

Taller 3

Predicción promedio del valor de acciones del mercado con redes

Redes neuronales

Escuela de Ingeniería de Sistemas y Computación



Emily Nuñez Ordoñez - 2240156
Victoria Andrea Volveras Parra - 2241874

Noviembre de 2025

Índice

1. Predicción promedio del valor de acciones del mercado con redes	2
1.1. Red LSTM	2
1.1.1. Gráfica mejor modelo 1 (LSTM 1.1)	2
1.1.2. Gráfica mejor modelo 2 (LSTM 2.2)	3
1.1.3. Gráfica mejor modelo 3 (LSTM 3.4)	3
1.1.4. Análisis	3
1.2. Red GRU	5
1.2.1. Gráfica mejor modelo 1 (GRU 1.2)	5
1.2.2. Gráfica mejor modelo 2 (GRU 2.2)	5
1.2.3. Gráfica mejor modelo 3 (GRU 3.4)	6
1.2.4. Análisis	6
1.3. Red de capas recurrentes apiladas	7
1.3.1. Gráfica mejor modelo 1 (SRLN 1.2)	8
1.3.2. Gráfica mejor modelo 2 (SRLN 2.2)	8
1.3.3. Gráfica mejor modelo 3 (SRLN 3.2)	8
1.3.4. Análisis	8
1.4. Conclusión	10

1. Predicción promedio del valor de acciones del mercado con redes

Se diseñaron diferentes modelos de redes neuronales recurrentes con memoria (LSTM, GRU y stacked recurrent layers network) para la predicción de valores de acciones. A continuación, se exponen los resultados del entrenamiento de los modelos diseñados.

Se presentan las gráficas de perdida (MSE) para los conjunto de entrenamiento y validación, así como las gráficas de predicción de cada red neuronal.

Se utilizó un conjunto de datos compuesto por 3202 valores de acciones diarias de Washington Real Estate Investment Trust (WRE). El 50 % de los datos se empleó en el conjunto de entrenaimiento , 25 % para validación y 25 % para testeo. Además, se incluyeron ventanas de tiempo de 30 días para el entrenamiento de las redes.

Finalmente, Se utilizó un dropout de 0.2 en todos los modelos empleados y la metrica para evaluar el rendimiento de cada red es Mean Absolute Error (MAE).

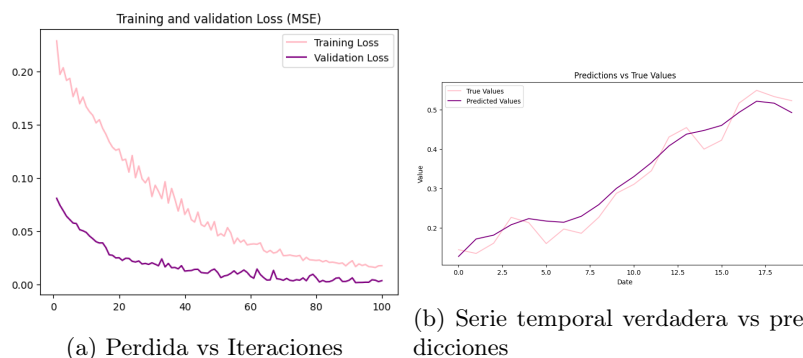
1.1. Red LSTM

Se analizó el comportamiento de la red LSTM variando los valores de batch size, sequence length, número de iteraciones, número de neuronas y habilitando o deshabilitando el recurrent dropout.

