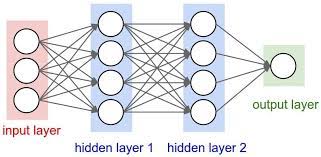
# Deep Nets

lunes, 15 de Enero de 2024

1. Abreviatura de Deep neural networks (redes neuronales profundas)
2. Un tipo de modelo de aprendizaje automático.
3. Este tipo de red esta compuesto por capas de neuronas interconectadas. Redes para reconocimiento de patrones, clasificación y generación de contenido.
4. Las conocidas son:

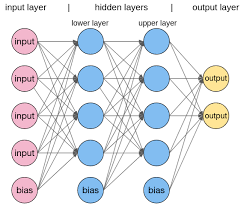
|  |  |
| --- | --- |
| CNN | RNN |
| Red neuronal convolucional | Red neuronal recurrente |
| Tareas de visión por computadora. | Tareas de secuencia de datos, como el lenguaje natural. |



# MLP

1. O multilayer perceptrón.
2. Es un tipo de red neuronal artificial que pertenece a la categoría de redes alimentadas hacia adelante.
3. Tenemos al menos 3 capas:

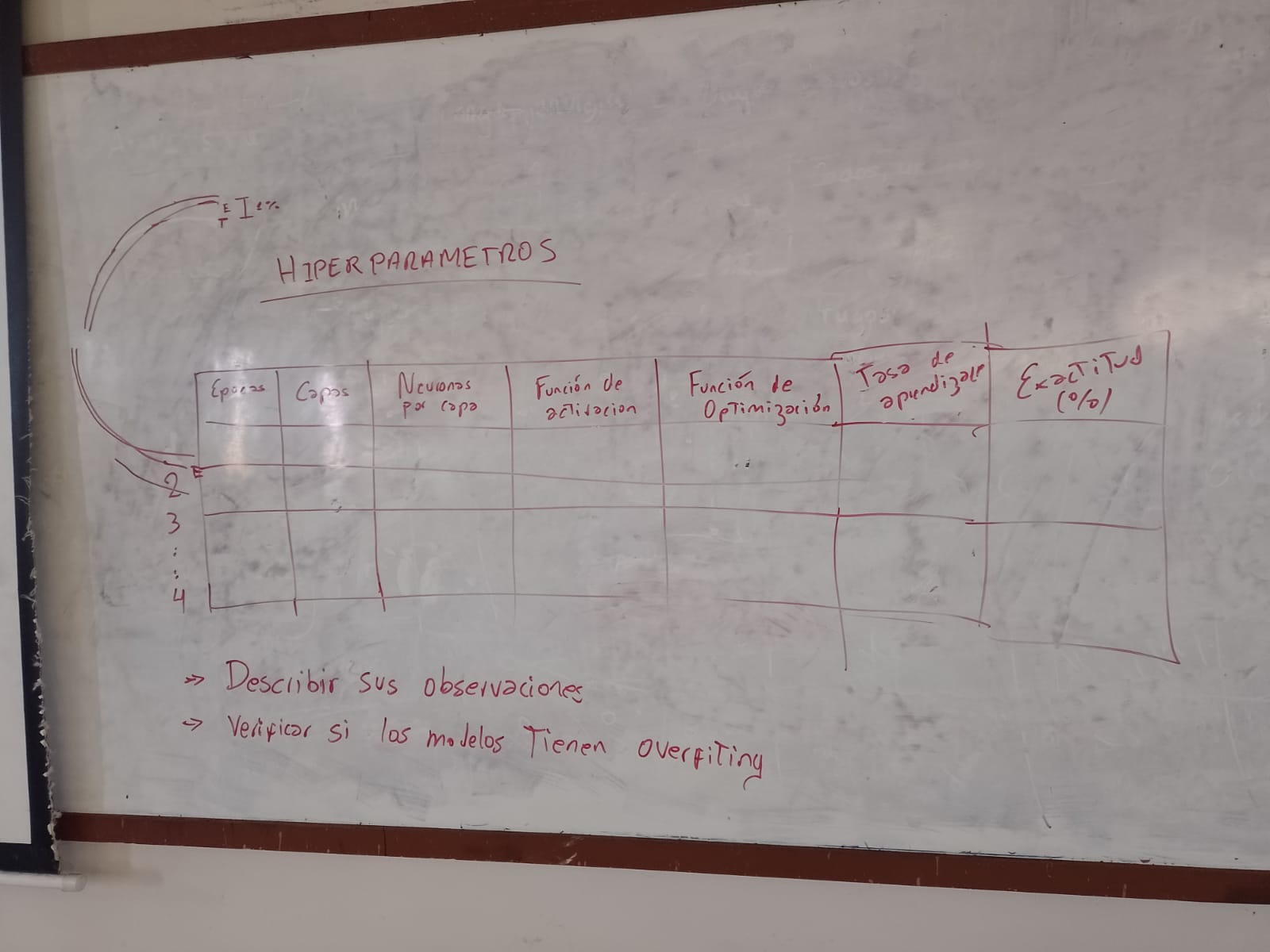
|  |  |
| --- | --- |
| Capa de entrada | Recibe señales de entrada. |
| Capas ocultas | Procesan información, usando pesos. |
| Capa de salida | Produce la salida final de la red. |



