y el **valor** del pico máximo.

Siguiendo los lineamientos de la bibliografía, las capas profundas serán convolucionales intercaladas con capas de *maxpooling*, seguidas de capas densas (intercaladas a su vez con *dropout*) y finalizando con una sigmoide para clasificación que devuelve un valor real entre 0 y 1.

En un problema de clasificación es muy importante considerar el balance de las clases durante el entrenamiento. Si se entrena una red con muchos más casos de una una clase C cuando le llegue el momento de predecir con datos nuevos estará mucho más inclinada hacia esa clase que hacia las demás⁶, teniendo esto presente se generaron tres conjuntos de datos: uno para entrenamiento y validación de la red, otro para ajuste de parámetros y comparación entre métodos, y uno final para pruebas del método completo.

⁵ En el apéndice B se incluyeron algunas de las diferentes opciones consideradas, unas con más o menos éxito y otras con grandes posibilidades de ser exploradas en trabajos futuros.

⁶ Supongamos un caso extremo en el que se tiene una red para decidir si una foto pertenece a un perro o un gato, y se la entrenó con un 98 % de fotos de gatos. Ante nuevas entradas la red probablemente siempre prediga que la imagen ingresada es la de un gato, ya que aprendió que simplemente haciendo eso va a acertar un 98 % de las veces.

Para el segundo y tercer conjunto se generaron igual cantidad de estrellas con 0, 1, 2, 3 y 4 planetas y se utilizaron todos sus periodogramas. Por la propia naturaleza de la generación, estos conjuntos van a estar muy desbalanceados hacia los casos negativos⁷, lo cual no se consideró un problema porque se ven como casos más reales de aplicación. Para el conjunto de entrenamiento, en cambio, se buscó balancear un poco los datos. Este conjunto consta de 13700 periodogramas y se toma aleatoriamente el 20 % para validación. El detalle de estos conjuntos se puede ver en la tabla 2.1.

Para evaluar el rendimiento de la red durante el entrenamiento y validación se pueden usar varias métricas, la más común es la de *accuracy* que es simplemente el porcentaje de casos acertados entre los casos totales, pero, como el conjunto sigue presentando un ligero desbalance no es del todo recomendable usarla⁸.

Hay otras métricas como las *F-measure* que al momento de evaluar el rendimiento sumarizan conceptos como precisión y exhaustividad (definidos en detalle en 3.1) más apropiadas para el problemas que se busca resolver. Para más detalle sobre este tema se puede ver [Bro20].

Se probaron diferentes configuraciones de la red hasta que se llegó a la versión definitiva de la figura 2.4 que logró un $F\beta$ -measure (con $\beta = 0.5$) de **0.86** en el conjunto de validación.

Conjunto 1 - 3425 estrellas.
40 % de casos positivos. Se utilizó para entrenamiento y validación de la red.
Conjunto 2 - 2500 estrellas.
Cantidades iguales de estrellas con 0, 1, 2, 3 y 4 planetas. 28 % de casos positivos. Se utilizó para comparaciones entre métodos y búsqueda de parámetros.
Conjunto 3 - 5000 estrellas.
Cantidades iguales de estrellas de 0, 1, 2, 3 y 4 planetas. 28 % de casos positivos. Se utilizó para aplicar el método completo a cada estrella.

Tab. 2.1: Conjuntos de datos generados

⁷ Lo cual es bastante claro si se piensa que de cada sistema se usan los cuatro periodogramas, por lo que por cada estrella de 0 planetas se tienen 4 periodogramas con etiquetas negativas, y de las otras no siempre se tiene un planeta en el pico máximo.

⁸ Siguiendo con el ejemplo de los perros y gatos, el *accuracy* de ese modelo que solo predice la clase dominante será de un 98 % cuando, claramente, no es una buena solución.

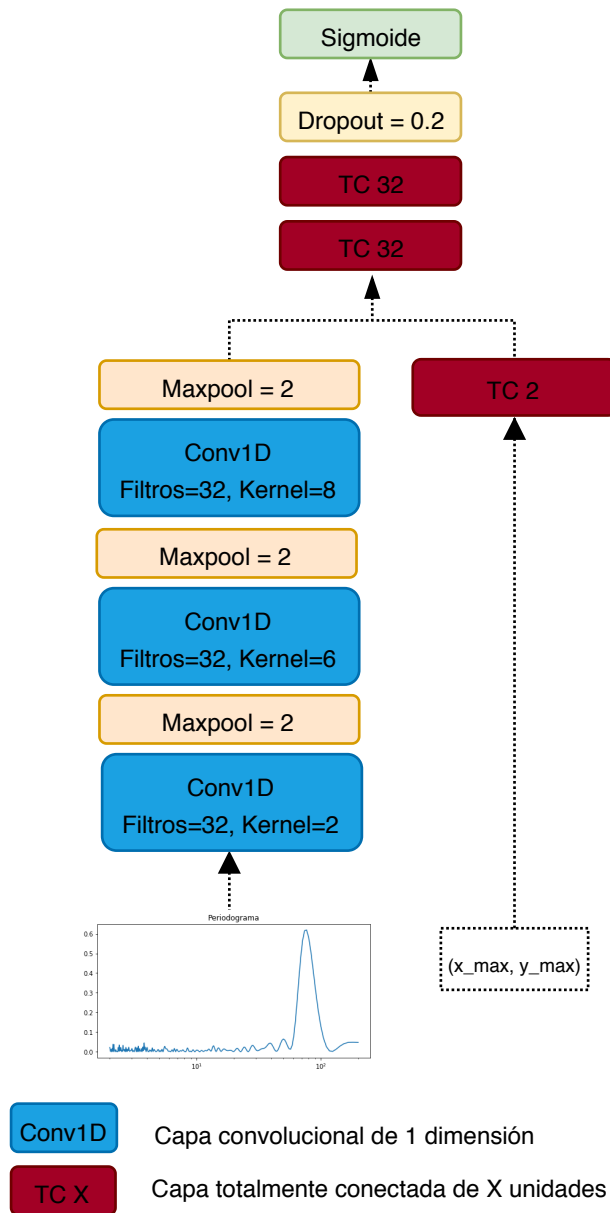


Fig. 2.4: Red para clasificación de picos.

3. RESULTADOS Y ANÁLISIS

“A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it.”
– Frank Herbert, Dune

Es este capítulo, se presentan análisis y comparaciones entre el método de detección de planetas usando el FAP y la red neuronal ya entrenada. Se busca un umbral apropiado para la detección; se analizan los planetas encontrados buscando caracterizar las capacidades de detección de cada método y, finalmente, se compara el rendimiento general al utilizar la red y el método tradicional en el *astrónomo virtual*.

3.1. Calidad de las predicciones

Cuando el FAP o la red analizan un pico, devuelven un número. En el primer caso es un valor, expresado en \log_{10} , que representa la probabilidad de falsa alarma del pico en cuestión; y en el segundo, es un valor entre 0 y 1 que expresa que cuanto más cerca de 1 sea debe ser más considerado como planeta que si estuviese cerca del 0. Ambas implementaciones llevan a tener que definir un umbral de corte cuando se las quiere usar para tomar decisiones.

Como se muestra en la figura 3.1 la elección de este umbral afecta directamente los resultados que pueden obtener; si es muy estricto, se tendrán detecciones positivas más confiables, pero se perderán casos; y si es muy laxo, se detectarán más planetas, pero se incorporarán más falsos positivos. Es un fino balance entre lo que se quiere ganar y lo que se está dispuesto a perder.

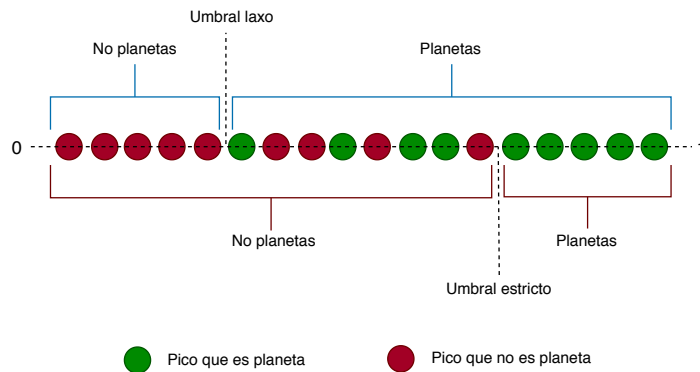


Fig. 3.1: Calidad de resultados al variar el umbral

Antes de fijar este umbral, se puede hacer un análisis de efectividad de cada método y compararlos entre sí, para ello se utilizaron curvas de *precision-recall* (precisión y exhaustividad). Estas métricas se definen como:

- **Precisión:** Es la razón de **verdaderos** casos positivos entre los que fueron **marcados** como positivos. En nuestro caso sería:

De todos los que fueron identificados como planetas, ¿cuál es la fracción de planetas verdaderos?

$$\text{Precisión} = \frac{\{\text{Planetas reales}\} \cap \{\text{Marcado como planeta}\}}{\{\text{Marcado como planeta}\}}$$

- **Exhaustividad:** Es el ratio de **verdaderos** casos positivos detectados entre los positivos **existentes**. Nuevamente, en este problema sería:

De todos los planetas que hay en la muestra, ¿qué fracción se encontró?

$$\text{Exhaustividad} = \frac{\{\text{Planetas reales}\} \cap \{\text{Marcado como planeta}\}}{\{\text{Planetas reales}\}}$$

Ambos resultados varían entre 0 y 1 y son mejores cuanto más altos son. La intuición detrás de la precisión es que busca minimizar los falsos positivos y la exhaustividad lo hace con los falsos negativos.

