Redes Neuronales para la clasificación de textos

Bocco, Alessio (boccoalessio@gmail.com)
Maldonado, Florencia (maldonado.florenciam@gmail.com)
Ramello de la Vega, Agustín (a.ramellodelavega@gmail.com)
Rubio, Ariel (arubio@novix.com)
Torres, Gonzalo (gonza.nicolastorres@gmail.com)

21 March, 2021

Contents

1 Introducción

El presente informe corresponde al práctico de Aprendizaje Profundo de la Diplomatura en Ciencia de Datos (UNC) realizado por el grupo 3.El práctico consistió en la clasificación de títulos de productos vendidos por Mercado Libre en distintas categorías. A continuación se hace una breve descripción del dataset de train. El mismo consta de 6119100 observaciones y cuatro variables. Sólo las variables `tittle` y `category` serán utilizadas ya que se han filtrado los títulos en idioma español y no se considerará la calidad de la observación en el entrenamiento de los modelos.

language	label_quality	title	category
spanish	reliable	Bateria Completa 5 Cuerpos Excelente	DRUMS
spanish	reliable	Cuaderno Anotador Espiral Ben 10 3d Original Angel Estrada	NOTEBOO
spanish	reliable	Fifa18 Ps4 Disco Fisico	VIDEO_GA
spanish	reliable	Botines Futbol adidas Messi 15.4 Cesped Hombre	FOOTBALL
spanish	reliable	Chops Sublimados - Nagual	DRINKING_
spanish	reliable	Cómoda Toillete Escritorio Y Mesas De Luz Reina Ana Roble	DRAWERS
spanish	reliable	Casio Fx 19 - Calculadora Cientifica	CALCULAT
spanish	reliable	Alfombra De Entrada Con Diseño, 60x40 - Rulo De Pvc	CARPETS
spanish	reliable	Singer Hd205c Semi Industrial Motor 1100ppm Cabezal	SEWING_M
spanish	reliable	Ventilador De Techo Centrex Madera 4 Palas Con 2 Luces	FANS

Si bien no se realizó un análisis exploratorio de los datos, las categorías se encuentran desbalanceadas. En la Figura 1 se muestra una histograma con la cantidad de observaciones por categoría.

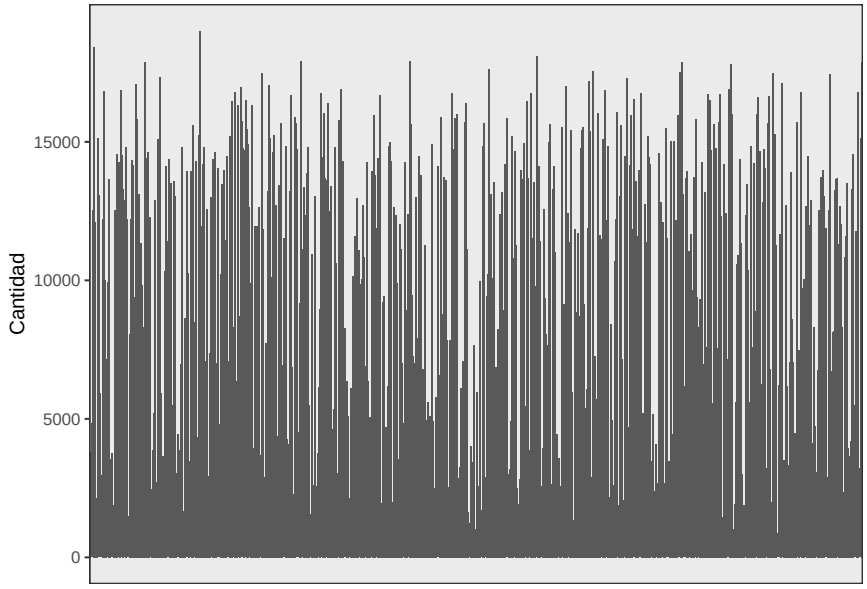

Figure 1: Histograma de observaciones por categoría

Analizar el dataset no forma parte del práctico pero sería interesante evaluar cuál es el impacto del mismo a la hora de entrenar la red. Sin dudas que al entrenar con menos recursos y utilizar una fracción del dataset algunas categorías pueden perderse. La métrica balanced accuracy contempla este problema por lo que es una muy buena elección.

2 Preprocesamiento

Dado que el curso está centrado en redes neuronales, el procesamiento de datos de texto no fue estudiado en profundidad. Hubiese sido interesante conocer trabajar y manipular este tipo de datos sobretodo para aplicar redes más avanzadas. El preprocesamiento incluyó la eliminación de tags, signos de puntuación, espacios en blanco, y caracteres numéricos. También se eliminaron conjunciones y preposiciones al igual que palabras muy cortas. Por último se transformaron las palabras en un índices de un diccionario.

3 Redes neuronales

La clasificación se realizó usando dos redes, por un lado, Multilayer Perceptron, y por otro, una red más avanzada, CNN. A continuación se muestran los resultados de los distintos experimentos realizados para cada una de las redes.

3.1 Multilayer Perceptron (MLP)

Con esta red se realizaron 6 experimentos. Los detalles de cada uno de ellos se encuentran dentro de la carpeta `mlruns` y la subcarpeta `mlp` que acompañan el presente reporte. Algunos experimentos se corrieron en Nabucodonosor mientras que otros en Google Colab.

```
## ./mlruns/mlp/1/383d80f436714dd0a659262f075d0260
## ./mlruns/mlp/1/4ad118cb0800403c99a2f46253f17b56
## ./mlruns/mlp/1/64f1201987904391828a5eb360918881
## ./mlruns/mlp/1/8eb38861d3bd4471930281382167f512
## ./mlruns/mlp/1/c1021d64981f4197b4a3531049ca0746
## ./mlruns/mlp/1/dd411f7374a540c0aaa041b8983d1a77
```

3.1.1 Experimento 1

El primer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
383d80f436714dd0a659262f075d0260	embeddings	./data/SBW-vectors-300-min5.txt.gz
383d80f436714dd0a659262f075d0260	embeddings_size	300
383d80f436714dd0a659262f075d0260	epochs	150
383d80f436714dd0a659262f075d0260	freeze_embedding	TRUE
383d80f436714dd0a659262f075d0260	hidden1_size	128
383d80f436714dd0a659262f075d0260	hidden2_size	128
383d80f436714dd0a659262f075d0260	model_name	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set de test y validación. La Figura 2 muestra la evolución de la métrica a lo largo de 150 épocas. Cabe mencionar que se utilizó un dataset con 1 millón de observaciones para evitar problemas con la memoria RAM en Google Colab.