PROBANDO CODIGO DEL PROFESOR:

lunes, 15 de Enero de 2024

# La tarea:

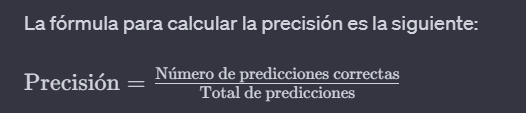


# LOSS

1. En machine learning se traduce como “perdida”.
2. Representa la discrepancia entre las predicciones del modelo y las respuestas reales.
3. Se tiene como objetivo minimizar esta perdida.

## ACCURACY

1. Precisión en español.
2. Esta metrica sirve para evaluar el rendimiento de un modelo de clasificación.



# PERDIDA DE ENTRENAMIENTO Y PERDIDA DE VALIDACION:

La pérdida de entrenamiento y la pérdida de validación son métricas que se utilizan para evaluar el rendimiento de un modelo de machine learning, especialmente en el contexto de redes neuronales. Estas métricas están relacionadas con la capacidad del modelo para aprender y generalizar a datos no vistos. Aquí hay una explicación más detallada:

## 1. \*\*Pérdida de Entrenamiento:\*\*

- La pérdida de entrenamiento es la medida de cuánto se equivoca el modelo en sus predicciones sobre el conjunto de datos de entrenamiento durante el proceso de entrenamiento.

- Durante cada época del entrenamiento, el modelo ajusta sus pesos y sesgos para minimizar esta pérdida. La meta es que la pérdida de entrenamiento disminuya a medida que el modelo aprende de los datos de entrenamiento.

## 2. \*\*Pérdida de Validación:\*\*

- La pérdida de validación mide cuánto se equivoca el modelo en sus predicciones sobre un conjunto de datos que no ha visto durante el entrenamiento, conocido como conjunto de datos de validación.

- El conjunto de datos de validación se utiliza para evaluar cómo bien el modelo generaliza a datos nuevos y no vistos. Durante el entrenamiento, el modelo no ajusta sus pesos y sesgos basándose en la pérdida de validación, sino en la pérdida de entrenamiento.

## 3. \*\*Sobreajuste (Overfitting):\*\*

- Un objetivo clave al entrenar modelos es lograr que el modelo aprenda patrones generales en los datos, en lugar de memorizar los datos de entrenamiento específicos. Si la pérdida de entrenamiento sigue disminuyendo, pero la pérdida de validación comienza a aumentar, podría ser un signo de sobreajuste.

- El sobreajuste ocurre cuando el modelo se adapta demasiado a los detalles idiosincráticos del conjunto de datos de entrenamiento y no generaliza bien a nuevos datos.

## 4. \*\*Convergencia:\*\*

- Idealmente, tanto la pérdida de entrenamiento como la pérdida de validación deberían disminuir y estabilizarse con el tiempo, indicando que el modelo está convergiendo y aprendiendo patrones útiles.

## 5. \*\*Ajuste de Hiperparámetros:\*\*

- Las gráficas de pérdida de entrenamiento y validación son útiles para ajustar hiperparámetros, como la tasa de aprendizaje o el tamaño de la red, para mejorar el rendimiento del modelo.

En resumen, la pérdida de entrenamiento y la pérdida de validación son medidas críticas para evaluar el rendimiento de un modelo. Observar cómo se comportan estas métricas durante el entrenamiento puede proporcionar información valiosa sobre la capacidad del modelo para aprender y generalizar.

# PERDIDA DE ENTRENAMIENTO Y PERDIDA DE VALIDACION DE EXACTITUD (ACCURACY)

¡Entendido! Si te refieres a la exactitud de entrenamiento y la exactitud de validación en el contexto de redes neuronales, las consideraciones son las siguientes:

## 1. \*\*Exactitud de Entrenamiento:\*\*

- La exactitud de entrenamiento se refiere a la proporción de muestras de entrenamiento clasificadas correctamente en relación con el total de muestras de entrenamiento. Es una medida de qué tan bien el modelo ha aprendido los patrones en los datos de entrenamiento.

- Una alta exactitud de entrenamiento indica que el modelo está realizando bien en los datos utilizados para entrenarlo.

## 2. \*\*Exactitud de Validación:\*\*

- La exactitud de validación mide la proporción de muestras de un conjunto de datos de validación clasificadas correctamente en relación con el total de muestras en ese conjunto. Este conjunto de datos de validación es independiente del conjunto de entrenamiento y se utiliza para evaluar la capacidad de generalización del modelo a datos no vistos.

- Es importante que la exactitud de validación sea alta para asegurarse de que el modelo no esté simplemente memorizando los datos de entrenamiento (sobreajuste), sino que realmente está aprendiendo patrones que se pueden aplicar a datos nuevos.