Conviene observar ambos resultados al mismo tiempo para evaluar el modelo, ya que, si tiene alta exhaustividad pero baja precisión, se van a encontrar muchos planetas, pero con muchos falsos positivos; y si tiene alta precisión y poca exhaustividad, es lo opuesto, pocas detecciones, pero de buena calidad. Un escenario ideal sería el que tiene ambos valores altos, esto es, muchas detecciones de verdaderos planetas.

La curva de precisión-exhaustividad muestra el compromiso entre estos valores. Se varia el umbral y se grafican los diferentes resultados de estas métricas y, cuanto mayor es el área debajo de esta curva, mayor es la exhaustividad y precisión general de la solución. Esto no solo nos permite hablar de la calidad de un modelo, también es útil para comparar dos soluciones diferentes al mismo problema.

Usando el segundo conjunto de datos¹ se realizó la curva de precisión-exhaustividad para el FAP y la red de la figura 3.2 y se calcularon sus AUC (*Area Under the Curve*).

¹ Que recordemos, son datos que la red nunca vió.

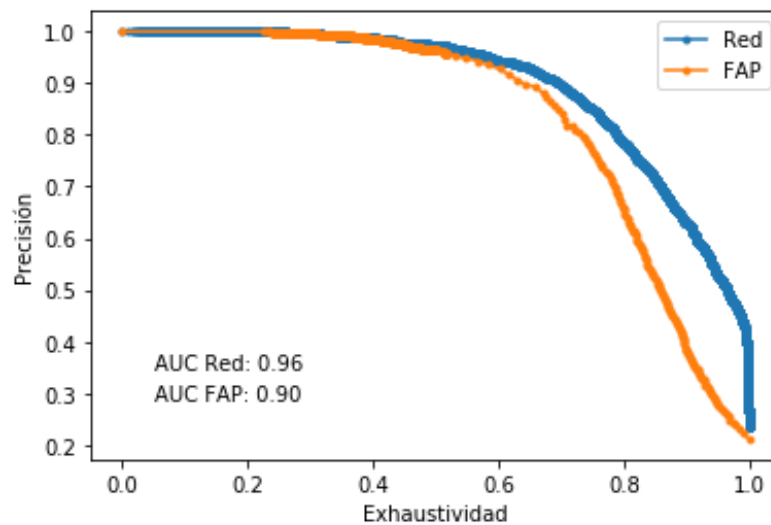


Fig. 3.2: Precisión-Exhaustividad y AUC de ambos métodos

La curva y las áreas muestran que la red no solo funciona sistemáticamente mejor que el método tradicional, sino que su área es mucho mayor. Esto ya muestra que la red es una mejor solución general al problema.

En la figura 3.3. se muestra un análisis más complejo: sobre el eje x se tiene la probabilidad que le asignan los métodos a los picos, y sobre el eje y se tienen la cantidad de los mismos que fueron marcados con esa probabilidad (separados por los que son planetas y los que son ruido). Entonces se puede ver, por ejemplo, que cuando ambos métodos asignan a los picos una probabilidad muy cercana a 1 es mucho mayor la cantidad de picos que son verdaderos planetas que los que son ruido (o el caso inverso si se acercan a 0). Lo cual está muy bien, pero, es al comparar ambas implementaciones cuando se ve como la red tiene una ventaja; ya asignando probabilidad bajas como 0.4, siempre tiene más casos de buenas detecciones que de malas; contrario al FAP, que recién lo logra para cuando asigna probabilidades mayores a 0.8. Concordantemente con la curva anterior, la red es más precisa en sus predicciones.

Ya una vez analizado el rendimiento general se puede volver al tema de la búsqueda de los umbrales. Para ello primero se analizó el caso del FAP.

En la introducción ya se mencionó que cuando se utiliza este método, se suele usar con un umbral de -1.3, que es un 0.05% de probabilidad de que el pico sea un falsa alarma. Es bastante conservador, pero, se busca estar lo más seguro posible que el pico es un planeta. Usando esto en la curva de precisión-exhaustividad se podría buscar que valor de umbral de la red da un mejor resultado en la precisión, pero se prefirió mostrarlo en una curva separada, por considerarlo más claro.

Simplemente se fijó este umbral en el FAP y se calcularon cuantos falsos positivos

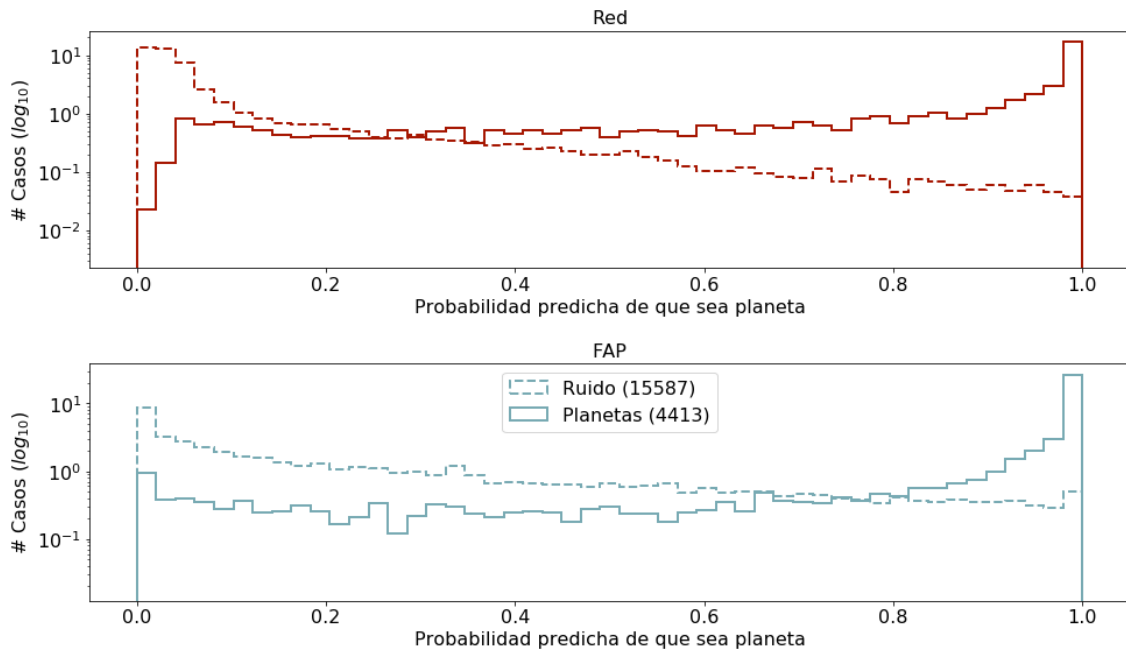
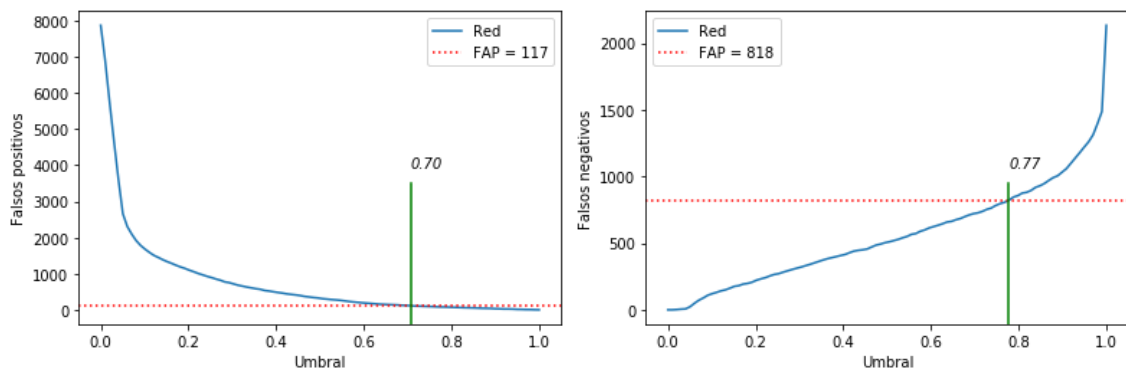


Fig. 3.3: Cantidad de falsas detecciones de ruido como planeta y de planetas verdaderos en función de la probabilidad asignada por la red y FAP.

y falsos negativos se obtienen, siendo estos valores 117 y 818 respectivamente. Como se pretende que la red de la menor cantidad posible de falsos positivos (sin aumentar los falsos negativos) se buscaron los valores de umbral en los que la red tenga un mejor desempeño que el método tradicional.

En la figura 3.4. se muestran los errores al ir variando el umbral en la red, y se marcaron los límites dentro de los que se puede mover este valor para obtener mejores resultados que los del FAP.