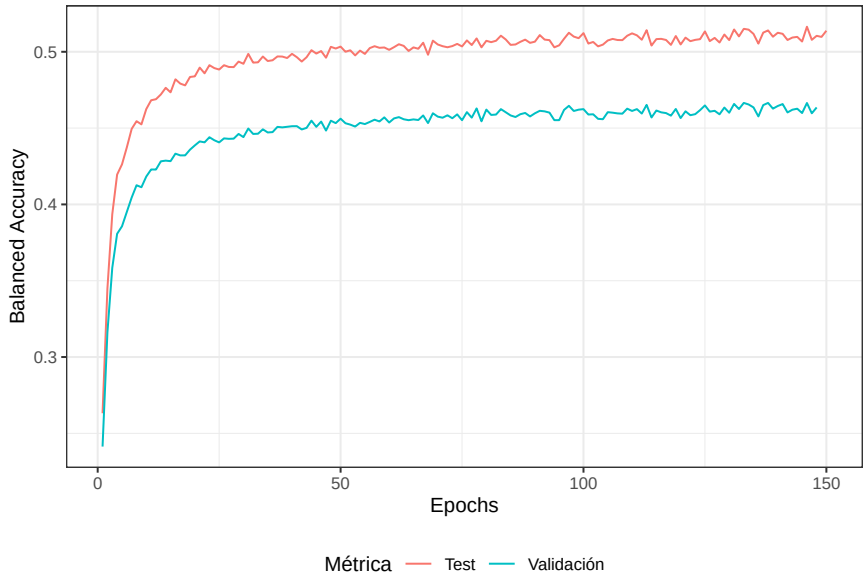

Figure 2: Resultado del Experimento 1

La Figura 2 muestra el resultado de la función de pérdida para los sets de validación y train.

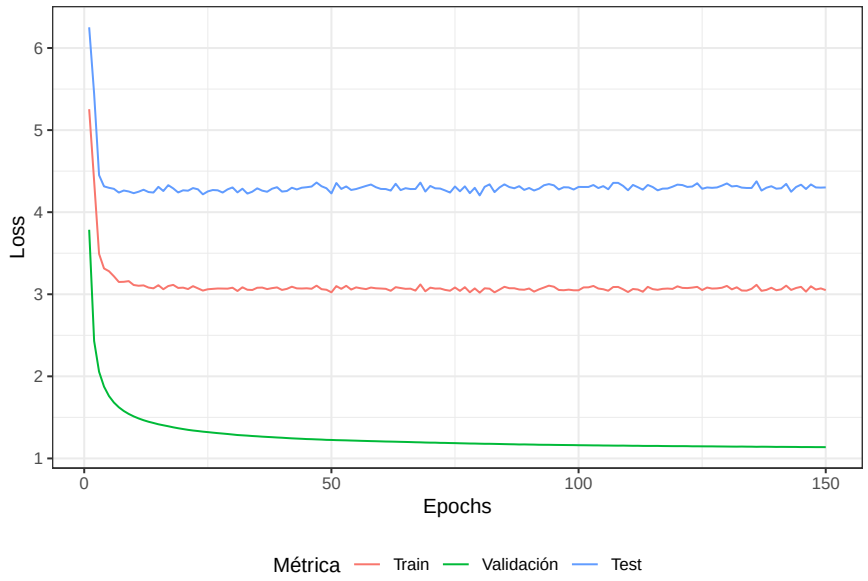

Figure 3: Resultado de la función de pérdida del Experimento 1

3.1.2 Experimento 2

El segundo experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
4ad118cb0800403c99a2f46253f17b56	dropout	0.3
4ad118cb0800403c99a2f46253f17b56	embeddings	./data/SBW-vectors-300-min5.txt.gz
4ad118cb0800403c99a2f46253f17b56	embeddings_size	300
4ad118cb0800403c99a2f46253f17b56	epochs	3
4ad118cb0800403c99a2f46253f17b56	hidden_layers	[256,128]
4ad118cb0800403c99a2f46253f17b56	model_type	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 4 muestra la evolución de la métrica a lo largo de 3 épocas.

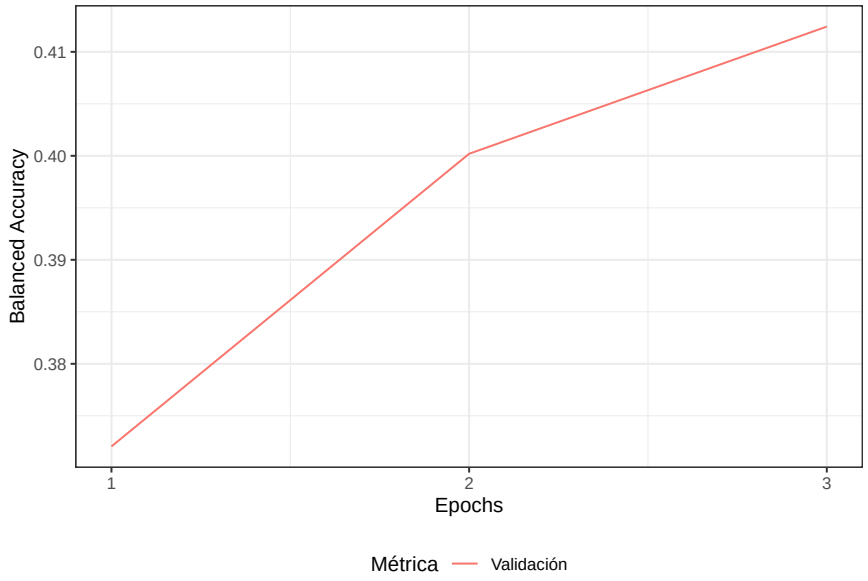

Figure 4: Resultado del Experimento 2

La Figura 5 muestra el resultado de la función de pérdida para los sets de validación y train.

