Modelo simplificado

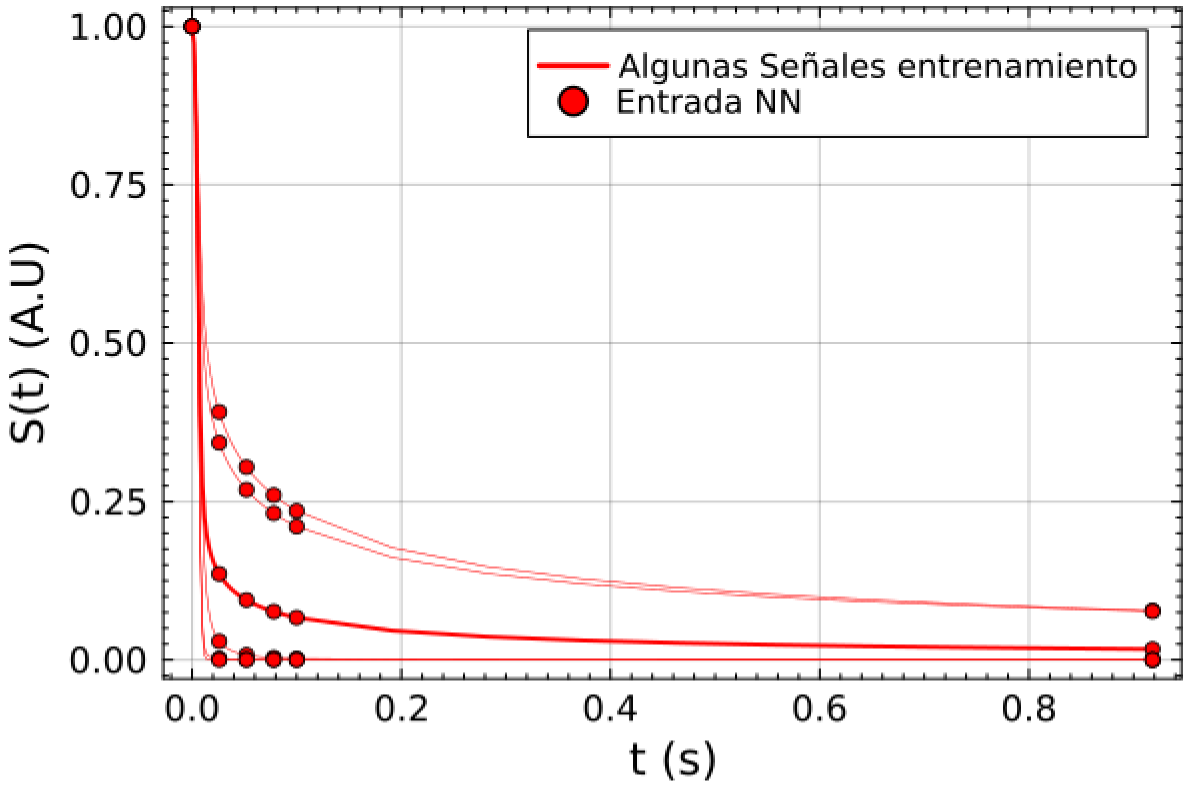
- Una de las ideas era no pasar ni por NODE, ni reducción de dimensionalidad ni nada. Simplemente usar puntos de las señales de Hahn para obtener los parámetros de las distribuciones de probabilidad subyacentes.
- Esta simplificación parecía no tener sentido porque por lo menos para que la red neuronal funcione con la entrada en dimensionalidad reducida se necesitaban muchos datos de la señal.