(a) Falsos positivos en función del umbral.

(b) Falsos negativos en función del umbral.

Fig. 3.4: Límite inferior y superior del umbral donde la red tiene menos falsos positivos y no aumentan los falsos negativos.

Cualquier valor de corte mayor a 0.70 dará menos falsos positivos, y para los menores a 0.77 dará menos falsos negativos. Se fijó, entonces, el umbral de la red en **0.77**.

Una vez fijados los umbrales de ambos métodos se analizó el efecto de aplicarlos en los datos del tercer conjunto², el mismo tenía unas 5000 estrellas (y por ende 20 000 periodogramas).

Lo primero que se hizo fueron las *matrices de confusión* de las tablas 3.1 y 3.2, que son matrices donde se ven cuantificados los resultados generales de las detecciones; las filas muestran las verdaderas cantidades de casos positivos y negativos en el conjunto; y las columnas las predicciones de cada método.

		Predicción	
		0	1
Valor real	0	15309	277
	1	1672	2742

Tab. 3.1: Matriz de confusión FAP

		Predicción	
		0	1
Valor real	0	15388	198
	1	1674	2740

Tab. 3.2: Matriz de confusión Red

El ajuste de la red dio los resultados buscados, se disminuyó en un **28.5%** la cantidad de falsos positivos, comparado con el método tradicional, y con apenas dos falsos negativos más. Esto se ve claramente reflejado en la comparación de la tabla 3.3 donde la exhaustividad se mantiene prácticamente igual, pero la precisión aumenta.

	FAP	Red
Precisión	0.908	0.932
Exhaustividad	0.621	0.620

Tab. 3.3: Precisión-Exhaustividad FAP y Red

3.2. Tiempo de ejecución

Se mostró en la introducción que, dependiendo del resultado, el cálculo del FAP puede ser por la fórmula de Baluev o mediante un algoritmo tipo Monte Carlo. Hacer la cuenta de Baluev no es muy costoso computacionalmente hablando, pero, cuando se ejecuta el Monte Carlo se tienen que generar 150 velocidades radiales y sus correspondientes periodogramas³. Todo esto lleva a que el tiempo promedio (entre ejecuciones de Baluev y Monte Carlo) necesario para calcular un valor FAP sea bastante alto.

² Al igual que en el caso anterior, estos son datos completamente nuevos para la red.

³ Generando el ruido y todo lo que se vió en el capítulo anterior, salvo la interferencia planetaria.

Para comparar los tiempos de ejecución entre los métodos primero hay que introducir las definiciones de *Wall-Time* y *Proc-Time*:

- **Proc-Time:** Es el tiempo usado exclusivamente para ejecutar un proceso, no tiene en cuenta cosas como acceso a disco u otros dispositivos de entrada/salida, interrupciones del sistema operativo, tiempo usado en otros procesos, etc.
- **Wall-Time:** Es el tiempo de reloj que tarda en ejecutarse algo, desde que se inició hasta que finalizó. A diferencia del anterior es simplemente un cronómetro, por lo que si el proceso está esperando para leer un dato o el sistema está muy ocupado con varios procesos sumará a este tiempo.

Qué tiempo mirar depende de lo que se esté haciendo o buscando optimizar, si se está escribiendo código para ejecución en paralelo y se quiere ver que tan bien se está distribuyendo la carga entre los procesos se podría ver ambos tiempos y compararlos. Un ideal sería que el *wall-time* sea igual a $proc-time/\#procesos$ ⁴, si se está muy lejos de este valor puede ser un indicador que se puede mejorar un poco más.

Por otro lado, si se quiere evaluar el tiempo de pequeños programas o porciones de código suele convenir ver solamente el *proc-time*.

El código para la generación de los periodogramas no es paralelo (por lo menos no en su implementación), por lo que estos dos tiempo deberían dar muy similares.

En el caso de la red es bastante diferente, una red entrenada no es otra cosa que una gran matriz con números y, hacer una predicción, es simplemente tomar la entrada y ejecutar operaciones básicas como sumas y multiplicaciones para devolver un valor. Ya por este simple hecho la red debería ser muchísimo más rápida⁵.

Para medir los tiempos de cada método se ejecutaron varias veces midiendo tiempos y se promediaron los resultados. En el caso del FAP fueron 100 ejecuciones completas (recordando que si el resultado de Baluev es lo suficientemente bueno se prescinde del Monte Carlo). Y para la red se hicieron predicciones sobre 10^6 periodogramas⁶, los resultados están reflejados en la tabla 3.4 y están expresados en la cantidad de segundos que se tarda en analizar un solo periodograma.

⁴ En un caso ideal, sin ningún *overhead* y con un código 100% paralelo, si se tiene un código que tarda en ejecutar, por ejemplo, 3 segundos y se distribuye en cuatro núcleos (con un proceso por núcleo), el *wall-time* sería 3, pero el *proc-time* suma el tiempo que se usó en cada proceso, por lo que resultaría en 12 segundos.

⁵ Solo para mencionar otra ventaja de la red, ocupa en memoria apenas 1.7kB.

⁶ Las ejecuciones fueron en una computadora de escritorio con un micro Intel Core I5 de 2,3 GHz con 8Gb de RAM.

	FAP	Red
<i>Wall-Time</i>	425.97 s x periodograma	0.0011 s x periodograma
<i>Proc-Time</i>	427.69 s x periodograma	0.0034 s x periodograma

Tab. 3.4: Tiempos de ejecución FAP y Red

La diferencia es enorme, centrándonos solamente en el *proc-time*, la red es 125 791 veces más rápida, esto es, una diferencia de 5 órdenes de magnitud.

Claramente el código para calcular el FAP no está optimizado, la generación de las diferentes formas de ruido y los periodogramas está afectando severamente el rendimiento. Se podría tratar de optimizar alguna de estas partes, e incluso paralelizar el Monte Carlo, pero la realidad es que la ventaja que le lleva la red es demasiado grande como para que cualquier optimización posible la pueda siquiera acercar a estos resultados.

3.3. Características de las detecciones

Luego de haber analizado el comportamiento general del FAP y la red, es interesante analizar las particularidades de las detecciones a fin de caracterizar y entender qué tipos de planetas encuentra cada método y como se comportan.

En la figura 3.5. se grafica la precisión y exhaustividad en función de los períodos de los planetas encontrados. Como es esperable, planetas con períodos más grandes son más fáciles de detectar para ambos métodos, lo que se ve reflejado en la curva de precisión.

En ambos gráficos se ve un comportamiento particular entre los períodos 30 y 60 días; nuestra interpretación es que, al ser períodos propios de la rotación estelar, la red está siendo más “cauta” al momento de etiquetar un pico de esa zona (que es lo mismo que haría un astrónomo). Entonces en esa área, el FAP, está etiquetando más picos de ruido como planetas y por eso pierde precisión. Mismo razonamiento para el gráfico de exhaustividad, el FAP se mantiene mas o menos constante a lo largo de los períodos, pero la red no se arriesgó tanto, por lo que se perdió de encontrar más planetas.

En la figura 3.6. se grafican la precisión y la exhaustividad en función de la potencia que tuvo el pico en el periodograma, si bien ambos son bastante similares se puede ver claramente que, para picos de baja potencia, la red es mucho más precisa. Esto lleva a pensar que, si bien la potencia indica una mayor probabilidad de que el pico sea un planeta, la red ve otras características generales del periodograma que le permite identificar mejor los picos planetarios. Claramente es en esta zona donde la red disminuye las falsas detecciones (comparado con el FAP) y explica los resultados de la sección anterior.

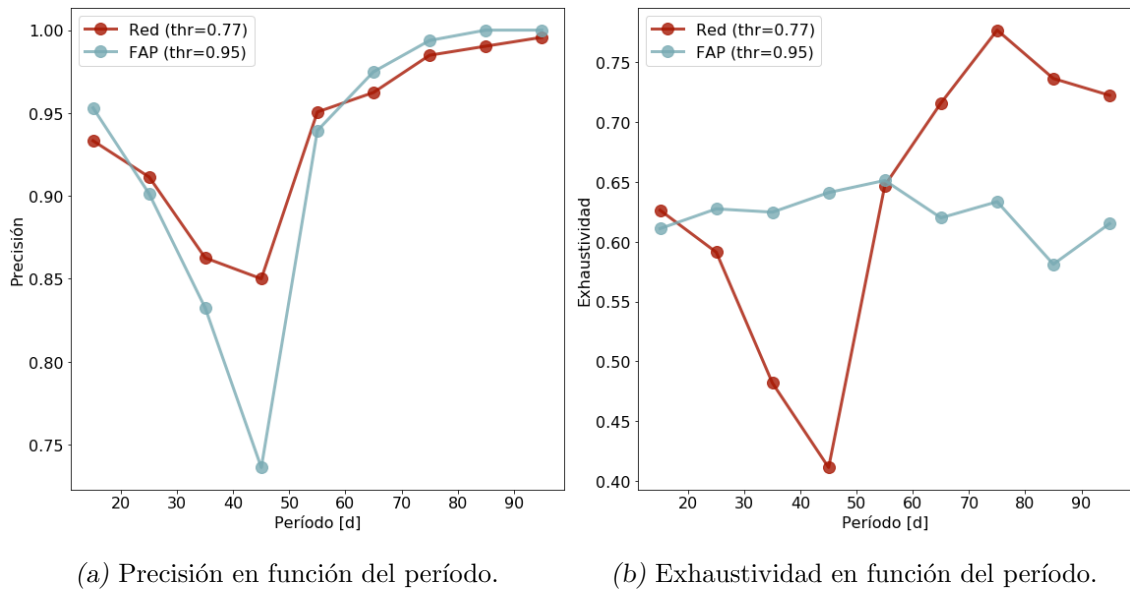


Fig. 3.5: Precisión y exhaustividad en función del período de los planetas detectados, usando el umbral (thr) 0.77 para la red y 0.95 para el FAP.