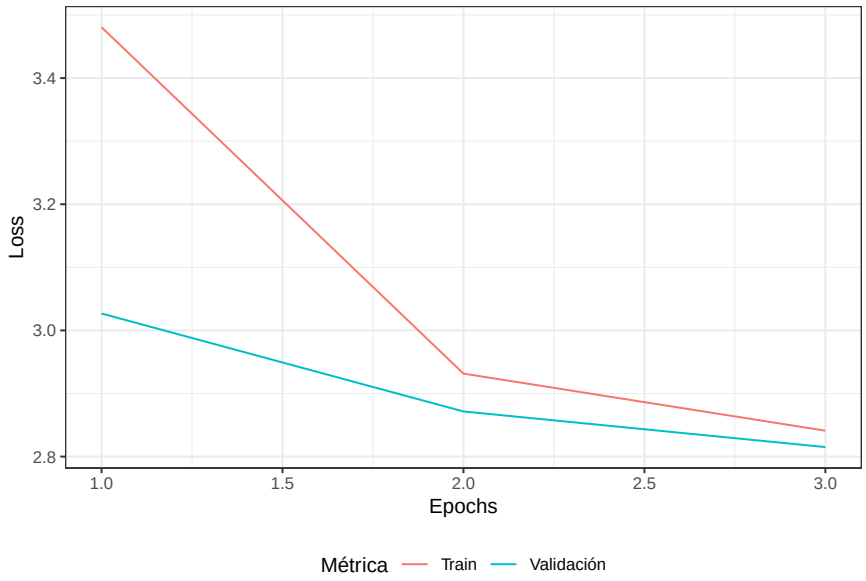

Figure 5: Resultado de la función de pérdida del Experimento 2

3.1.3 Experimento 3

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
64f1201987904391828a5eb360918881	dropout	0.5
64f1201987904391828a5eb360918881	embeddings	./data/SBW-vectors-300-min5.txt.gz
64f1201987904391828a5eb360918881	embeddings_size	300
64f1201987904391828a5eb360918881	epochs	5
64f1201987904391828a5eb360918881	hidden_layers	[256,128]
64f1201987904391828a5eb360918881	model_type	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 6 muestra la evolución de la métrica a lo largo de 5 épocas.

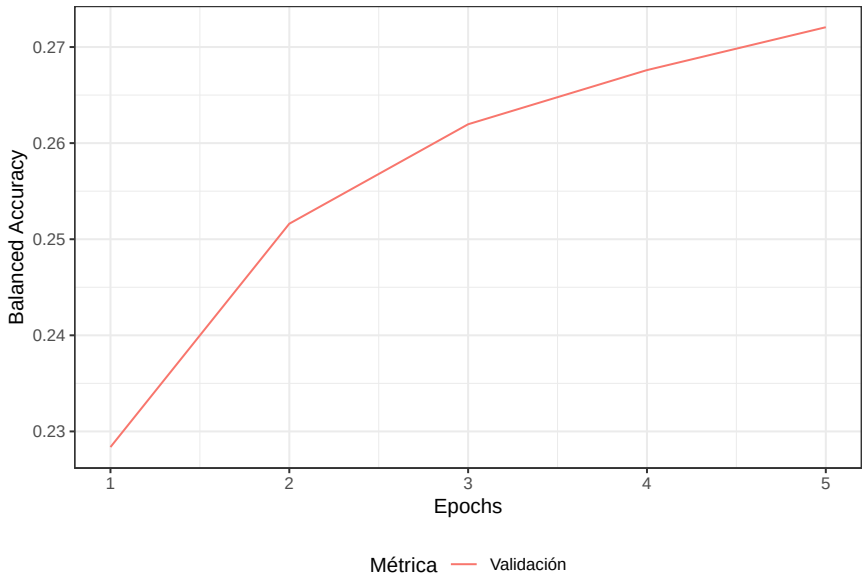

Figure 6: Resultado del Experimento 3

La Figura 7 muestra el resultado de la función de pérdida para los sets de validación y train.

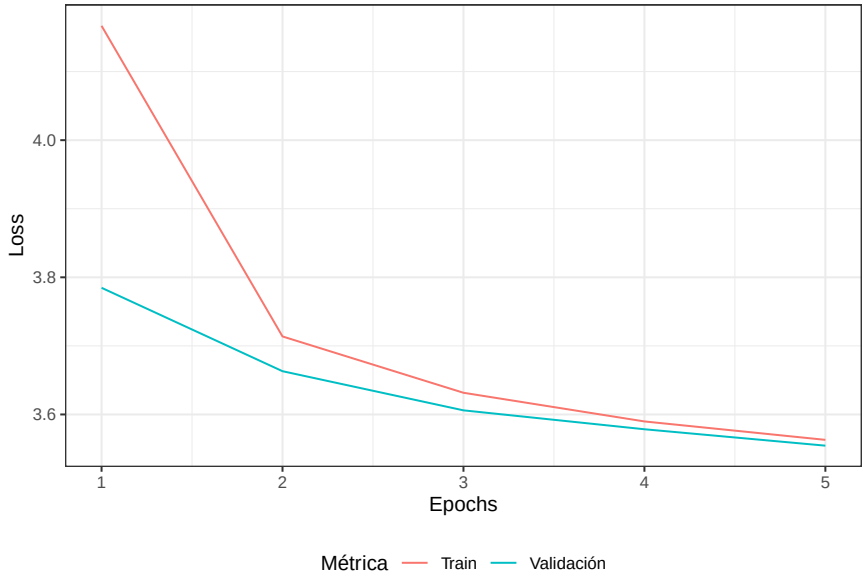

Figure 7: Resultado de la función de pérdida del Experimento 3

3.1.4 Experimento 4

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
8eb38861d3bd44719302813821672f512	dropout	0.3
8eb38861d3bd44719302813821672f512	embeddings	./data/SBW-vectors-300-min5.txt.gz
8eb38861d3bd44719302813821672f512	embeddings_size	300
8eb38861d3bd44719302813821672f512	epochs	3
8eb38861d3bd44719302813821672f512	hidden_layers	[256,128]
8eb38861d3bd44719302813821672f512	model_type	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 8 muestra la evolución de la métrica a lo largo de 5 épocas.