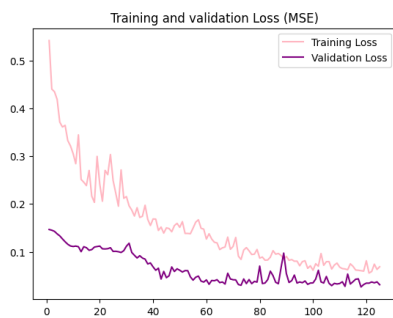
Red Neuronal	Batch size	Seq_length	Max Iter.	Recurrent dropout	MAE	MSE
LSTM 1.1	32	64	100	True	0.186	0.095
LSTM 1.2	32	128	100	True	0.210	0.113
LSTM 1.3	32	32	100	True	0.193	0.068
LSTM 1.4	32	32	100	False	0.237	0.114
LSTM 2.1	64	64	125	True	0.282	0.135
LSTM 2.2	64	128	125	True	0.271	0.113
LSTM 2.3	64	32	125	True	0.315	0.161
LSTM 2.4	64	64	125	False	0.299	0.136
LSTM 3.1	128	64	150	True	0.418	0.243
LSTM 3.2	128	128	150	True	0.374	0.240
LSTM 3.3	128	32	150	True	0.378	0.197
LSTM 3.4	128	128	150	False	0.348	0.183

Cuadro 1: Variación de hiperparámetros en una LSTM.

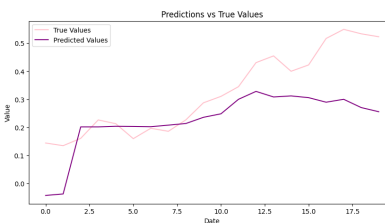
1.1.1. Gráfica mejor modelo 1 (LSTM 1.1)



1.1.2. Gráfica mejor modelo 2 (LSTM 2.2)

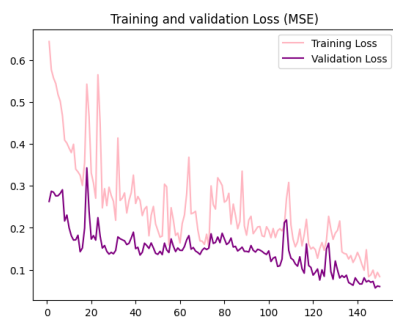


(a) Perdida vs Iteraciones

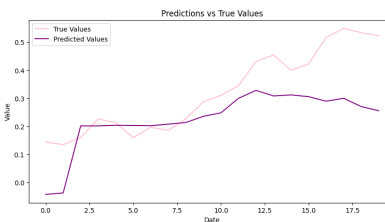


(b) Serie temporal verdadera vs predicciones

1.1.3. Gráfica mejor modelo 3 (LSTM 3.4)



(a) Perdida vs Iteraciones



(b) Serie temporal verdadera vs predicciones

1.1.4. Análisis

Primero, se realiza un análisis dividido en tres grupos:

- **Grupo 1:** configuraciones 1.1 a 1.4
- **Grupo 2:** configuraciones 2.1 a 2.4
- **Grupo 3:** configuraciones 3.1 a 3.4

Grupo 1

Al entrenar las redes con un *batch_size* de 32 y 100 *iteraciones*, el mejor resultado lo obtuvo la red LSTM 1.1 con 64 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.186 (MAE). Aunque la red LSTM 1.3 presenta un valor de MAE cercano al de la red LSTM 1.1 (0.19) y un valor MSE menor que el de LSTM 1.1, su gráfica de predicción muestra una mayor separación entre los valores reales y los predichos. Por lo tanto, la mejor opción del grupo 1 es la configuración con 64 neuronas.

Por otra parte, la red LSTM 1.4 corresponde a un modelo entrenado con 32 neuronas recurrentes sin recurrent dropout, la cual obtuvo el menor desempeño de todo el grupo, con un MAE de 0.237. Teniendo una diferencia de 0.044 con la red LSMT 1.3, siendo esta su equivalente con dropout activo.

Grupo 2

Al entrenar las redes con un *batch_size* de 64 y 125 *iteraciones*, el mejor resultado lo obtuvo la red LSTM 2.2 con 128 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.271 (MAE).

Al observar los resultados de las configuraciones de 32 neuronas podemos afirmar que el modelo necesita más complejidad para obtener mejores resultados.

Cabe aclarar que las gráficas de predicciones de este grupo estuvieron bastante distantes de los valores originales, por lo que, en general, el desempeño no fue el deseado.