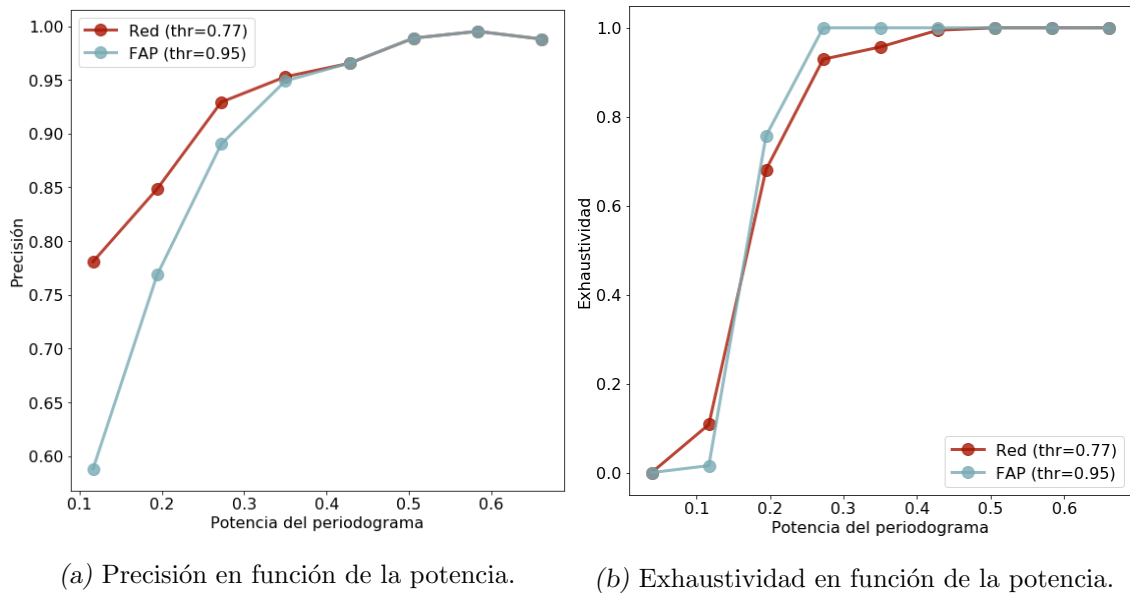


Fig. 3.6: Precisión y exhaustividad en función de la potencia en el periodograma del planeta detectado, usando el umbral (thr) 0.77 para la red y 0.95 para el FAP.

3.4. Método completo

Hasta ahora, todas las pruebas fueron con los periodogramas de los conjuntos, pero no se estudió como se comportan los métodos cuando se usan en el *astrónomo virtual*.

Se realizó, entonces, un último análisis en el que se fijaron los umbrales a los ya mencionados y se aplicaron ambas implementaciones del *astrónomo virtual* a cada estrella de las 5000 del tercer conjunto. Se busca ver efectivamente cuantos planetas son capaces de encontrar en una ejecución completa y, sobre todo, cómo se comportan con los falsos positivos.

Si bien del análisis anterior se desprende que el FAP funciona peor en lo que se refiere a los falsos positivos, no está tan claro como esto puede afectar al rendimiento general del método completo, ya que, por un lado, si identifica más picos de ruido como si fuesen planetas esto significa que el *astrónomo virtual* con el FAP va a continuar su ejecución más veces, y aumentan las chances de que en esas nuevas iteraciones detecte un planeta al que el método con la red no pudo llegar (por haber sido mejor al identificar el pico); y por el otro, esto podría significar muchos más falsos positivos.

Para ver mejor qué ocurre en cada caso, se separaron las estrellas por cantidad de planetas (hay 1000 de cada categoría) y se contabilizan los dos tipos de errores para ambas implementaciones del método completo. En las figuras 3.7. y 3.8. se muestran los falsos positivos y falsos negativos de las ejecuciones.

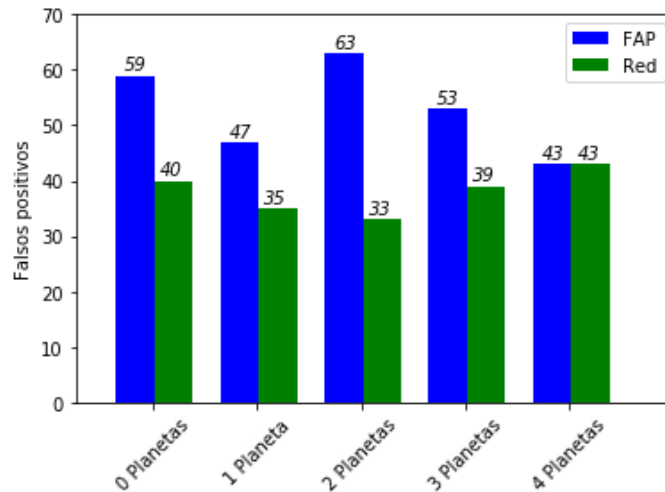


Fig. 3.7: Falsos positivos del *astrónomo virtual* con ambas implementaciones.

Concordantemente con el análisis general anterior, el número de falsos positivos del método usando la red se mantiene por debajo del obtenido por el FAP y variando muy poco entre los diferentes tipos de estrellas, no así el método con FAP que varía el error de

forma más marcada, aunque los errores se igualan en las estrellas con cuatro planetas.

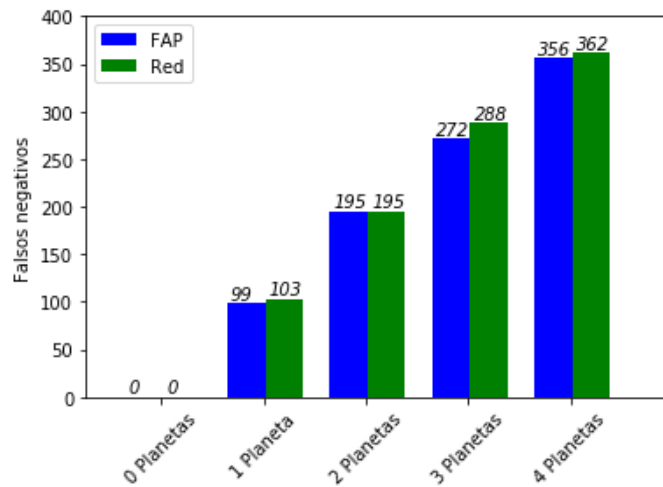


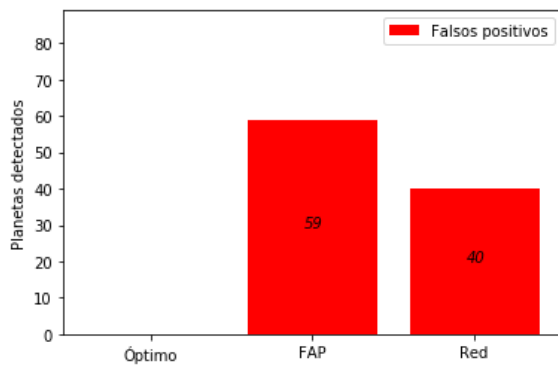
Fig. 3.8: Falsos negativos del *astrónomo virtual* con ambas implementaciones.

Los falsos negativos en ambas implementaciones son muy similares, aunque, en tres de los cuatro casos que importan, el método con la red tiene algunos más. Probablemente se pueda ajustar el umbral pero, viendo los resultados del gráfico anterior, afectaría los falsos positivos en las estrellas con cuatro planetas y seguramente daría peor que el método con la implementación tradicional.

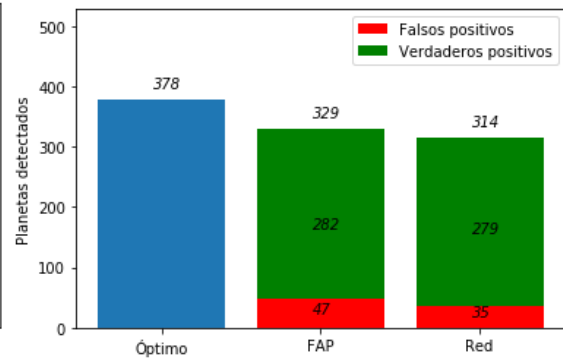
Para ver el real alcance de las implementaciones en la detección de todos los planetas de las series temporales, se comparó contra el óptimo posible, esto es, el máximo número de planetas que podría encontrar el *astrónomo virtual* si tuviese un mecanismo de decisión perfecto ⁷. Se muestran todos estos resultados en los histogramas de la figura 3.9.