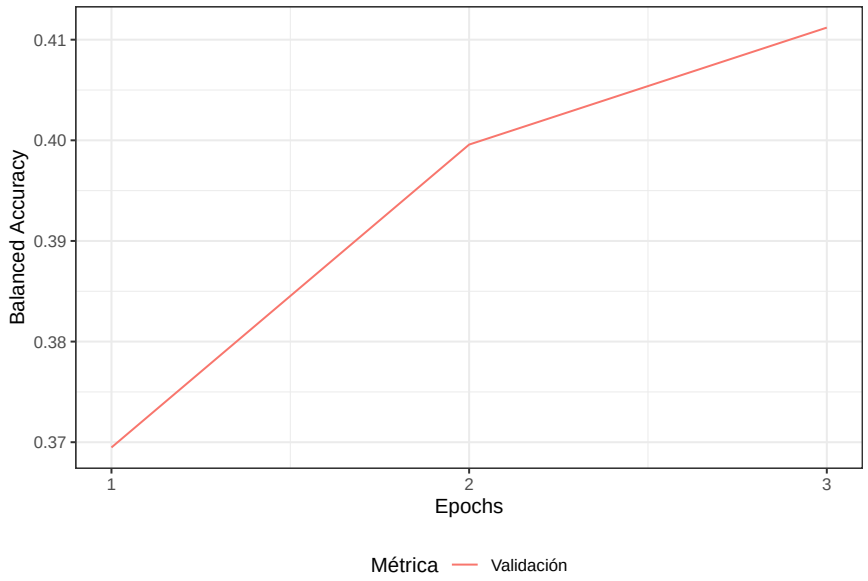

Figure 8: Resultado del Experimento 4

La Figura 9 muestra el resultado de la función de pérdida para los sets de validación y train.

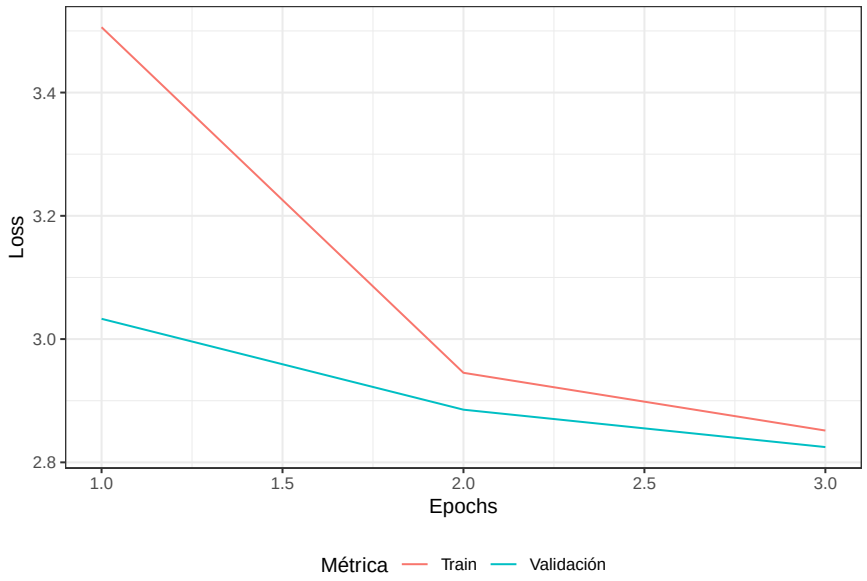

Figure 9: Resultado de la función de pérdida del Experimento 4

3.1.5 Experimento 5

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
c1021d64981f4197b4a3531049ca0746	dropout	0.5
c1021d64981f4197b4a3531049ca0746	epochs	5
c1021d64981f4197b4a3531049ca0746	hidden_layers	[384,256]
c1021d64981f4197b4a3531049ca0746	model_type	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 10 muestra la evolución de la métrica a lo largo de 5 épocas.

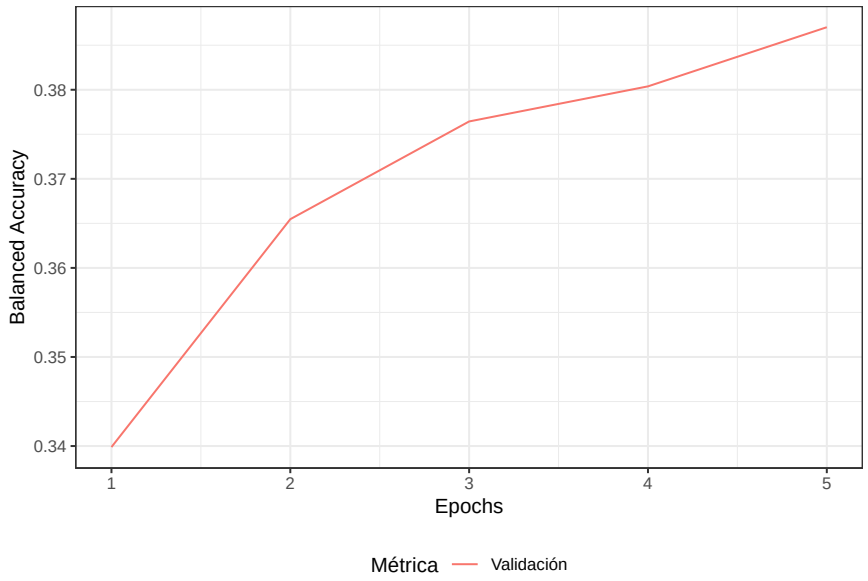

Figure 10: Resultado del Experimento 5

La Figura 11 muestra el resultado de la función de pérdida para los sets de validación y train.