## 3. \*\*Comparación entre Exactitud de Entrenamiento y Validación:\*\*

- Idealmente, querrías que la exactitud de entrenamiento y la exactitud de validación fueran altas y cercanas entre sí. Esto indicaría que el modelo no solo está memorizando los datos de entrenamiento sino que también generaliza bien a datos nuevos.

- Si la exactitud de entrenamiento es significativamente mayor que la exactitud de validación, podría ser una señal de sobreajuste, indicando que el modelo está demasiado adaptado a los datos de entrenamiento y no generaliza bien.

## 4. \*\*Monitoreo durante el Entrenamiento:\*\*

- Al igual que con la pérdida y el accuracy, es común monitorear tanto la exactitud de entrenamiento como la exactitud de validación durante el entrenamiento. Esto ayuda a identificar posibles problemas de sobreajuste o subajuste, y también puede guiar el ajuste de hiperparámetros.

En resumen, la exactitud de entrenamiento y la exactitud de validación son métricas cruciales para evaluar cómo se está desempeñando un modelo durante el entrenamiento y cómo generaliza a datos nuevos y no vistos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PERDIDA DE ENTRENAMIENTO | PERDIDA DE VALIDACION | EXACTITUD DE ENTRENAMIENTO | EXACTITUD DE VALIDACION |
|  | (Cuanto se equivoca en predecir)/(conjunto de datos de entrenamiento) , durante el proceso de entrenamiento. | (Cuantó se equivoca)/(conjunto de datos no visto). | (Cuantas muestra se clasificaron bien)/(total de muestras).  Durante el entrenamiento. | (Datos clasificados correctamente)/(conjuntos de datos no vistos). |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

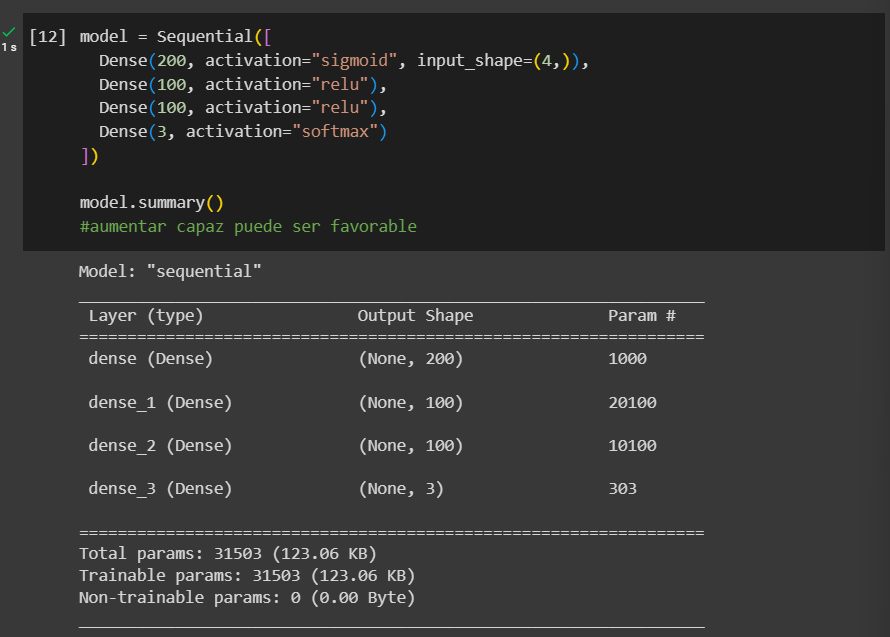
TAREA:

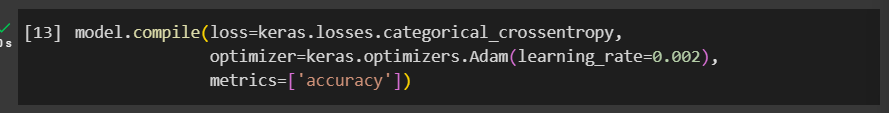
A continuación vamos a realizar algunos experimentos en el cuadernillo de “Redes neuronales” cambiando algunos hiperparametros para obtener los resultados esperados.