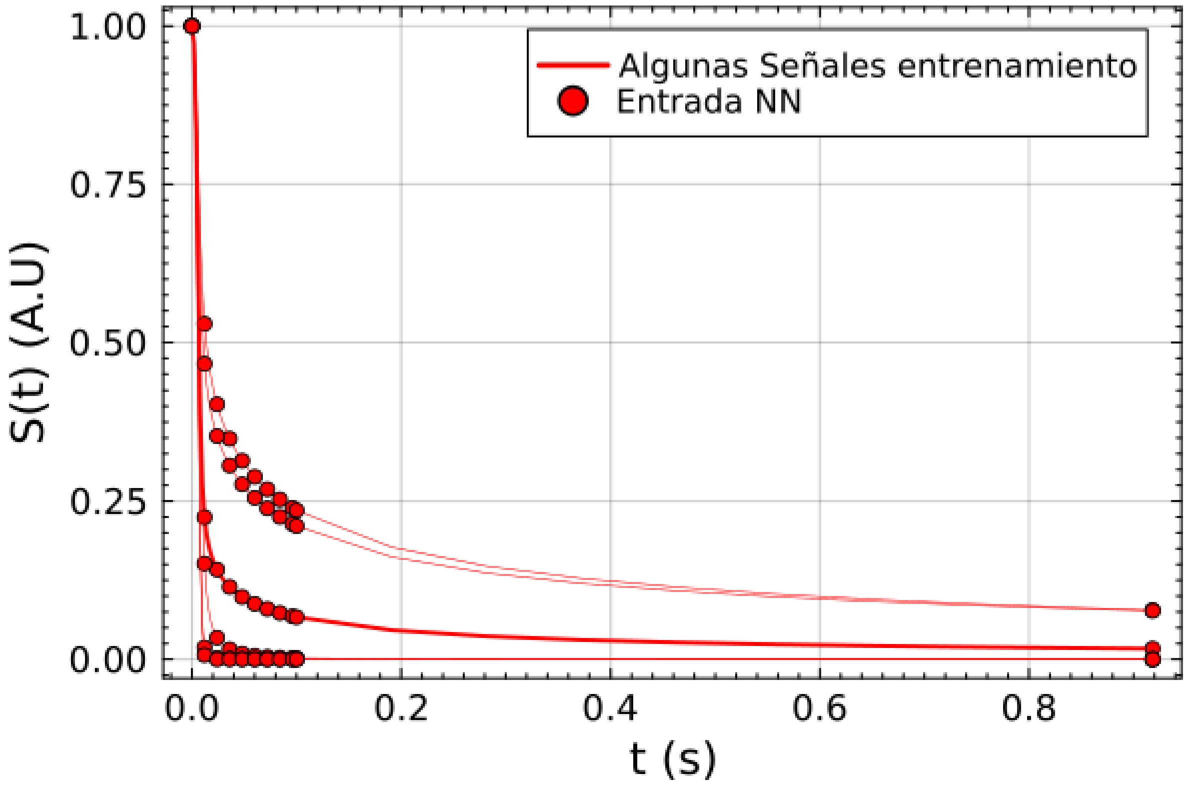
Si comenzamos con esta idea, una pregunta sería: ¿qué tiempos tomar para medir la señal?