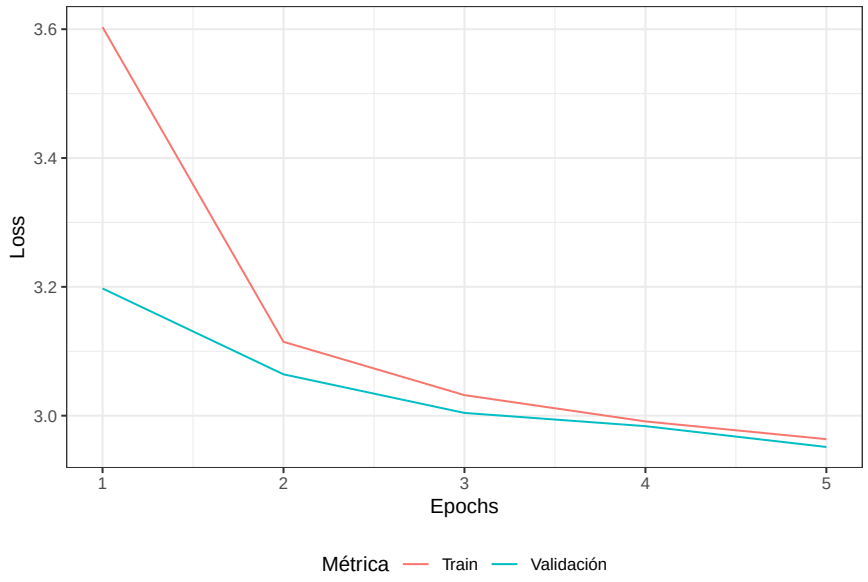

Figure 11: Resultado de la función de pérdida del Experimento 5

3.1.6 Experimento 6

El sexto experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
dd411f7374a540c0aaa041b8983d1a77	dropout	0.3
dd411f7374a540c0aaa041b8983d1a77	embeddings	./data/SBW-vectors-300-min5.txt.gz
dd411f7374a540c0aaa041b8983d1a77	embeddings_size	300
dd411f7374a540c0aaa041b8983d1a77	epochs	3
dd411f7374a540c0aaa041b8983d1a77	hidden_layers	[384,256]
dd411f7374a540c0aaa041b8983d1a77	model_type	MultilayerPerceptron

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 12 muestra la evolución de la métrica a lo largo de 5 épocas.

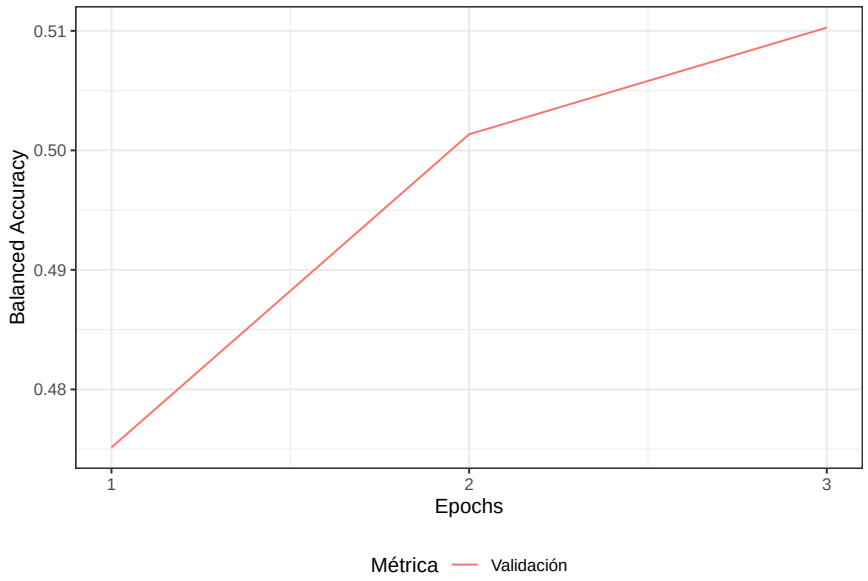

Figure 12: Resultado del Experimento 6

La Figura 13 muestra el resultado de la función de pérdida para los sets de validación y train.