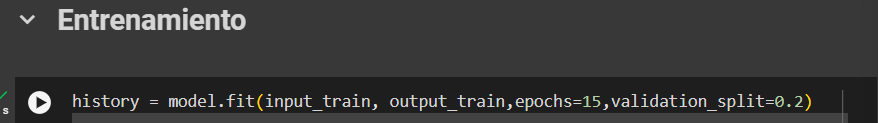
## EXPERIMENTO 1

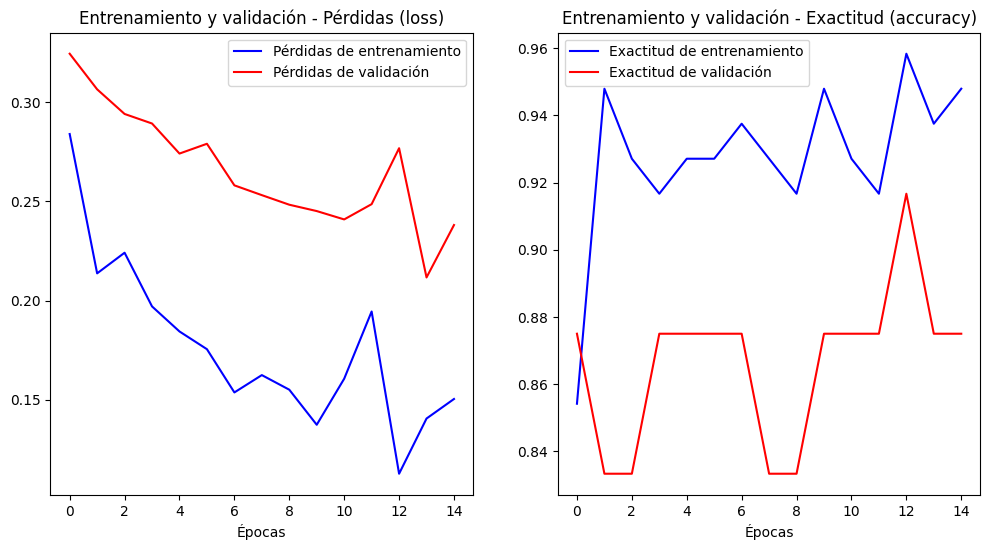
El primer experimento no servirá de referencia para comparar los resultados.

Podemos ver que el primer grafico las curvas se distancias mucho, aunque indican que las perdidas están disminuyendo. En la segunda grafica los resultados no son muy optimos asi que trataremos de cambiar estos resultados.





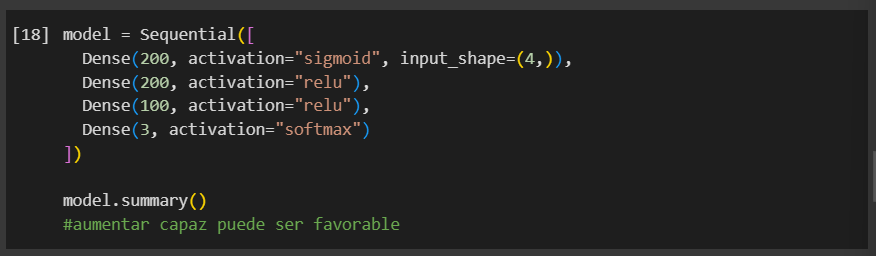


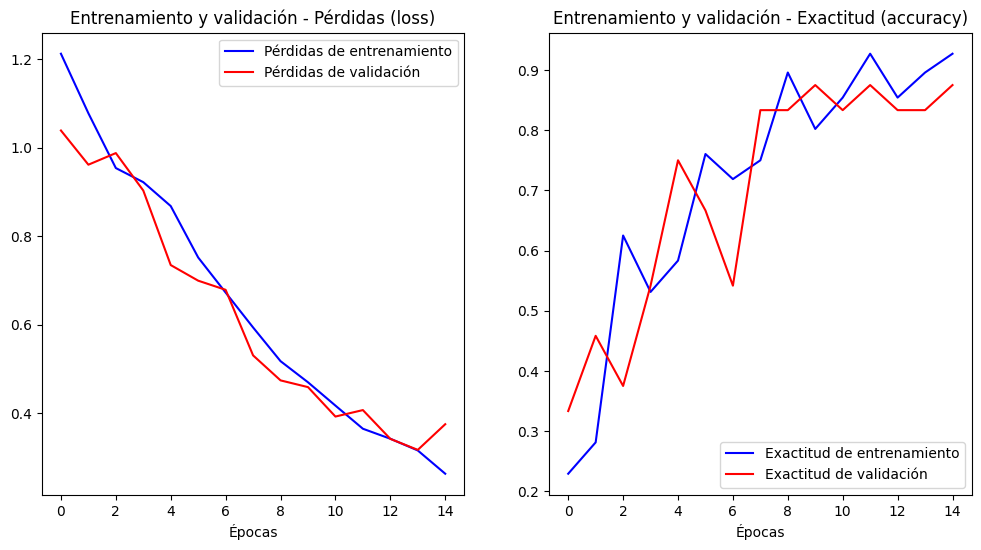


## EXPERIMENTO 2

Cambio en el numero de neuronas en la segunda capa.

Con este solo cambio obtenemos resultados muy favorables. En la primer y segunda grafica las curvas son mas homogéneas.