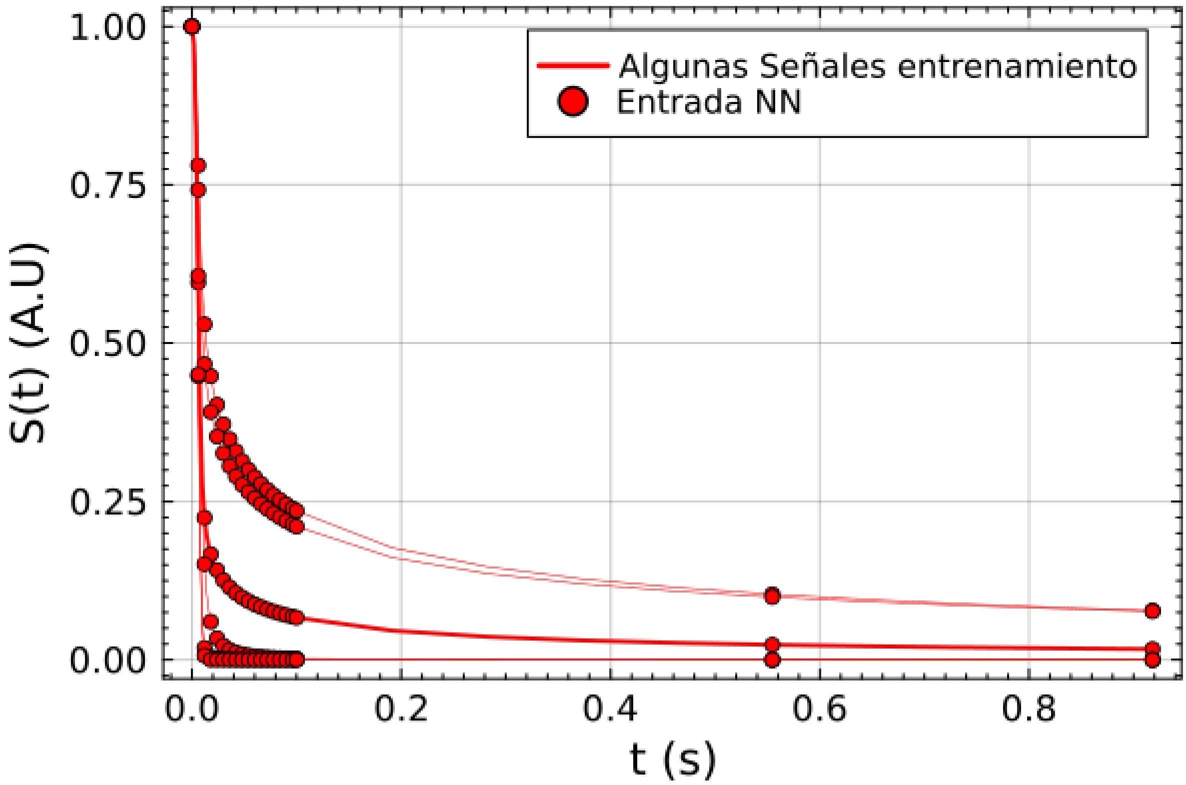
Esto es complicado porque todas las señales son muy distintas entre sí; algunas, al llegar a 0.1 s, ya decayeron completamente, otras aún no, y otras ni siquiera comienzan a mostrar el comportamiento de decaimiento exponencial. Esto depende mucho de los parámetros de las distribuciones de tamaño subyacentes l_{cm} (tamaño de correlación medio) y σ (desviación estándar). En este caso, como no queremos perdernos información de ninguna señal, hacemos lo siguiente: si vamos a tomar N puntos de la señal para hacer de entrada a la red, el 90% va a pertenecer al intervalo de tiempo $t \in (0, 0.1)$ s. El resto al intervalo $t \in (0.1, 1)$ s.

Al final tenemos datos de entrada como las que se muestran en las imágenes para $N = 5,\ 10,\ 20$. Estos son para algunas señales de nuestro dataset

```
using PlutoUI
```