Grupo 3

Al entrenar las redes con un *batch_size* de 128 y 150 *iteraciones*, el mejor resultado lo obtuvo la red LSTM 3.4 con 64 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.348 (MAE).

Es importante mencionar que en el grupo 3, se observa un patrón en la predicción de los valores de las acciones, evidente en sus respectivas gráficas. La presencia de este salto en las curvas indica que las redes LSTM no están capturando adecuadamente la dinámica real de las acciones y, en consecuencia, generan oscilaciones o explosiones numéricas.

General

Se evidenció que en cada uno de los casos tomados, para el modelo con el que se obtuvo un menor *MAE*, la cantidad de neuronas recurrentes era el doble al *batch_size*. A excepción del LSTM 3, donde para el mejor modelo la cantidad de neuronas recurrentes es igual al *batch_size*. Lo cual se puede deber a lo siguiente:

Si las neuronas son muy pocas en comparación con el *batch_size*, la red tiene una capacidad limitada para procesar toda la información diversa que llega en un solo batch. Esto puede llevar a un subajuste, ya que la red no es lo suficientemente compleja para capturar los patrones en los datos.

Si las neuronas son muchas en comparación con el *batch_size*, la capacidad del modelo supera la cantidad de información que recibe en cada actualización, por lo tanto, la red cuenta con suficientes parámetros como para aprender patrones muy específicos de cada lote, haciéndola propensa al sobreajuste, perdiendo así generalización. El recurrent dropout ayuda a mitigar esto, pero no completamente. Es decir, con un batch más grande la red parece necesitar ventanas más largas para compensar la pérdida de variabilidad en cada lote.

En batch sizes pequeños o medianos, la red recibe muestras con mayor variabilidad interna, lo que puede introducir ruido en los gradientes. Por lo tanto, el recurrent dropout, ayuda a evitar que el modelo memorice patrones específicos de cada lote. Esto favorece una mejor capacidad de generalización, ya que obliga a la red a mantener representaciones más robustas. En contraste, cuando se usan batch sizes grandes, la estimación del gradiente se vuelve mucho más estable y menos ruidosa, ya que los lotes son más representativos del conjunto total de datos. En este caso, el uso del dropout puede volverse contraproducente, ya que, al agregar ruido sobre un gradiente ya muy estable, el entrenamiento puede volverse más lento y menos eficiente. Esto puede llevar a peores mínimos o a un estancamiento del aprendizaje.

Se observa una relación inversa entre el tamaño del batch y el rendimiento, donde modelos con batch size más pequeño (32) superan a aquellos con batches más grandes. El mejor modelo fue LSTM 1.1, el cual alcanza un MAE de 0.186 y MSE de 0.095, demostrando la efectividad de esta combinación de parámetros. Esto sugiere que batches más pequeños proporcionan un efecto de regularización que beneficia la capacidad de generalización del modelo.

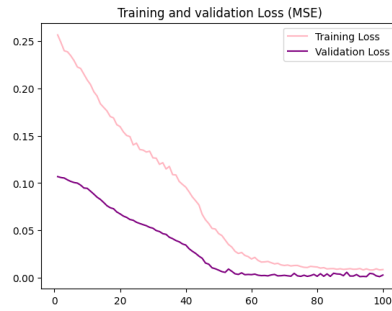
1.2. Red GRU

En la siguiente tabla, se presenta la variación de los hiperparámetros empleados en el entrenamiento de la red neuronal GRU para la predicción de valores de acciones.

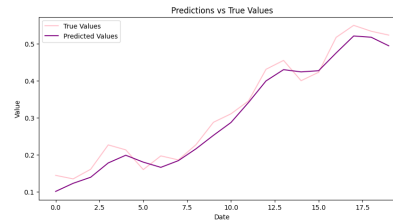
Red Neuronal	Batch size	Seq_length	Max Iter.	Recurrent dropout	MAE	MSE
GRU 1.1	32	32	100	True	0.051	0.005
GRU 1.2	32	64	100	True	0.037	0.002
GRU 1.3	32	128	100	True	0.043	0.003
GRU 1.4	32	128	100	False	0.134	0.039
GRU 2.1	64	32	125	True	0.126	0.032
GRU 2.2	64	64	125	True	0.055	0.005
GRU 2.3	64	128	125	True	0.066	0.008
GRU 2.4	64	128	125	False	0.120	0.031
GRU 3.1	128	32	150	True	0.228	0.098
GRU 3.2	128	64	150	True	0.096	0.018
GRU 3.3	128	128	150	True	0.049	0.004
GRU 3.4	128	128	150	False	0.049	0.004

Cuadro 2: Variación de hiperparámetros en una GRU.