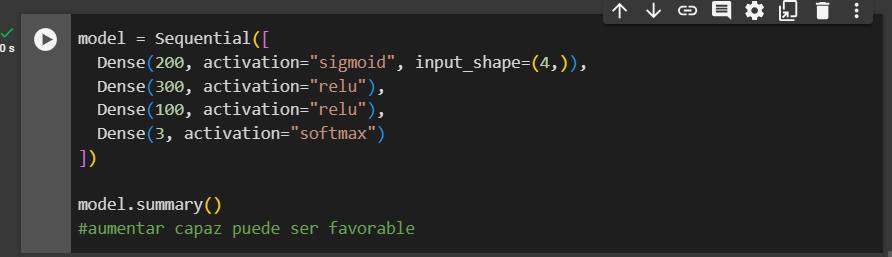


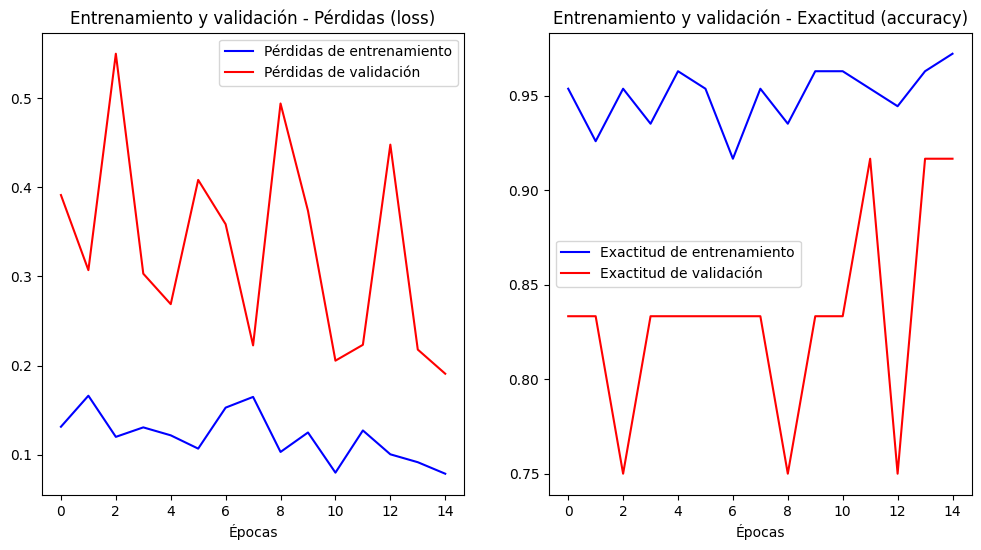
No hay overfiting pues so obtiene mejores resultado y curvas estables.

## EXPERIMENTO 3

Nuevamente aumentamos el numero de neuronas en la segunda capa.

Los resultados de validación no son buenos.



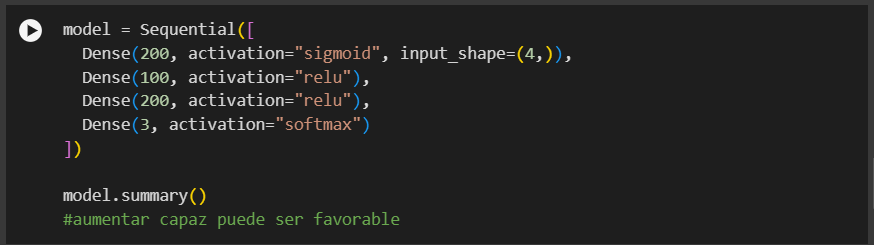


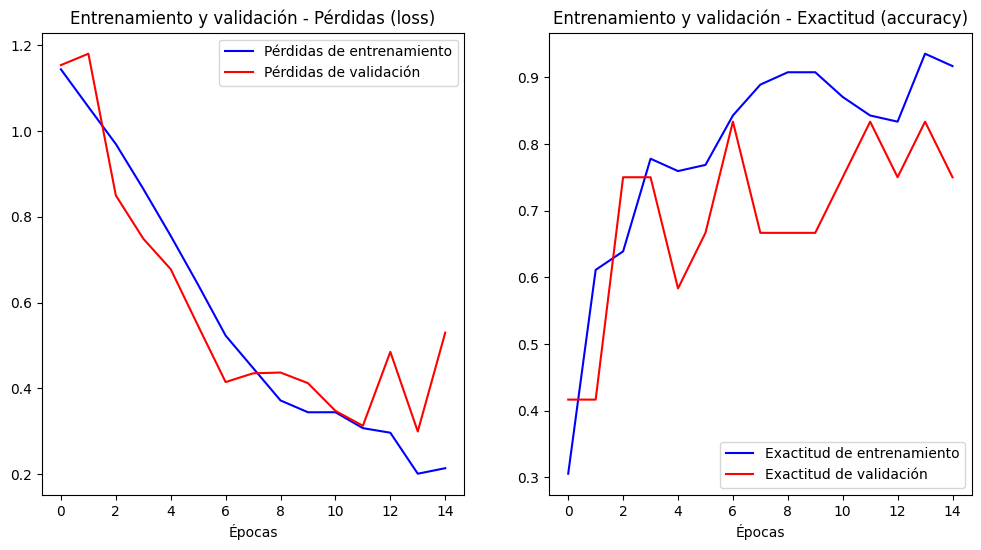
Hay overfiting dada la divergencia de las curvas en ambos graficos.