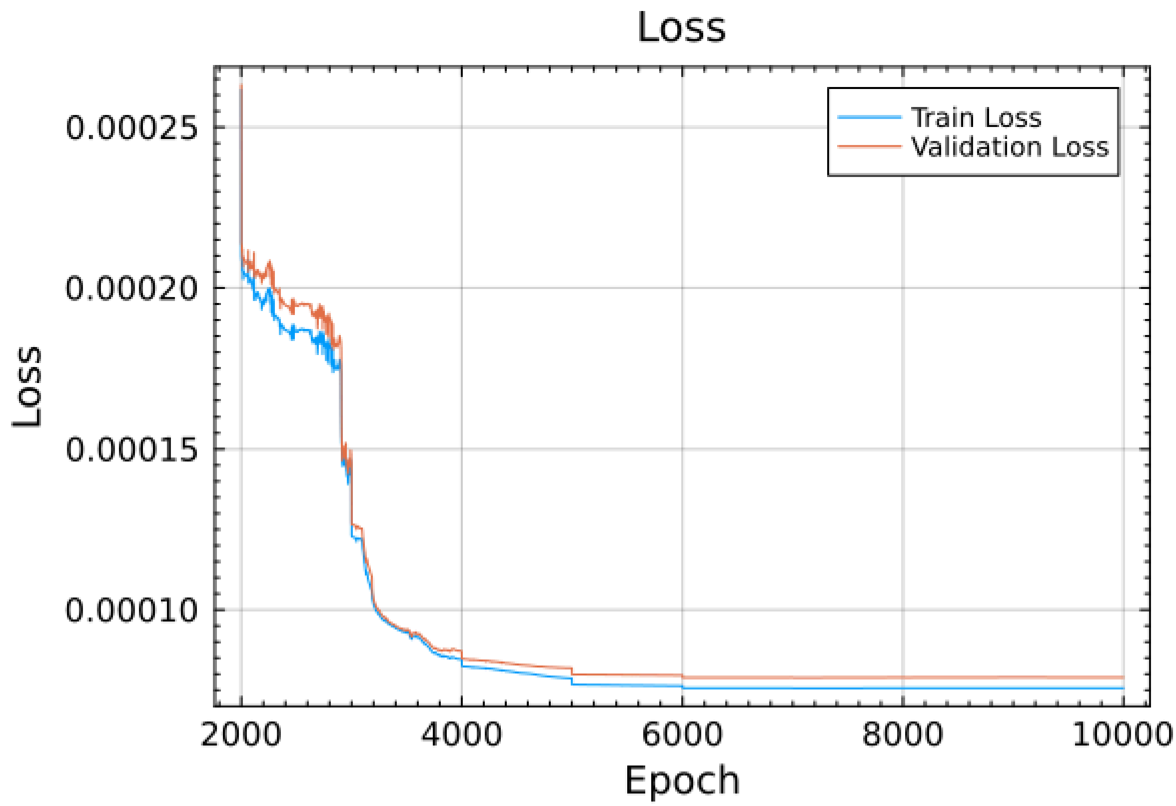


Haciendo una exploración de hiperparámetros donde nuestras redes van desde N hasta un par $(l_{cm},\ \sigma)$ donde se explora, tamaño de la arquitecura, numero de datos de entrada, lambda de regularización L2, y cantidad de señales en el dataset (dejé solo las que tienen todo el dataset) tenemos:

ID	Arq	N	$Lambd$	$MSETrain$	$MSEVal$	$MSETest$	Nu
7	$[6,16,32,16,2]$	5	0.0	0.06677	0.07003	0.06706	55
8	$[6,32,64,16,2]$	5	0.0	0.0793	0.08181	0.07917	55
9	$[6,16,32,16,8,2]$	5	0.0	0.09322	0.09717	0.09262	55
10	$[6,32,64,16,8,2]$	5	0.0	0.09323	0.09738	0.09224	55
11	$[6,30,25,20,15,10,2]$	5	0.0	0.05589	0.05868	0.05624	55
12	$[6,32,64,32,16,2]$	5	0.0	0.09216	0.09611	0.09142	55
19	$[6,16,32,16,2]$	5	0.1	0.9587	0.9504	0.961	55
20	$[6,32,64,16,2]$	5	0.1	0.9587	0.9504	0.961	55
21	$[6,16,32,16,8,2]$	5	0.1	0.9859	0.978	0.9881	55
22	$[6,32,64,16,8,2]$	5	0.1	0.9859	0.978	0.9881	55
23	$[6,30,25,20,15,10,2]$	5	0.1	1.021	1.014	1.023	55
24	$[6,32,64,32,16,2]$	5	0.1	0.9859	0.978	0.9881	55
31	$[11,16,32,16,2]$	10	0.0	0.003579	0.003672	0.003597	55
32	$[11,32,64,16,2]$	10	0.0	0.003017	0.003084	0.002987	55
33	$[11,16,32,16,8,2]$	10	0.0	0.001993	0.002094	0.002069	55
34	$[11,32,64,16,8,2]$	10	0.0	0.005792	0.005746	0.00587	55
35	$[11,30,25,20,15,10,2]$	10	0.0	0.003659	0.003815	0.003662	55
36	$[11,32,64,32,16,2]$	10	0.0	0.005	0.005135	0.005094	55
43	$[11,16,32,16,2]$	10	0.1	0.8009	0.7974	0.8038	55
45	$[11,16,32,16,8,2]$	10	0.1	0.8281	0.8251	0.8309	55
46	$[11,32,64,16,8,2]$	10	0.1	0.8281	0.8251	0.8309	55
47	$[11,30,25,20,15,10,2]$	10	0.1	0.8633	0.8607	0.8661	55
48	$[11,32,64,32,16,2]$	10	0.1	0.8281	0.8251	0.8309	55
55	$[21,16,32,16,2]$	20	0.0	0.0003108	0.0003169	0.0003186	55
56	$[21,32,64,16,2]$	20	0.0	0.0002175	0.0002201	0.0002262	55
57	$[21,16,32,16,8,2]$	20	0.0	0.0001325	0.0001383	0.0001379	55
58	$[21,32,64,16,8,2]$	20	0.0	0.0001117	0.0001134	0.0001144	55
59	$[21,30,25,20,15,10,2]$	20	0.0	0.0001084	0.0001144	0.0001127	55
60	$[21,32,64,32,16,2]$	20	0.0	$7.564e-5$	$7.897e-5$	$7.734e-5$	55
67	$[21,16,32,16,2]$	20	0.1	0.7518	0.7494	0.7542	55
68	$[21,32,64,16,2]$	20	0.1	0.7518	0.7494	0.7542	55
69	$[21,16,32,16,8,2]$	20	0.1	0.7756	0.7738	0.7778	55
70	$[21,32,64,16,8,2]$	20	0.1	0.7756	0.7738	0.7778	55
71	$[21,30,25,20,15,10,2]$	20	0.1	0.8079	0.8062	0.8099	55
72	$[21,32,64,32,16,2]$	20	0.1	0.7756	0.7738	0.7778	55

Como se ve ciertas arquitecturas con 20 puntos de entrada (mas el 1 que siempre es conocido) tienen valores de MSE bajos, tanto que la arquitectura 60, el MSE tanto para entrenamiento, validación y test es del orden de las mejores redes obtenidas al hacer una red que reciba únicamente las 3 componentes principales luego de reducir dimensionalidad.