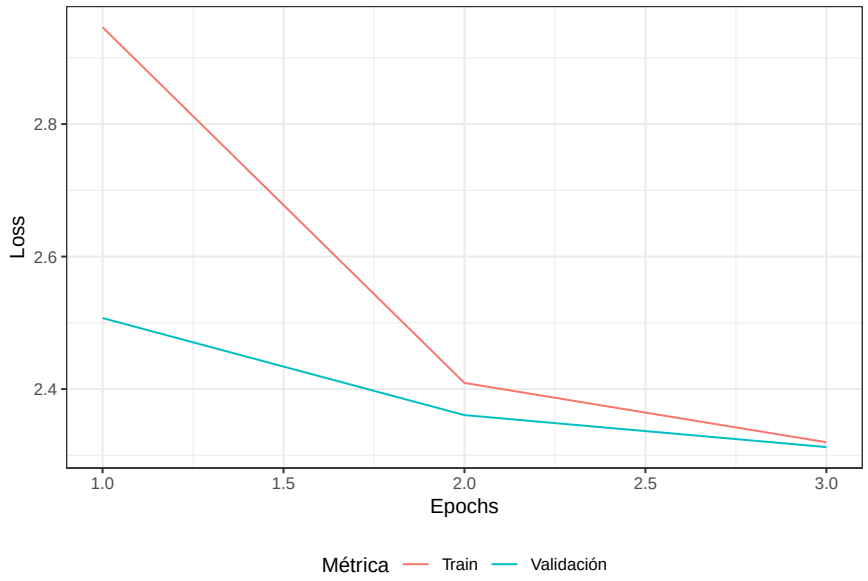

Figure 13: Resultado de la función de pérdida del Experimento 6

3.2 Redes neuronales convolucionales

Con esta red se realizaron 5 experimentos. Los detalles de cada uno de ellos se encuentran dentro de la carpeta `mlruns` y la subcarpeta `cnn` que acompañan el presente reporte. Algunos experimentos se corrieron en Nabucodonosor mientras que otros en Google Colab.

```
## ./mlruns/cnn/1/23ebde03beb1493d8bb501079f671af1
## ./mlruns/cnn/1/7ae725339f2340e9aece2661064cf30e
## ./mlruns/cnn/1/82d5f741ff5f446cb18000e5e355f9c7
## ./mlruns/cnn/1/e2ca84ce8e704a26a37a61bf1c9b23f6
## ./mlruns/cnn/1/ec42c61062f24933bd05e096d4a23e6c
```

Los experimentos de CNN consistieron en modificar distintos hiperparámetros y arquitectura de la red. Principalmente se centraron en lo siguiente:

- Se experimento modificando Learning Rate y Weight Decay,
- Se modificó la ultima capa usando Softmax y Relu,
- Se cambiaron los tamaños de los batchs,
- Se uso max pooling y avg pooling.

3.2.1 Experimento 1

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
23ebde03beb1493d8bb501079f671af1	dropout	0.3
23ebde03beb1493d8bb501079f671af1	embeddings	./data/SBW-vectors-300-min5.txt.gz
23ebde03beb1493d8bb501079f671af1	embeddings_size	300
23ebde03beb1493d8bb501079f671af1	epochs	5
23ebde03beb1493d8bb501079f671af1	model_type	CNN

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 14 muestra la evolución de la métrica a lo largo de 5 épocas.

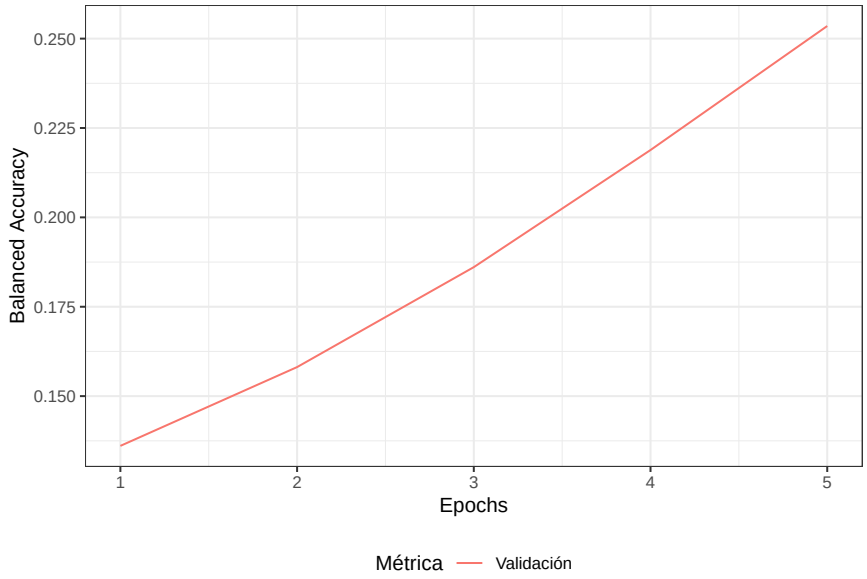

Figure 14: Resultado del Experimento 1

La Figura 15 muestra el resultado de la función de pérdida para los sets de validación y train.

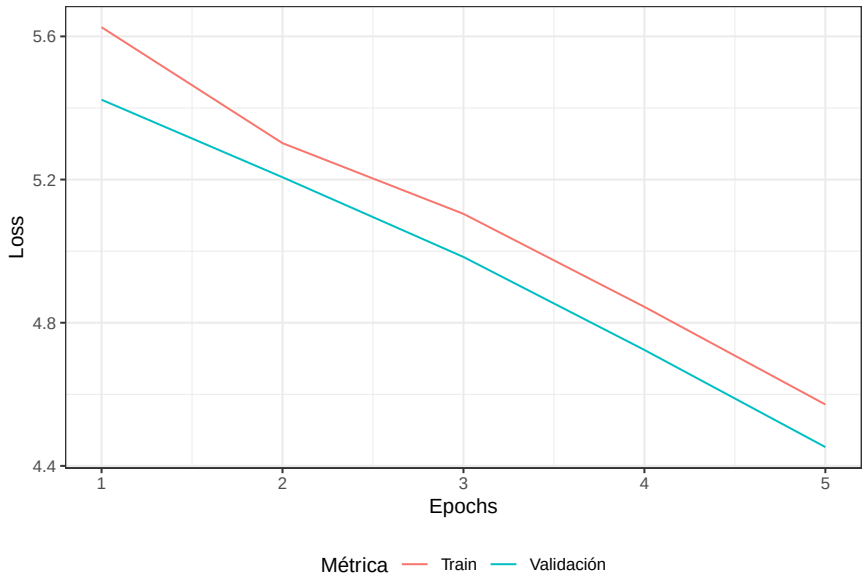

Figure 15: Resultado de la función de pérdida del Experimento 1

3.2.2 Experimento 2

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
7ae725339f2340e9aece2661064cf30e	dropout	0.3
7ae725339f2340e9aece2661064cf30e	embeddings	./data/SBW-vectors-300-min5.txt.gz
7ae725339f2340e9aece2661064cf30e	embeddings_size	300
7ae725339f2340e9aece2661064cf30e	epochs	5
7ae725339f2340e9aece2661064cf30e	model_type	CNN

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 16 muestra la evolución de la métrica a lo largo de 5 épocas.