## EXPERIMENTO 4

Ahora hacemos un aumento en el numero de neuronas de la tercera capa.

Los resultados son buenos como en el experimento 2.



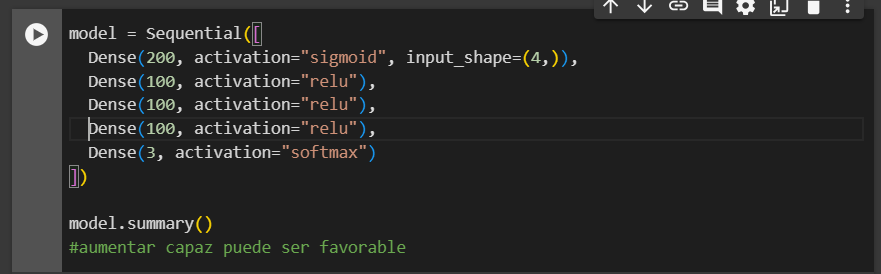


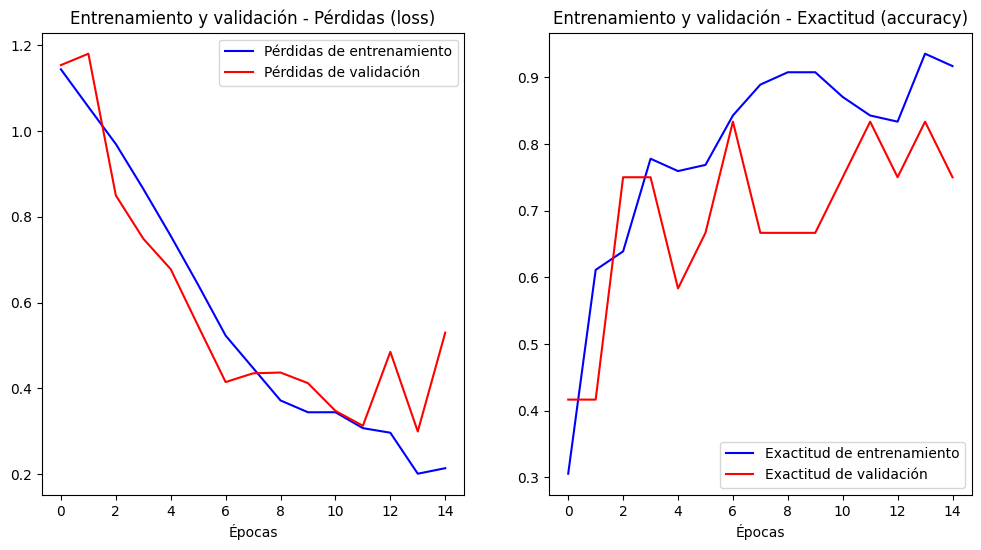
Consideramos que no hay overfiting en la curva perdidas.

## EXPERIMENTO 5

Ahora aumentamos una capa.

No hay overfiting dado que tenemos resultados coherentes de mejora en cada época sin embargo se podría tener mejores resultados.

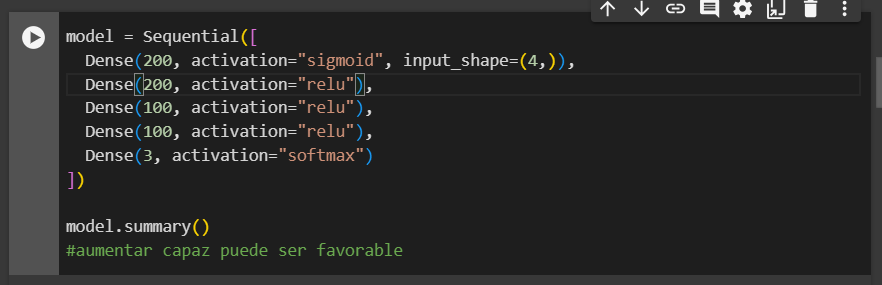


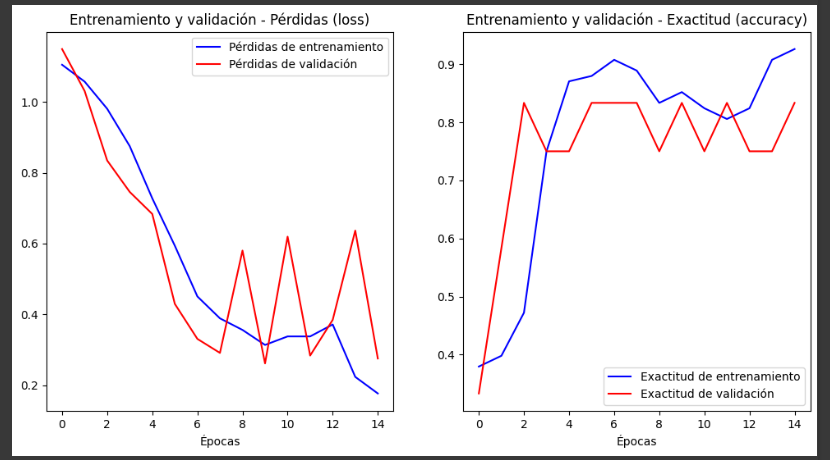


## EXPERIMENTO 6

El experimento anterior mas un aumento de neuronas en la primera capa.