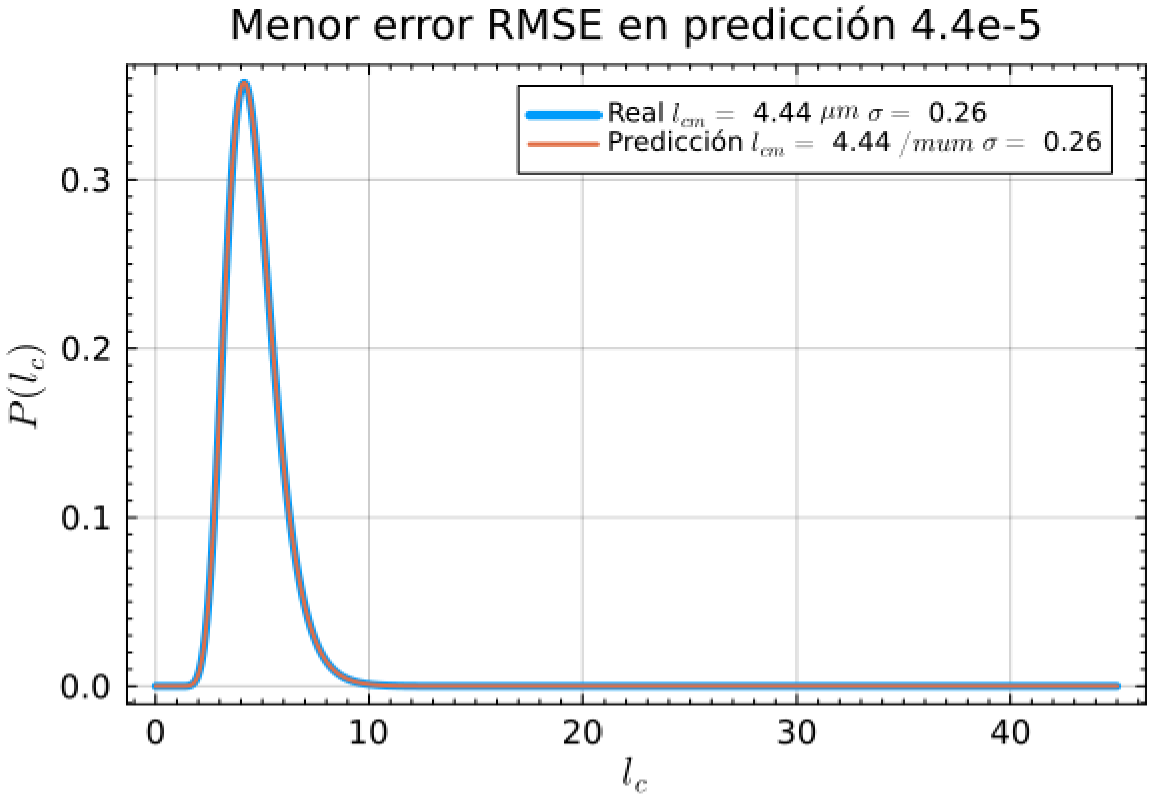
El loss MSE en función de las épocas de entrenamiento es este