1.2.1. Gráfica mejor modelo 1 (GRU 1.2)

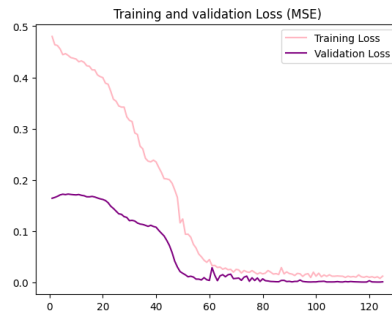


(a) Perdida vs Iteraciones

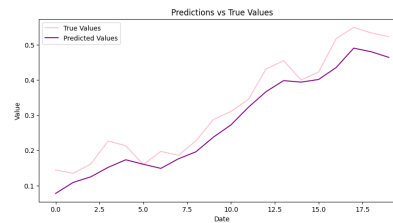


(b) Serie temporal verdadera vs predicciones

1.2.2. Gráfica mejor modelo 2 (GRU 2.2)

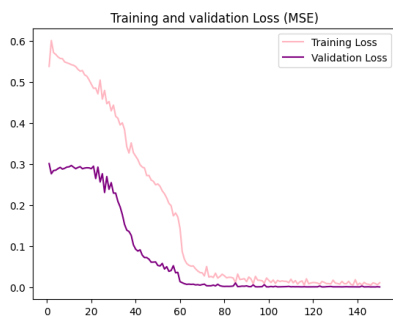


(a) Perdida vs Iteraciones

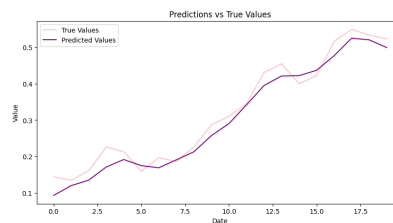


(b) Serie temporal verdadera vs predicciones

1.2.3. Gráfica mejor modelo 3 (GRU 3.4)



(a) Pérdida vs Iteraciones



(b) Serie temporal verdadera vs predicciones

1.2.4. Análisis

Primero, se realiza un análisis dividido en tres grupos:

- **Grupo 1:** configuraciones 1.1 a 1.4
- **Grupo 2:** configuraciones 2.1 a 2.4
- **Grupo 3:** configuraciones 3.1 a 3.4

Grupo 1

Al entrenar las redes con un *batch_size* de 32 y 100 *iteraciones*, la red con mejor desempeño fue la red GRU 1.2 con 64 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.037 (MAE).

Al comparar los resultados de las diferentes cantidades de neuronas recurrentes analizadas, se puede afirmar que los modelos del grupo 1 necesitan cierto nivel de complejidad para tener un desempeño adecuado. En este caso, cuando la cantidad de neuronas es menor que 64, la pérdida aumenta. Asimismo, cuando la cantidad de neuronas aumenta a 128, la pérdida presenta el mismo comportamiento.

El modelo entrenado con 128 neuronas sin *dropout* obtuvo el desempeño más bajo del grupo, siendo casi cuatro veces peor que su equivalente con *dropout* activo (GRU 1.3), lo que constituye un ejemplo claro de que, se necesita cierta variabilidad en la información, pero al mismo tiempo, una arquitectura que permita procesar y aprender correctamente las ventanas de datos sin caer en sobreajuste o subajuste. Sin embargo, los valores de MAE y las gráficas de predicciones obtenidas en este grupo son buenos indicios de desempeño.