Vemos la aparición de pico en la curva de perdidas de validación. No hay overfiting sin embargo se podría tener mejores resultados.

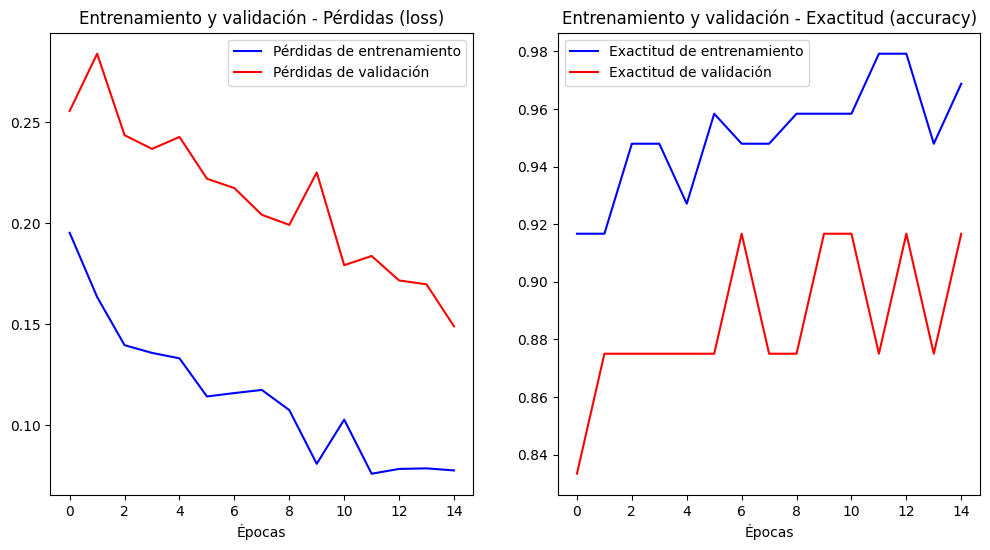




## EXPERIEMENTO 7

Al experimento 5 le cambiamos el valor de “validation\_slip”. Este parámetro nos indica el porcentaje de muestras que se usaran de validación. Hay overfiting.

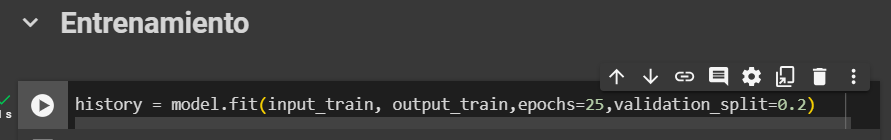


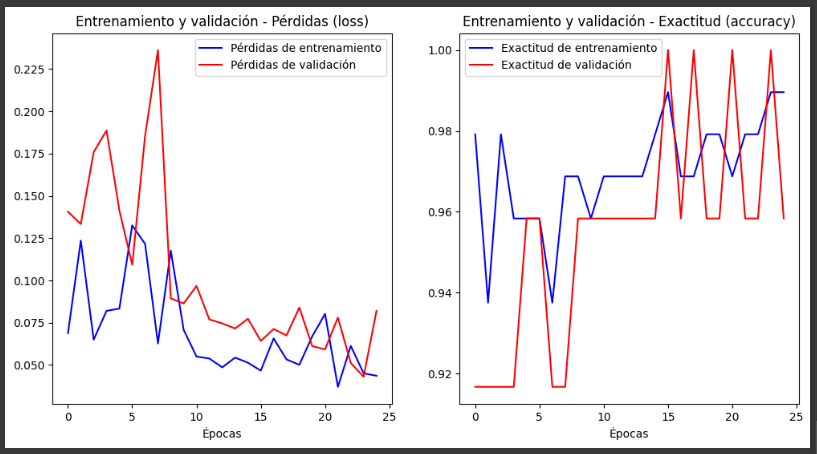


## EXPERIMENTO 8

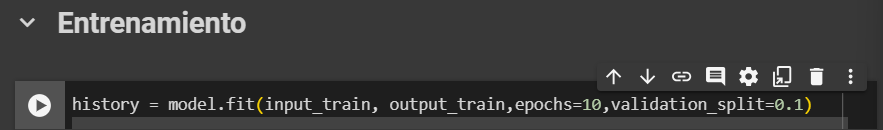
Al caso anterior le variamos el número de épocas de 15 a 25 épocas.