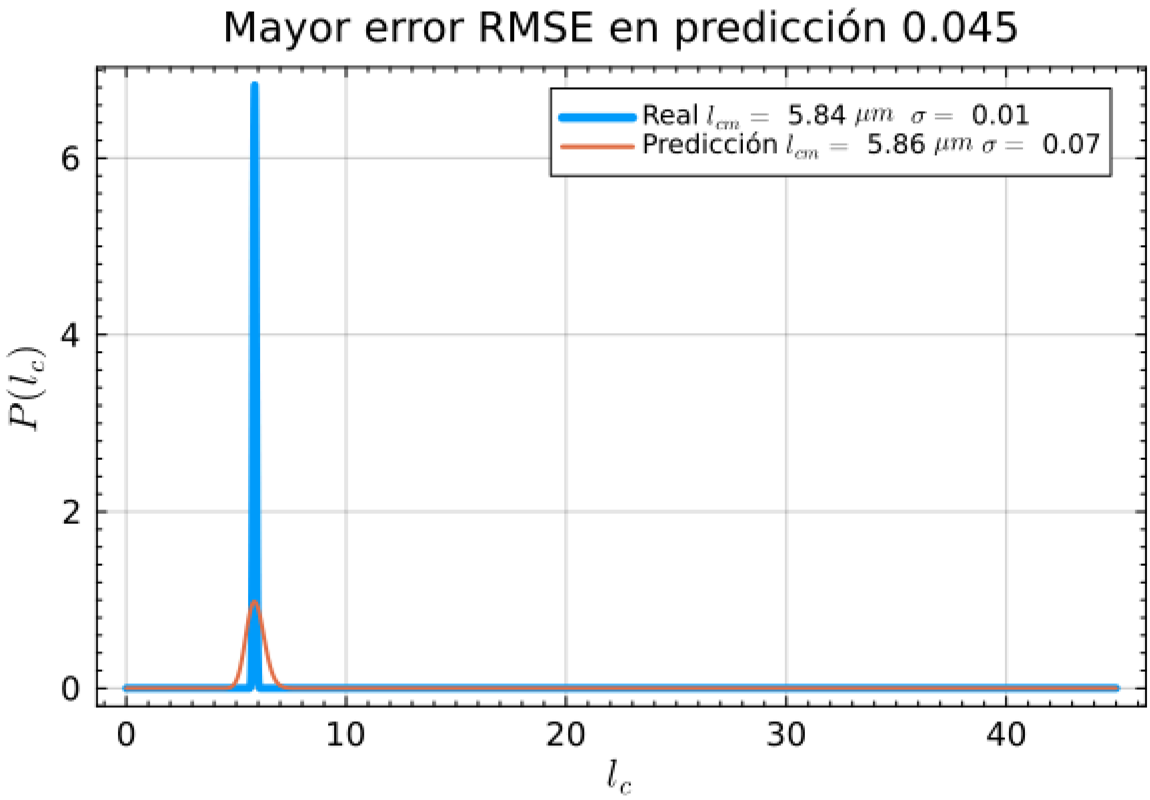


Los errores Mean Absolute Error de todo los puntos y Root Mean Absolut error de todas las predicciones son:

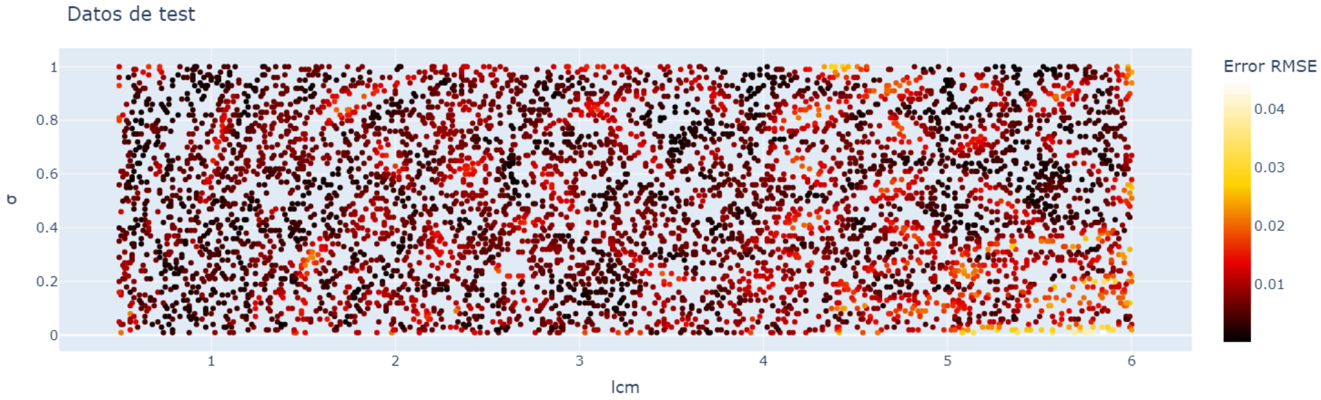
- MAE Train: 0.0065
- RMAE Train: 0.0087
- MAE Valid: 0.0067
- RMAE Valid: 0.0089

Los mejores y peores errores en la predicción de parámetros





Si observamos el error en la predicción de los parámetros de manera individual utilizando el error RMSE en los datos de test, obtenemos el siguiente mapa de calor. Esta grilla tiene aproximadamente 5000 predicciones de los parámetros en la grilla de l_{cm} y σ.



Vemos que el máximo en datos de señales con las que la red no fue entrenada el error no pasa el 5% con esta métrica de error, lo cual es suficientemente bueno para lo buscado.

Teniendo esto en mente se puede hacer un bypass a la reducción de dimensionalidad, porque podemos hacer que a partir de suponiendo 5 puntos una red (NODE u otra) nos de los restantes para poder entrar a la red neruonal entrenada y predecir los parámetros.