Si bien los resultados son prácticamente iguales en cantidad de planetas, claramente el haber continuado (cuando no debería) le dio una ventaja extra al *astrónomo virtual* cuando se utiliza con el FAP. Encontró algunos planetas más, pero, para llegar a ellos tuvo que pagar el precio de los falsos positivos que ya se habían surgido en los gráficos anteriores.

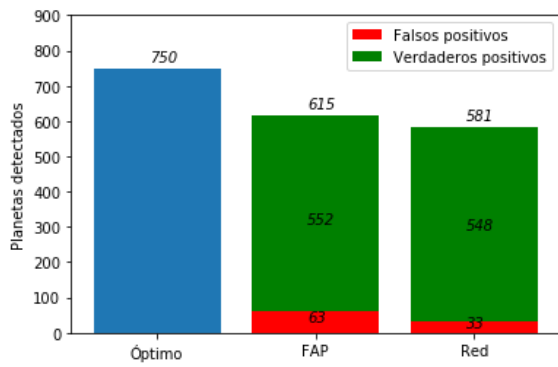
⁷ Recordemos que el hecho de que una estrella tenga N planetas no significa que todos esos planetas sean alcanzados por el *astrónomo virtual*, basta con que el planeta no esté en el pico máximo del periodograma para que este no lo “vea”.



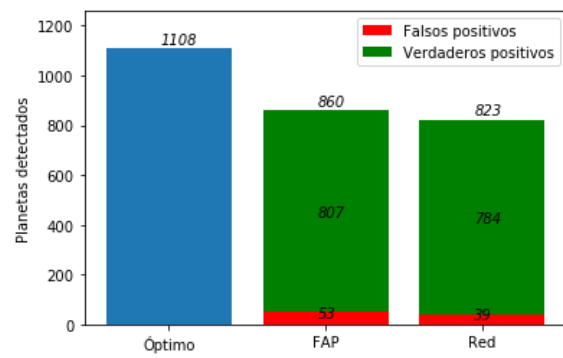
(a) 1000 estrellas con 0 planetas.



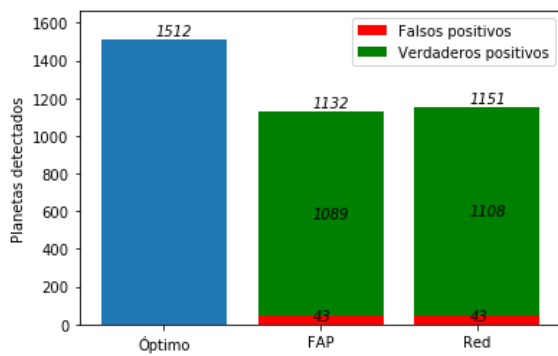
(b) 1000 estrellas con 1 planeta.



(c) 1000 estrellas con 2 planetas.



(d) 1000 estrellas con 3 planetas.



(e) 1000 estrellas con 4 planetas.

Fig. 3.9: Comparación entre la cantidad de detecciones del *astrónomo virtual*, con ambas implementaciones, contra el óptimo posible.

Para finalizar, y tener una visión mas general, se realizaron las matrices de confusión 3.5 y 3.6 de aplicar el método completo.

		Predicción	
		0	1
Valor real	0	4061	265
	1	922	2730

Tab. 3.5: Matriz de confusión FAP

		Predicción	
		0	1
Valor real	0	4031	190
	1	948	2719

Tab. 3.6: Matriz de confusión Red

Si bien no se pueden comparar las matrices directamente (el FAP siguió más veces, por lo que tiene más casos) si se pueden usar para calcular la precisión y exhaustividad de cada implementación.

	<i>Astrónomo virtual</i>	<i>Astrónomo virtual</i>
	FAP	Red
Precisión	0.911	0.934
Exhaustividad	0.747	0.741

Tab. 3.7: Precisión-Exhaustividad del Astrónomo virtual

A pesar de las diferencias en las ejecuciones del *astrónomo virtual* para cada implementación, los valores de precisión y exhaustividad mantienen la relación que tenían cuando se los aplicaba sin el método completo, es decir, exhaustividad prácticamente igual pero la red aumenta en precisión.

4. DISCUSIÓN Y CONCLUSIONES

“To myself I am only a child playing on the beach, while vast oceans of truth lie undiscovered before me.”

– Sir Isaac Newton

A lo largo de este trabajo, se presentó la técnica de velocidad radial para la búsqueda de planetas extrasolares, se mostraron sus diferentes componentes y se hizo hincapié en el cálculo de la *probabilidad de falsa alarma* (FAP) que, tomando una señal candidata, busca decidir si ésta procede de la influencia de un planeta o si es ruido. Se desarrolló también un mecanismo automatizado para la búsqueda de planetas (bautizado como *astrónomo virtual*) que, utilizando este cálculo, busca encontrar los planetas que se encuentran en una serie temporal de velocidades radiales.

Se simuló señales con perturbaciones planetarias y diversos componentes de ruido intrínseco, propio de estrellas de tipo solar, y se generaron unas 11 000 series temporales. Con esas series se construyeron unos 44 000 periodogramas con los que no solo se evaluó la calidad de detección del FAP, sino que además se usaron para entrenar y evaluar una red neuronal convolucional diseñada para reemplazarlo.

El área debajo de la curva en los gráficos de precisión-exhaustividad mostró categóricamente que la red es una mejor solución general que el FAP y, al buscar un umbral adecuado, se obtuvieron mejoras en la confianza de las predicciones al disminuir la cantidad de falsos positivos en un 28 % y, por ende, aumentar la precisión del *astrónomo virtual*.

Otro progreso significativo fue en el tiempo de cómputo necesario para el análisis de las señales, la red neuronal presenta una mejora de, al menos, cinco órdenes de magnitud en los tiempos de ejecución y el espacio en memoria que ocupa es apenas de 1.7kB.

Estas ventajas hacen de la red una gran candidata para reemplazar el cálculo del FAP y para cumplir uno de los objetivos principales de este trabajo, su aplicación en grandes volúmenes de datos.

4.1. Dificultades y posibles mejoras

La mayor dificultad fue al momento de decidir qué parte del problema modelar con la red. Fueron varias implementaciones y pruebas hasta poder encontrar, no solo una buena solución, sino una que se pudiese comparar directamente con algún otro método ya establecido y de esta forma poder evaluar realmente el comportamiento esta red y estimar su verdadero alcance.

Otra dificultad fue el tiempo excesivo que se necesitó para generar los sistemas, y que podía retrasar por varios días las diferentes pruebas, hasta que se tuvo un conjunto apropiado de periodogramas para las diferentes etapas de este trabajo.

4.2. Trabajo futuro

El principal trabajo futuro es buscar aplicar la red sobre datos reales. Para ello primero hay que realizar una modificación en la generación de las series temporales, principalmente en los tiempos de observación, donde es necesario modelar una toma de mediciones más realista. Si bien los tiempos modelados en este trabajo no son equiespaciados, están lejos de la gran variabilidad propia de las observaciones astronómicas. Esto es importante porque, si el muestreo es muy ocasional, aparecen muchos picos espúreos que dificultarían la detección de picos verdaderos. Seguramente se tenga que modificar la red para adaptarla a esta complejidad y también para poder aceptar periodogramas reales, que suelen ser más grandes. Con estos dos pasos ya se podrá entrenar y aplicar la red sobre series temporales obtenidas de mediciones reales.

Otras opciones pueden ser volver a analizar algunas de las diferentes variantes del problema mostradas en el apéndice B donde se pueden utilizar, además de los periodogramas, algunos de los otros componentes generadas durante las simulaciones, o incluso explorar las capacidades de las GAN (*Generative Adversarial Networks*) para generar datos lo más realistas posibles.