Grupo 2

Al entrenar las redes con un *batch_size* de 64 y 125 *iteraciones*, el mejor resultado lo obtuvo la red GRU 2.2 con 64 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.055 (MAE).

Al igual que en el grupo 1, los modelos GRU de este grupo no requieren una arquitectura ni demasiado sencilla ni demasiado compleja. Aunque el tamaño del *batch* aumentó, se observa la misma situación descrita anteriormente: las redes con 32 o 128 neuronas tienden a incrementar la pérdida.

Grupo 3

Al entrenar las redes con un *batch_size* de 128 y 150 *iteraciones*, el mejor resultado lo obtuvo la red GRU 3.4 con 64 neuronas recurrentes. Esta configuración alcanzó un valor de pérdida de 0.348 (MAE).

Para este grupo, al duplicar el tamaño del *batch*, se requiere una arquitectura capaz de aprender más información y generalizar de mejor manera; por lo tanto, es necesario un modelo más complejo. Esto se deduce del comportamiento de las redes GRU 3.1 y 3.2, que tienen 32 y 64 neuronas respectivamente y presentan valores de pérdida más altos. Dicho comportamiento es coherente con el modelo que mejor se desempeñó en este grupo (GRU 3.4).

General

Al analizar los tres grupos de modelos GRU, se identificó una relación entre la cantidad de neuronas y el tamaño del *batch*: si las neuronas son muy pocas en comparación con el *batch_size*, la red tiene una capacidad limitada para procesar toda la información que llega en un solo lote. Por el contrario, si las neuronas son muchas en comparación con el *batch_size*, la capacidad del modelo supera la cantidad de información que recibe en cada actualización; por lo tanto, la red cuenta con suficientes parámetros para aprender patrones muy específicos de cada lote, haciéndola propensa al sobreajuste y perdiendo capacidad de generalización. Asimismo, el *recurrent dropout* ayuda a mitigar este comportamiento, pero no completamente. Es decir, con un *batch* más grande, la red parece necesitar ventanas más largas para compensar la pérdida de variabilidad en cada lote.

En este grupo, el modelo requiere cierto grado de variabilidad de la información para lograr un aprendizaje óptimo (decidir qué aprender y qué olvidar); por ello, las redes que mejor desempeño obtuvieron fueron aquellas con un *batch_size* similar a la cantidad de neuronas.

En general, estas redes evidencian cómo las GRU son eficientes para tareas que requieren aprendizaje a largo plazo. Además, se observa una relación inversa entre el tamaño del *batch* y el rendimiento, donde modelos con *batch size* más pequeño (32) superan a aquellos con *batches* más grandes. El mejor modelo fue GRU 1.2, el cual alcanzó un MAE de 0.037 y un MSE de 0.002, demostrando la efectividad de esta combinación de parámetros. Esto sugiere que *batches* más pequeños proporcionan un efecto de regularización que beneficia la capacidad de generalización del modelo.

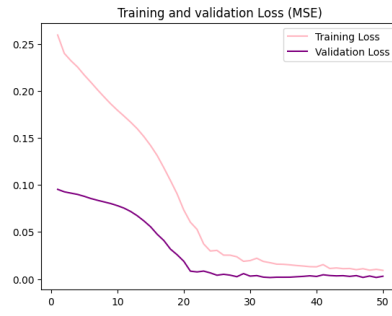
1.3. Red de capas recurrentes apiladas

En la siguiente tabla, se presenta la variación de los hiperparámetros empleados en el entrenamiento de la red neuronal de capas recurrentes apiladas para la predicción de valores de acciones. Para esta red se usó una arquitectura GRU con dos capas apiladas.