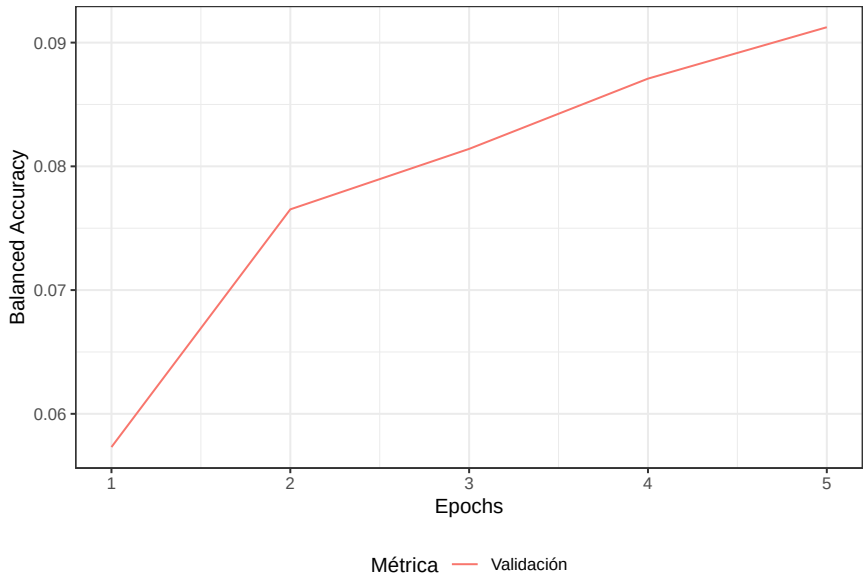

Figure 16: Resultado del Experimento 2

La Figura 17 muestra el resultado de la función de pérdida para los sets de validación y train.

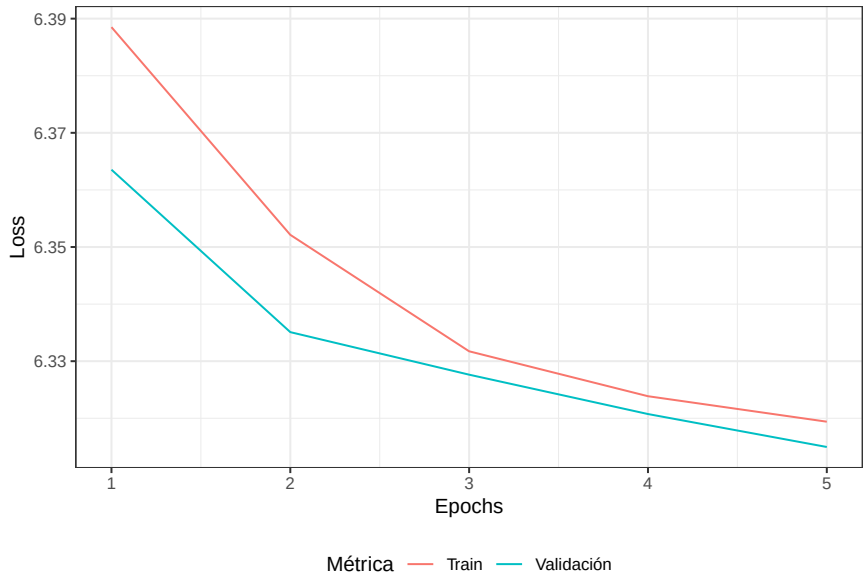

Figure 17: Resultado de la función de pérdida del Experimento 2

3.2.3 Experimento 3

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
82d5f741ff5f446cb18000e5e355f9c7	dropout	0.3
82d5f741ff5f446cb18000e5e355f9c7	embeddings	./data/SBW-vectors-300-min5.txt.gz
82d5f741ff5f446cb18000e5e355f9c7	embeddings_size	300
82d5f741ff5f446cb18000e5e355f9c7	epochs	5
82d5f741ff5f446cb18000e5e355f9c7	model_type	CNN

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 18 muestra la evolución de la métrica a lo largo de 5 épocas.

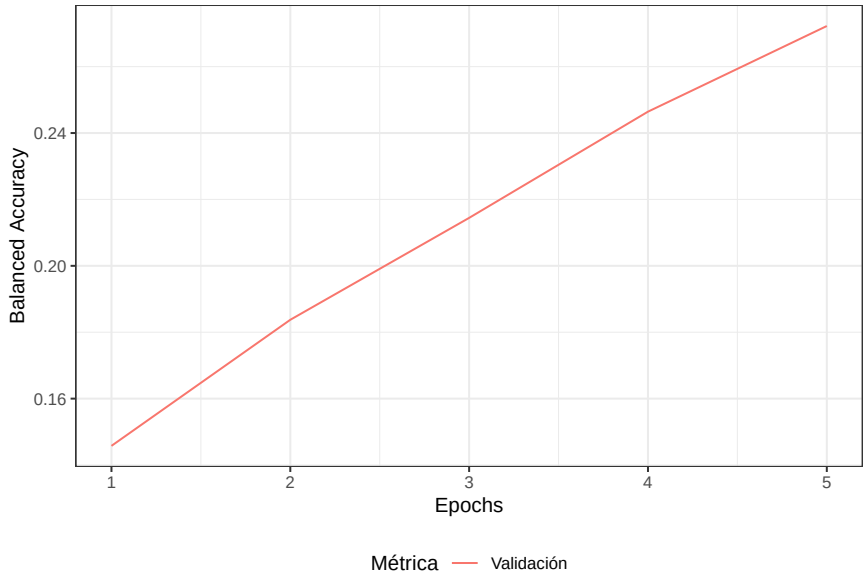

Figure 18: Resultado del Experimento 3

La Figura 19 muestra el resultado de la función de pérdida para los sets de validación y train.