Bibliografía

*Lento en mi sombra, la penumbra hueca
exploro con el báculo indeciso,
yo, que me figuraba el Paraíso
bajo la especie de una biblioteca.*

– Jorge Luis Borges, Poema de los dones (21-24)

- [AGM⁺18] David J Armstrong, Maximilian N Günther, James McCormac, Alexis M S Smith, Daniel Bayliss, François Bouchy, Matthew R Burleigh, Sarah Casewell, Philipp Eigmüller, Edward Gillen, Michael R Goad, Simon T Hodgkin, James S Jenkins, Tom Loudon, Lionel Metrailler, Don Pollacco, Katja Poppenhaefer, Didier Queloz, Liam Raynard, Heike Rauer, Stéphane Udry, Simon R Walker, Christopher A Watson, Richard G West, and Peter J Wheatley. Automatic vetting of planet candidates from ground-based surveys: machine learning with NGTS. *Monthly Notices of the Royal Astronomical Society*, 478(3):4225–4237, 05 2018.
- [APS16] D. J. Armstrong, D. Pollacco, and A. Santerne. Transit shapes and self-organizing maps as a tool for ranking planetary candidates: application to Kepler and K2. *Monthly Notices of the Royal Astronomical Society*, 465(3):2634–2642, 11 2016.
- [Bal08] R. V. Baluev. Assessing the statistical significance of periodogram peaks. *Monthly Notices of the Royal Astronomical Society*, 385(3):1279–1285, 03 2008.
- [BKB⁺11] William J. Borucki, David G. Koch, Gibor Basri, Natalie Batalha, Timothy M. Brown, Stephen T. Bryson, Douglas Caldwell, Jørgen Christensen-Dalsgaard, William D. Cochran, Edna DeVore, Edward W. Dunham, Thomas N. Gautier III, John C. Geary, Ronald Gilliland, Alan Gould, Steve B. Howell, Jon M. Jenkins, David W. Latham, Jack J. Lissauer, Geoffrey W. Marcy, Jason Rowe, Dimitar Sasselov, Alan Boss, David Charbonneau, David Ciardi, Laurance Doyle, Andrea K. Dupree, Eric B. Ford, Jonathan Fortney, Matthew J. Holman, Sara Seager, Jason H. Steffen, Jill Tarter, William F. Welsh, Christopher Allen, Lars A. Buchhave, Jessie L. Christiansen, Bruce D. Clar-

- ke, Santanu Das, Jean-Michel Désert, Michael Endl, Daniel Fabrycky, Francois Fressin, Michael Haas, Elliott Horch, Andrew Howard, Howard Isaacson, Hans Kjeldsen, Jeffery Kolodziejczak, Craig Kulesa, Jie Li, Philip W. Lucas, Pavel Machalek, Donald McCarthy, Phillip MacQueen, Søren Meibom, Thibaut Miquel, Andrej Prsa, Samuel N. Quinn, Elisa V. Quintana, Darin Ragozzine, William Sherry, Avi Shporer, Peter Tenenbaum, Guillermo Torres, Joseph D. Twicken, Jeffrey Van Cleve, Lucianne Walkowicz, Fred C. Witteborn, and Martin Still. Characteristics of planetary candidates observed by kepler. ii. analysis of the first four months of data. *The Astrophysical Journal*, 736(1):19, 2011.
- [Bro20] Jason Brownlee. *Tour of Evaluation Metrics for Imbalanced Classification*, 2020. <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/>.
- [CBLM00] D. Charbonneau, T. M. Brown, D. W. Latham, and M. Mayor. Detection of Planetary Transits Across a Sun-like Star. *The Astrophysical Journal*, 529:L45–L48, jan 2000.
- [CRG⁺19] Alexander Chaushev, Liam Raynard, Michael Goad, Philipp Eigmüller, David Armstrong, Josh Briegal, Matthew Burleigh, Sarah Casewell, Sam Gill, James Jenkins, Louise Nielsen, Christopher Watson, Richard West, Peter Wheatley, Stéphane Udry, and Jose Vines. Classifying exoplanet candidates with convolutional neural networks: Application to the next generation transit survey. 07 2019.
- [dBVS⁺20] Zoe L. de Beurs, Andrew Vanderburg, Christopher J. Shallue, Xavier Dumusque, Andrew Collier Cameron, Lars A. Buchhave, Rosario Cosentino, Adriano Ghedina, Raphaëlle D. Haywood, Nicholas Langellier, David W. Latham, Mercedes López-Morales, Michel Mayor, Giusi Micela, Timothy W. Milbourne, Annelies Mortier, Emilio Molinari, Francesco Pepe, David F. Phillips, Matteo Pinamonti, Giampaolo Piotto, Ken Rice, Dimitar Sasselov, Alessandro Sozzetti, Stéphane Udry, and Christopher A. Watson. Identifying exoplanets with deep learning. iv. removing stellar activity signals from radial velocity measurements using neural networks, 2020.
- [Día18] Rodrigo F. Díaz. Modelling Light and Velocity Curves of Exoplanet Hosts. In *Asteroseismology and Exoplanets: Listening to the Stars and Searching for New Worlds*, volume 49, page 199, Jan 2018.

-
- [DSU⁺16] Díaz, R. F., Ségransan, D., Udry, S., Lovis, C., Pepe, F., Dumusque, X., Marmier, M., Alonso, R., Benz, W., Bouchy, F., Coffinet, A., Collier Cameron, A., Deleuil, M., Figueira, P., Gillon, M., Lo Curto, G., Mayor, M., Mordasini, C., Motalebi, F., Moutou, C., Pollacco, D., Pompei, E., Queloz, D., Santos, N., and Wyttenbach, A. The harps search for southern extra-solar planets - xxxviii. bayesian re-analysis of three systems. new super-earths, unconfirmed signals, and magnetic cycles. *A&A*, 585:A134, 2016.
- [DUL⁺11] X. Dumusque, S. Udry, C. Lovis, N. C. Santos, and M. J. P. F. G. Monteiro. Planetary detection limits taking into account stellar noise. i. observational strategies to reduce stellar oscillation and granulation effects. *Astronomy and astrophysics*, 525:A140, jan 2011.
- [FAEA⁺16] D Fischer, Guillem Anglada-Escudé, Pamela Arriagada, Roman Baluev, Jacob L. Bean, Francois Bouchy, Lars A. Buchhave, Thorsten Carroll, Abhijit Chakraborty, Justin R. Crepp, Rebekah I. Dawson, Scott A. Diddams, Xavier Dumusque, Jason D. Eastman, Michael Endl, P Figueira, Eric B. Ford, Daniel Foreman-Mackey, Paul Fournier, and Jason Wright. State of the field: Extreme precision radial velocities. 128, 02 2016.
- [Fuk80] Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [Heb49] Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949.
- [HKP91] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Longman Publishing Co., Inc., USA, 1991.
- [HMBV00] Gregory W. Henry, Geoffrey W. Marcy, R. Paul Butler, and Steven S. Vogt. A Transiting “51 Peg-like” Planet. *The Astrophysical Journal*, 529(1):L41–L44, January 2000.
- [HW59] David H. Hubel and Torsten N. Wiesel. Receptive fields of single neurons in the cat’s striate cortex. *Journal of Physiology*, 148:574–591, 1959.

- [KJY17] Muhammad Salman Khan, James Jenkins, and Nestor Yoma. Discovering new worlds: a review of signal processing methods for detecting exoplanets from astronomical radial velocity data. *IEEE Signal Processing Magazine*, 34, 01 2017.
- [LBD⁺89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.
- [MH08] Eric E. Mamajek and Lynne A. Hillenbrand. Improved age estimation for solar-type dwarfs using activity-rotation diagnostics. *The Astrophysical Journal*, 687(2):1264–1293, nov 2008.
- [MJC⁺15] Sean D. McCauliff, Jon M. Jenkins, Joseph Catanzarite, Christopher J. Burke, Jeffrey L. Coughlin, Joseph D. Twicken, Peter Tenenbaum, Shawn Seader, Jie Li, and Miles Cote. AUTOMATIC CLASSIFICATION OF KEPLER PLANETARY TRANSIT CANDIDATES. *The Astrophysical Journal*, 806(1):6, jun 2015.
- [MNFSH18] Pablo Márquez-Neila, Chloe Fisher, Raphael Sznitman, and Kevin Heng. Supervised machine learning for analysing spectra of exoplanetary atmospheres. *Nature Astronomy*, 2, 09 2018.
- [MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.
- [MQ95] M. Mayor and D. Queloz. A Jupiter-mass companion to a solar-type star. *Nature*, 378:355–359, nov 1995.
- [Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [SMB⁺18] Snehanshu Saha, Archana Mathur, Kakoli Bora, Surbhi Agrawal, and Suryoday Basak. SbaF: A new activation function for artificial neural net based habitability classification. 06 2018.
- [SV18] Christopher J. Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94, jan 2018.

-
- [UDL⁺19] Udry, S., Dumusque, X., Lovis, C., Ségransan, D., Diaz, R. F., Benz, W., Bouchy, F., Coffinet, A., Lo Curto, G., Mayor, M., Mordasini, C., Motalebi, F., Pepe, F., Queloz, D., Santos, N. C., Wyttenbach, A., Alonso, R., Collier Cameron, A., Deleuil, M., Figueira, P., Gillon, M., Moutou, C., Pollacco, D., and Pompei, E. The harps search for southern extra-solar planets - xlv. eight harps multi-planet systems hosting 20 super-earth and neptune-mass companions. *A&A*, 622:A37, 2019.
- [Wal16] I. P. Waldmann. Dreaming of Atmospheres. *The Astrophysical Journal*, 820(2):107, April 2016.
- [WF92] A. Wolszczan and D. A. Frail. A planetary system around the millisecond pulsar psr1257 + 12. *Nature*, 355:145 EP –, Jan 1992.
- [ZK09] Zechmeister, M. and Kürster, M. The generalised lomb-scargle periodogram - a new formalism for the floating-mean and keplerian periodograms. *A&A*, 496(2):577–584, 2009.

Apéndice

A. REDES NEURONALES, BREVE HISTORIA Y EVOLUCIÓN

“I am glad you are here with me. Here at the end of all things, Sam.”