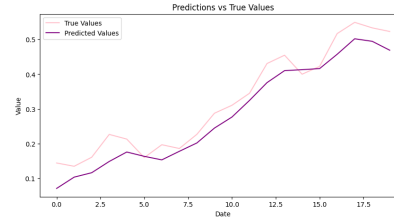
Red Neuronal	Batch size	# Layer 1	# Layer 2	Max Iter.	MAE	MSE
SRLN 1.1	32	64	32	50	0.161	0.057
SRLN 1.2	32	128	32	50	0.141	0.049
SRLN 1.3	32	128	64	50	0.165	0.053
SRLN 2.1	64	64	32	100	0.124	0.031
SRLN 2.2	64	128	32	100	0.109	0.040
SRLN 2.3	64	128	64	100	0.130	0.040
SRLN 3.1	128	64	32	150	0.089	0.016
SRLN 3.2	128	128	32	150	0.053	0.007
SRLN 3.3	128	128	64	150	0.055	0.005

Cuadro 3: Variación de hiperparámetros en una red de capas recurrentes apiladas.

1.3.1. Gráfica mejor modelo 1 (SRLN 1.2)

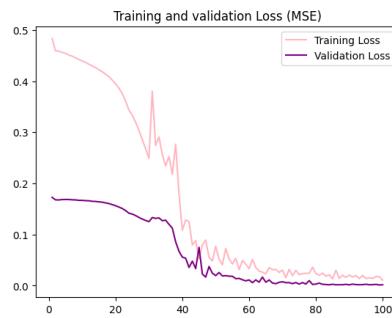


(a) Perdida vs Iteraciones

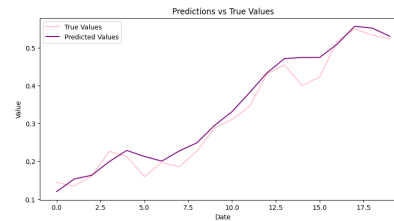


(b) Serie temporal verdadera vs predicciones

1.3.2. Gráfica mejor modelo 2 (SRLN 2.2)

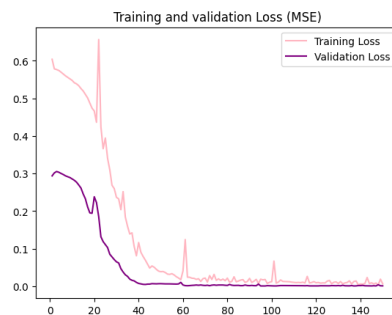


(a) Perdida vs Iteraciones

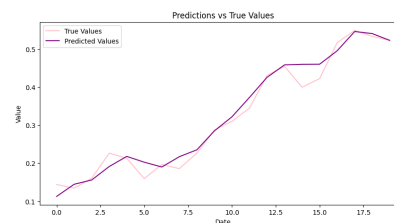


(b) Serie temporal verdadera vs predicciones

1.3.3. Gráfica mejor modelo 3 (SRLN 3.2)



(a) Perdida vs Iteraciones



(b) Serie temporal verdadera vs predicciones

1.3.4. Análisis

Primero, se realiza un análisis dividido en tres grupos:

- **Grupo 1:** configuraciones 1.1 a 1.3
- **Grupo 2:** configuraciones 2.1 a 2.3
- **Grupo 3:** configuraciones 3.1 a 3.3

Grupo 1

Al entrenar las redes con un *batch_size* de 32 y 50 *iteraciones*, el mejor resultado lo obtuvo la red SRLN 1.2 con dos capas, con una cantidad de neuronas de 128 para la primera capa y 32 para la segunda. Esta configuración alcanzó un valor de pérdida de 0.141 (MAE).

Los resultados obtenidos indican que un incremento moderado en la capacidad del modelo mejora su habilidad para capturar patrones relevantes sin introducir una complejidad excesiva que afecte la estabilidad del entrenamiento. En la red SRLN 1.3, la segunda capa es relativamente más grande respecto a la cantidad de información útil filtrada por la primera capa. Esto puede hacer que la segunda capa amplifique ruido o redundancia, afectando la estabilidad del entrenamiento.

Grupo 2

Al entrenar las redes con un *batch_size* de 64 y 100 *iteraciones*, el mejor resultado lo obtuvo la red SRLN 2.2 con dos capas, con una cantidad de neuronas de 128 para la primera capa y 32 para la segunda. Esta configuración alcanzó un valor de pérdida de 0.109 (MAE).