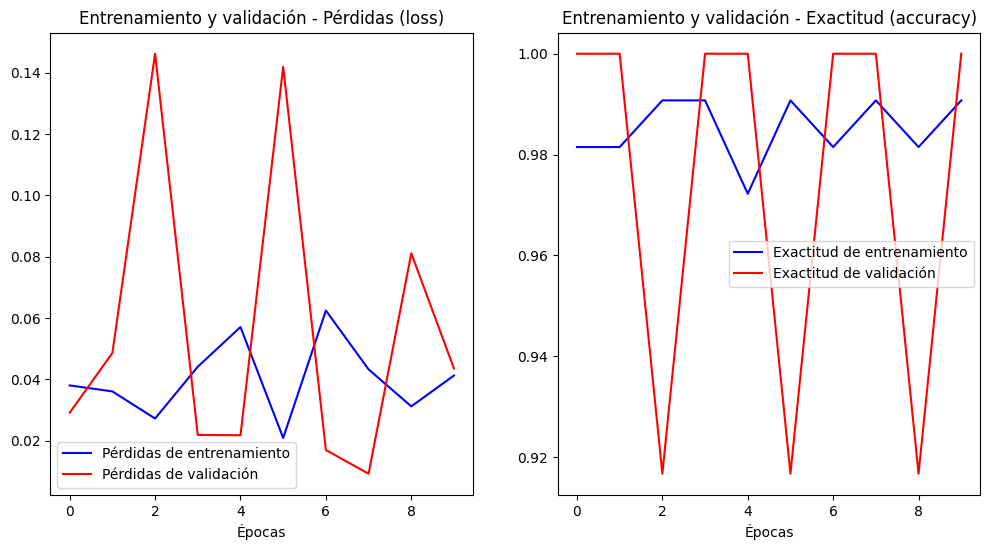
Con el aumento de épocas reducimos aun mas las perdidas. Creemos que existe overfiting dada la cantidad de picos de inestabilidad que existe en la grafica de perdidas.





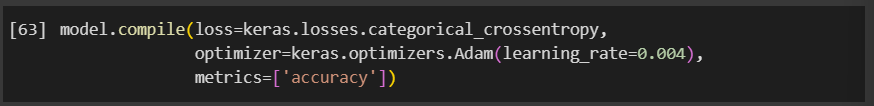
## EXPERIMENTO 9

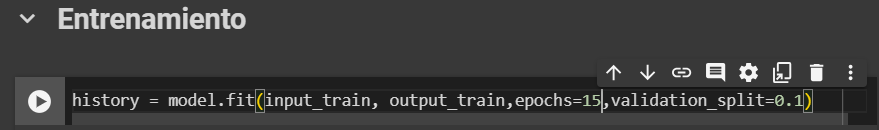
Ahora le disminuimos el numero de épocas sucede que tenemos resultados erráticos además que no logramos disminuir perdidas o aumentar la exactitud. En ambos gráficos vemos el fenómeno de overfiting puesto que existe mucha divergencia entre las curvas.  


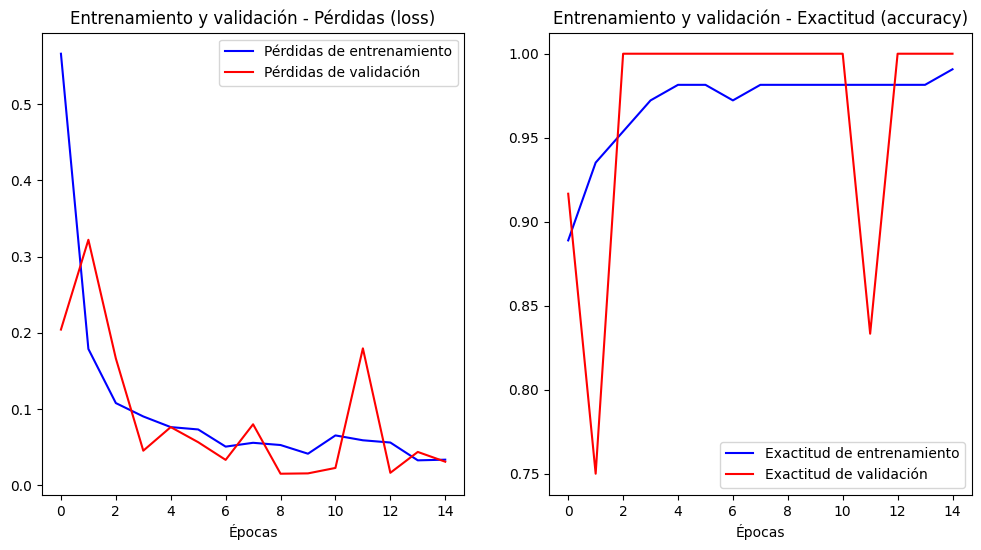


## EXPERIMENTO 10

Ahora aumentamos la tasa de aprendizaje al experimento, para el caso anterior. El resultado de reducción de perdidas es bueno y aumentamos el de exactitud de validación aunque ya no son curvas tan suaves. Podemos ver que en las curvas de de precisión hubo overfiting puesto que al aumento de épocas no le acompaña una mejora sino mas bien se estanca.







<https://colab.research.google.com/drive/1agrEOZ8eawmC91gBvMxU0GXPpJOGTy0N?usp=sharing>