– J.R.R. Tolkien, The Return of the King

Como se señaló en la introducción, en esta nueva explosión de la inteligencia artificial frecuentemente se mencionan términos como *aprendizaje automático*, *aprendizaje profundo*, *redes neuronales* o la propia *inteligencia artificial* como si fuesen cosas equivalentes e intercambiables. Es importante establecer una diferenciación entre estos conceptos y áreas, por lo menos básica, a fin de posicionarnos y poder detallar las implementaciones que fueron utilizadas en este trabajo.

El concepto más general de los anteriores es el de inteligencia artificial y nació como disciplina en la década del 50 en lo que se conoce como los *Dartmouth workshops*, bajo la entusiasta suposición de que la inteligencia humana *“Puede ser tan precisamente descripta que se puede hacer una máquina que la simule”*¹. Las definiciones fueron cambiando con el tiempo, pero la idea general es que la inteligencia artificial es la inteligencia llevada a cabo por máquinas (en contraparte a la inteligencia humana) y donde estos *agentes inteligentes* perciben su entorno y toman decisiones que maximizan las posibilidades de alcanzar sus objetivos.

Dentro de esta disciplina se encuentra el aprendizaje automático que, utilizando la definición de la universidad de Stanford, es *“La ciencia de hacer que las computadoras actúen de una manera específica sin programarlas explícitamente para que lo hagan”*, esto es, se le proveen datos a una computadora para que *aprenda*, de forma tal que pueda realizar predicciones acertadas ante nuevos escenarios, por lo que es una manera de implementar la inteligencia artificial. Se puede pensar que es algo relativamente moderno pero técnicas clásicas como la regresión lineal (que ya tiene unos doscientos años) entran en este concepto general.

Otro de los términos en boga es el de aprendizaje profundo. Este es un subconjunto del aprendizaje automático y es un tipo de aprendizaje que está inspirando en la forma en que el cerebro humano procesa la información. Se lo llama profundo porque se implementa con una red neuronal de varias capas.. En la figura [A.1](#) se muestra un diagrama con esta jerarquía.

Se comenzará definiendo una neurona artificial y se irán ampliando conceptos hasta llegar al aprendizaje profundo.

¹ <http://raysolomonoff.com/dartmouth/boxa/dart564props.pdf>

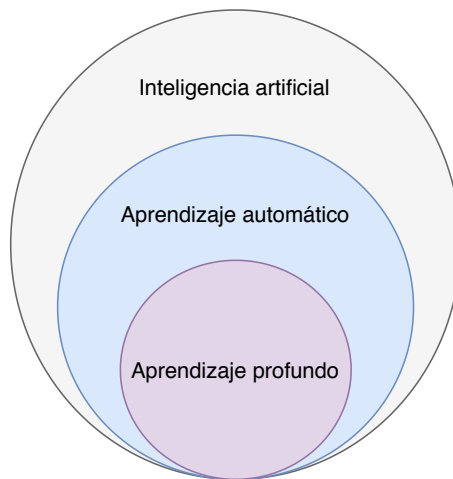


Fig. A.1: Jerarquía.

Todo el trabajo en redes neuronales artificiales, o simplemente redes neuronales, fue motivado desde sus comienzos por el hecho que el cerebro humano tiene una forma de trabajar y resolver problemas completamente distinto al de las computadoras y, no solo eso, en algunas tareas es claramente superior y tiene características que serían muy deseables en cualquier sistema artificial:

- Es robusto y tolerante a fallas: todos los días mueren células nerviosas sin afectar significativamente su rendimiento.
- Es flexible: se adapta y ajusta a un nuevo entorno *aprendiendo*.
- Puede trabajar con información confusa, ruidosa, incompleta o inconsistente.
- Es masivamente paralelo.
- Es pequeño, compacto y disipa muy poca energía.

Parte de su éxito en estas áreas es por su capacidad de organizar sus componentes estructurales, conocidos como neuronas, para realizar cálculos específicos (por ejemplo, reconocimiento de patrones, percepción y control motor) varias veces más rápido que las computadoras digitales. Solo en acciones que involucren tareas aritméticas la computadora le gana al cerebro.

A.1. Neuronas artificiales y redes neuronales

Nuestro cerebro está compuesto por cerca de 10^{11} neuronas de muchos tipos. Redes de forma arbórea de nervios llamados **dendritas** están conectadas al cuerpo de la célula

o **soma**, donde el núcleo de la célula está localizado. Desde el cuerpo se extiende una única fibra llamada **axón** y en los extremos de estos se encuentran los transmisores de las uniones sinápticas, o **sinapsis**, a otras neuronas.

La transmisión de una señal de una célula a otra en una sinapsis es un proceso químico complejo en el que se liberan sustancias transmisoras específicas desde el lado emisor de la unión. El efecto es aumentar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora y, si este potencial alcanza un umbral, se envía un impulso (de fuerza y duración fijas por el axón), es ahí cuando decimos que la célula ha “disparado”. Luego el pulso se ramifica a través de la arborización axonal a uniones sinápticas con otras células. En la figura A.2. se puede ver un esquema de una neurona biológica.

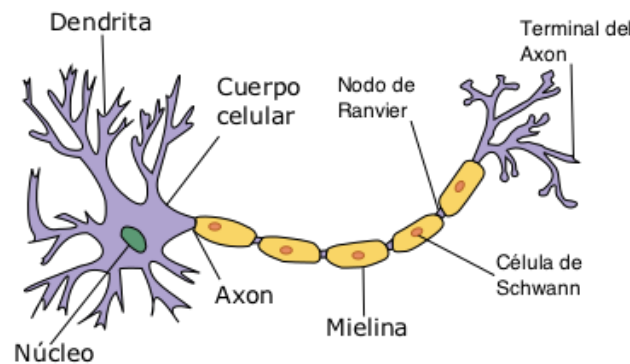


Fig. A.2: Representación de una neurona biológica, Wikipedia.

Basado en esta comprensión básica del principio operativo de la neurona, McCulloch & Pitts propusieron el primer modelo matemático de una neurona artificial en su artículo de 1943 llamado “*A logical calculus of the ideas immanent in nervous activity*” [MP43]. Si bien este modelo era bastante simple ha demostrado ser extremadamente versátil y en 1958 Frank Rosenblatt lo tomó como base para el **perceptrón** [Ros58].

Rosenblatt utilizó el concepto de Donald Hebb de aprendizaje auto organizado [Heb49] y mostró que realizando algunas modificaciones a la idea original de McCulloch & Pitts se podía lograr que las neuronas artificiales aprendan de los datos.

El perceptrón, cuya representación puede verse en la figura A.3., consta de cuatro partes:

- Valores de entrada, x_1, x_2, \dots, x_n .
- Pesos w_1, w_2, \dots, w_n , y una constante asociada al peso sináptico, también conocida como sesgo o *bias* cuyo valor es 1. Los pesos permiten que el perceptrón evalúe la importancia relativa de cada una de las entradas en el resultado y el sesgo es un paso técnico que permite mover la curva de la función de activación, o sea, permite

ajustar la salida numérica del perceptrón.

- Suma ponderada de las entradas con sus pesos.
- Función de activación σ . Esta función determina cuando debe dispararse o no la salida del perceptrón basada en lo relevante que sea la entrada a la predicción del modelo, si se supera cierto *threshold* se determina una salida x , caso contrario será y .

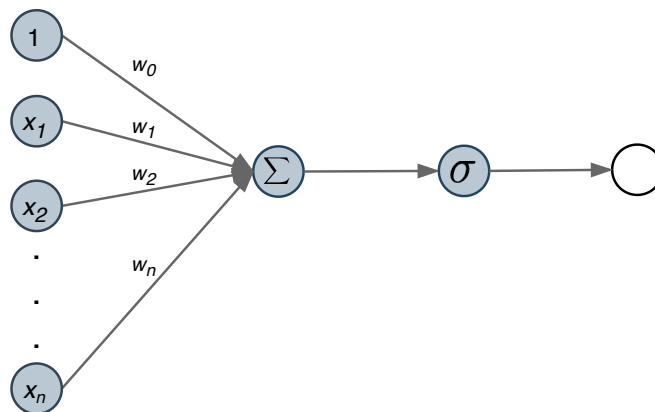


Fig. A.3: Neurona artificial del perceptrón.

La salida del perceptrón es binaria, por lo que esta neurona puede servir para resolver un problema de clasificación (recibe una entrada y decide si pertenece o no a cierta clase) o también se podría modelar un problema de lógica donde la salida es *verdadero* o *falso*.

Para que el perceptrón aprenda se necesita un conjunto de datos de entrada con sus respectivas etiquetas, se llamará a esto el *conjunto de entrenamiento*. El procedimiento que se realiza es el siguiente:

1. Se inicializan los pesos del perceptrón con valores aleatorios.
2. Se toma un ítem del conjunto de entrenamiento y se computa la salida del perceptrón multiplicando las entradas por sus pesos y calculando su suma.
3. Se pasa este valor a la función de activación y se obtiene un resultado.
4. Si la salida no coincide con la esperada, se modifican los pesos para que lo haga. Por ejemplo: si la salida fue 0, y debería haber sido 1, se incrementan los pesos; y si fue 1, y debería ser 0, se decrementan.
5. Se toma el siguiente elemento del conjunto de entrenamiento y se repiten los pasos 2 a 4 hasta que el perceptrón no cometa más errores.