Para este grupo se presentó algo parecido que en el anterior, donde se puede evidenciar que un incremento moderado en la capacidad del modelo permite capturar mejor las relaciones presentes en los datos sin introducir una complejidad excesiva. La configuración SRLN 2.3, con una segunda capa más grande (64 neuronas), muestra un rendimiento inferior. Lo que indica que para el tamaño del *batch* utilizado y el nivel de información procesado por la primera capa, una segunda capa demasiado amplia puede introducir redundancia, aumentar la varianza de los gradientes y dificultar el proceso de optimización.

Grupo 3

Al entrenar las redes con un *batch_size* de 64 y 150 *iteraciones*, el mejor resultado lo obtuvo la red SRLN 3.2 con dos capas, con una cantidad de neuronas de 128 para la primera capa y 32 para la segunda. Esta configuración alcanzó un valor de pérdida de 0.053 (MAE).

La red SRLN 3.1 obtiene el peor rendimiento dentro del grupo. Este desempeño puede deberse a que una primera capa relativamente pequeña puede limitar la capacidad de la red para extraer representaciones apropiadas de los datos, especialmente cuando se trabaja con *batches* grandes que tienden a suavizar el gradiente y requieren modelos más complejos para capturar la variabilidad.

La red SRLN 3.3 presenta un desempeño muy cercano al de SRLN 3.2, con un MAE de 0.055. El ligero empeoramiento respecto a la mejor configuración puede deberse a que la *segunda capa*, al ser más grande, introduce mayor complejidad de parámetros sin proveer un beneficio significativo en representación. Esto puede indicar que con *batches* grandes, este aumento de capacidad puede hacer que el modelo se ajuste más a patrones específicos de los datos del *batch*, reduciendo su generalización.

General

Se evidencia que aumentar moderadamente el número de neuronas en la primera capa recurrente tiende a mejorar el rendimiento en todos los grupos. Pasar de 64 a 128 neuronas en la *primera capa* resulta beneficioso cuando el modelo procesa *batches* más grandes y gradientes más estables. Sin embargo, incrementar de manera equivalente la capacidad de la *segunda capa* no genera beneficios comparables. Cuando la *segunda capa* se hace demasiado grande respecto al volumen de información ya procesado por la primera, puede provocar que la red incorpore redundancia, lo cual introduce mayor varianza en los gradientes y dificulta la optimización. Esto explica por qué las configuraciones 128-32 superan a las de 128-64, pese a tener menor número total de parámetros.

En relación con el *batch size*, a medida que el *batch size* aumenta, las redes requieren mayor capacidad y

más *iteraciones* para poder converger adecuadamente. Con *batches* pequeños, el gradiente es ruidoso y actúa como regularizador natural, por lo que una red demasiado grande puede volverse inestable o sobreajustar rápidamente. En cambio, con *batches* grandes, donde el gradiente se estabiliza y se asemeja más al gradiente real del dataset, el modelo necesita una arquitectura más compleja para capturar relaciones más finas en los datos. Por esto, los mejores resultados del estudio aparecen cuando $batchsize = 128$ es acompañado de suficientes *iteraciones* (150) y una *primera capa* amplia (128 neuronas).

El número de *iteraciones* también influye de manera consistente, ya que con pocas *iteraciones* (50), la red tiene tiempo limitado para ajustar parámetros en arquitecturas más complejas, lo que favorece configuraciones moderadas. Conforme se incrementa esta cantidad, las redes con mayor complejidad logran aprovechar mejor su estructura, convergen a mejores mínimos y capturan patrones más complejos. La combinación de alta capacidad, *batch* grande y más *iteraciones* es precisamente lo que permite el resultado más bajo ($MAE = 0.053$) en el modelo SRLN 3.2.

1.4. Conclusión

El modelo que obtuvo el mejor desempeño fue la red neuronal **GRU 1.2** con un valor MAE de 0.037 y MSE de 0.002.