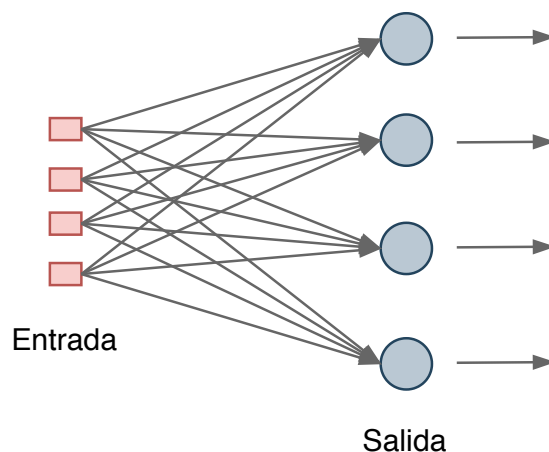


Fig. A.4: Multi perceptrón.

Como se mencionó anteriormente, este procedimiento permite entrenar un perceptrón para distinguir entre dos posibles categorías, pero ¿qué pasa si el problema de clasificación que se está tratando de resolver tiene más de una categoría? la solución es simple y se muestra en la figura A.4.: hay que agregar más perceptrones (uno por cada categoría) en lo que se llama una *capa* y se los interconecta de forma que todos reciban la misma entrada. De esta forma, la salida de cada perceptrón es responsable por una de las categorías, devolviendo 1 y todos los demás 0.

Rosenblatt implementó esta idea en hardware, y demostró que podría usarse para aprender a clasificar correctamente formas simples utilizando entradas de 20x20 píxeles. Y así nació el aprendizaje automático.

A.2. Redes profundas

La simplicidad y eficiencia de este algoritmo de aprendizaje para problemas linealmente separables es una de las razones por las que se hizo tan popular a fines de los años cincuenta y principios de los sesenta. Sin embargo, esta popularidad hizo que Rosenblatt sobreestimara la capacidad de aprendizaje del perceptrón, dando lugar a expectativas poco realistas en la comunidad científica. La realidad es que este perceptrón adolece de grandes limitaciones que restringen en gran medida su aplicabilidad a problemas de la vida real, como mostraron en 1969 Marvin Minsky y Seymour Papert en el libro “*Perceptrons An Introduction to Computational Geometry*” [MP69].

En este libro, los autores no solo mostraron que es imposible que un perceptrón aprenda la simple función lógica XOR, sino que específicamente argumentaron que la forma en la que podía resolverse era con *múltiples capas* de perceptrones (lo que hoy llamamos redes profundas) ejemplificado en la figura A.5. Esto conllevaba un problema, el algoritmo de

aprendizaje de Rosenblatt no funcionaba para más de una capa. La razón es bastante intuitiva, si recordamos el procedimiento por el que un perceptrón aprende veremos que solo se especifica la salida correcta para, justamente, la capa de salida, no tenemos idea como se ajustarían los pesos de las capas interiores.

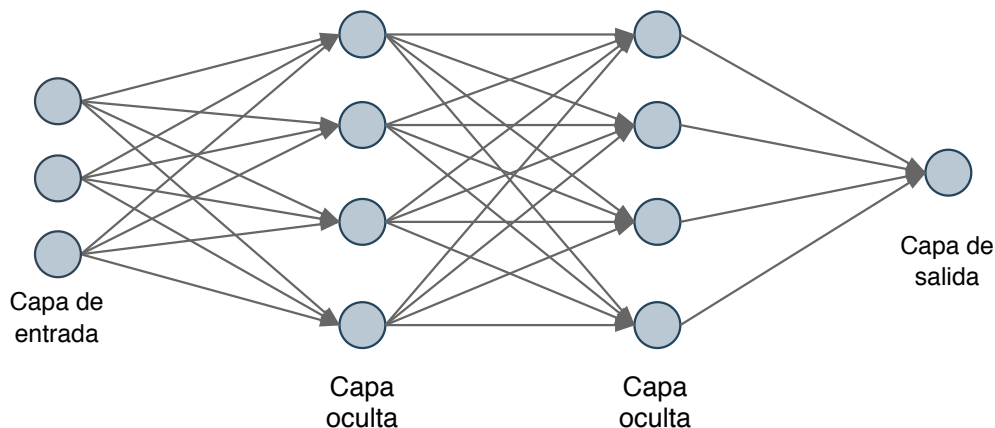
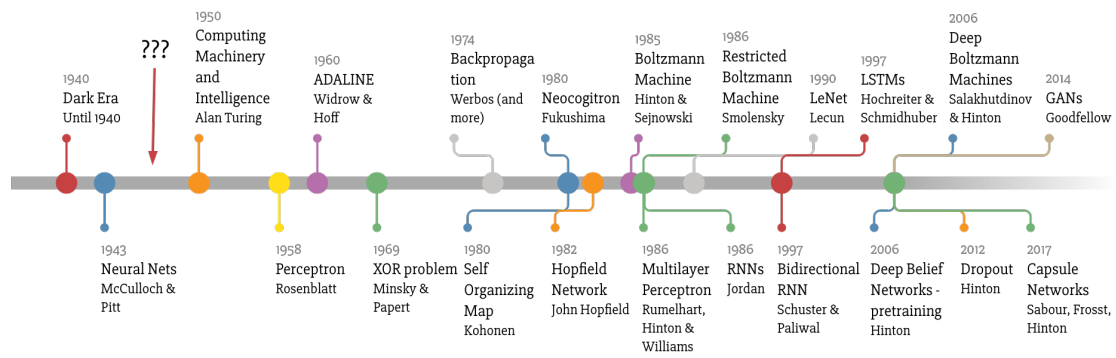


Fig. A.5: Perceptrón profundo

La solución vino de la mano de la regla de la cadena. No importaba mucho si las neuronas no eran del todo perceptrones, pero si la función de activación utilizada (además de ser no lineal) era diferenciable, se podía usar la derivada para ajustar los pesos y minimizar el error, y no solo eso, usando la regla de la cadena se podía calcular la derivada de todas las neuronas de la capa anterior y poder ajustarlas.

Más simplemente: Se puede usar el cálculo para asignar parte de la *culpa* de cualquier error del conjunto de entrenamiento en la capa de salida a cada neurona en la capa oculta anterior, y luego, dividir aún más esta *culpa* si hay otra capa oculta. De esta forma se puede ir *propagando* el error hacia atrás, este mecanismo se denominó **backpropagation**. De esta forma se podía encontrar cuánto cambia el error al cambiar cualquier peso en la red neuronal. La técnica más común para encontrar los pesos óptimos que minimizan el error es la del descenso de gradiente estocástico, una variación de la clásica técnica de Cauchy.

En la figura A.6. se puede ver una interesante cronología de algunos de los diferentes avances en este campo.



Made by Favio Vázquez

Fig. A.6: Línea de tiempo del aprendizaje profundo²

² A “weird” Introduction to Deep Learning., Favio Vázquez. <https://www.bbvadata.com/a-weird-introduction-to-deep-learning>

B. MODELOS DESCARTADOS

Durante la etapa de modelado se barajaron varias opciones antes de llegar a la solución propuesta. Algunas de estas consideraciones se descartaron rápidamente y otras se exploraron más en detalle antes de elegir otro camino. Estas fueron algunas de las más importantes:

- Darle a la red la serie temporal y pensarlo como un problema de clasificación con una clase para cada cantidad de planetas. Fue la primera prueba cuando se contó con la generación de las series temporales. Fue interesante porque además de las redes convolucionales se probó con redes LSTM¹, pero en realidad no se ajustaba mucho a la definición de problema de clasificación (las clases no son realmente disjuntas) y las pruebas apenas daban mejoras sobre una clasificación aleatoria.

Después de descartar este enfoque fue que se empezaron a usar los periodogramas, por eso se “heredó” el muestreo que se menciona en la sección de trabajo futuro.

- Darle a la red el periodograma y que la red diga cuales de los picos eran planetas. Fue una opción muy ambiciosa, el problema es que era muy difícil de validar, y prácticamente imposible de usar en la práctica. Si, por ejemplo, la red predecía una posición entre dos picos no se puede saber cual elegir.
- Pensarlo como un problema de regresión lineal: la red recibía un periodograma y la salida era un número real que buscaba ser la cantidad de planetas presentes. Se investigó bastante esta opción, la red parecía diferenciar bien los periodogramas sin planetas de los otros, pero los resultados eran más bien difusos. Haciendo pruebas, se llegó a la conclusión que la red estaba haciendo una especie de “suma de potencias” de los picos, o sea, si el periodograma tenía un pico muy alto la red daba un número más cercano a 4, y con picos más bajos la red daba valores más chicos.

¹ Las LSTM (*Long Short Term Memory*) son un tipo de red recurrente. Estas redes mantienen un estado interno (una “memoria”) donde pueden almacenar información sobre ejecuciones anteriores. Suelen ser útiles en problemas donde se quiere tener en cuenta información de contexto.