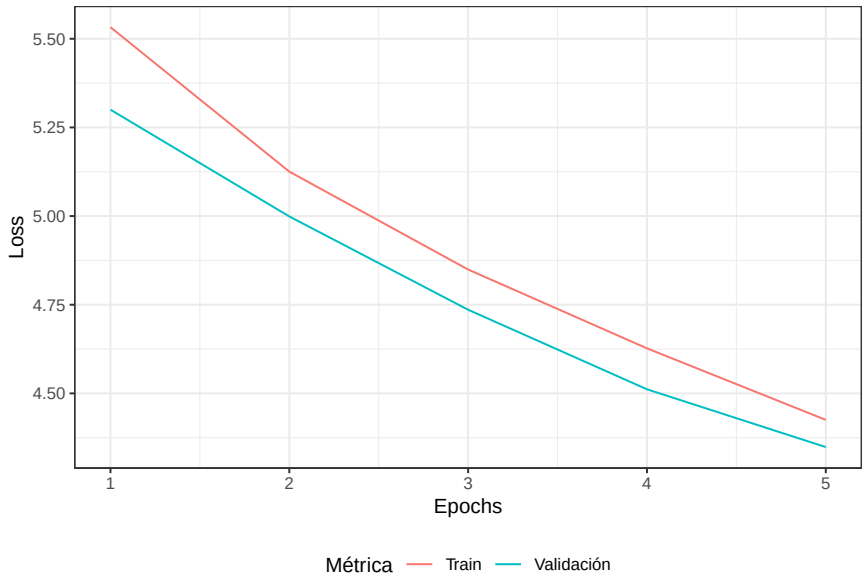

Figure 19: Resultado de la función de pérdida del Experimento 3

3.2.4 Experimento 4

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
e2ca84ce8e704a26a37a61bf1c9b23f6	dropout	0.3
e2ca84ce8e704a26a37a61bf1c9b23f6	embeddings	./data/SBW-vectors-300-min5.txt.gz
e2ca84ce8e704a26a37a61bf1c9b23f6	embeddings_size	300
e2ca84ce8e704a26a37a61bf1c9b23f6	epochs	5
e2ca84ce8e704a26a37a61bf1c9b23f6	model_type	CNN

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 20 muestra la evolución de la métrica a lo largo de 5 épocas.

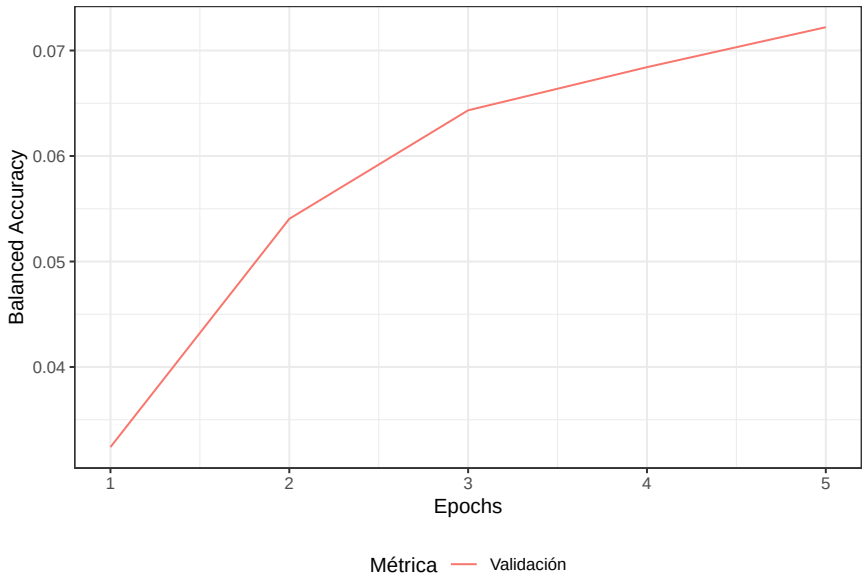

Figure 20: Resultado del Experimento 4

La Figura 21 muestra el resultado de la función de pérdida para los sets de validación y train.

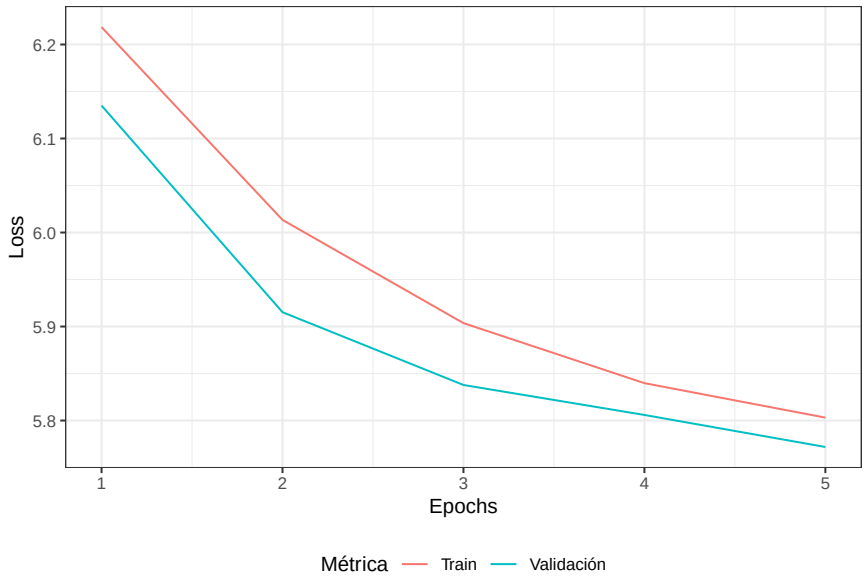

Figure 21: Resultado de la función de pérdida del Experimento 4

3.2.5 Experimento 5

El tercer experimento se realizó con los siguientes parámetros.

experimento	parametro	valor
ec42c61062f24933bd05e096d4a23e6c	dropout	0.3
ec42c61062f24933bd05e096d4a23e6c	embeddings	./data/SBW-vectors-300-min5.txt.gz
ec42c61062f24933bd05e096d4a23e6c	embeddings_size	300
ec42c61062f24933bd05e096d4a23e6c	epochs	5
ec42c61062f24933bd05e096d4a23e6c	model_type	CNN

Para evaluar los resultados del entrenamiento se calculó el `balanced accuracy` para los set validación. La Figura 22 muestra la evolución de la métrica a lo largo de 5 épocas.

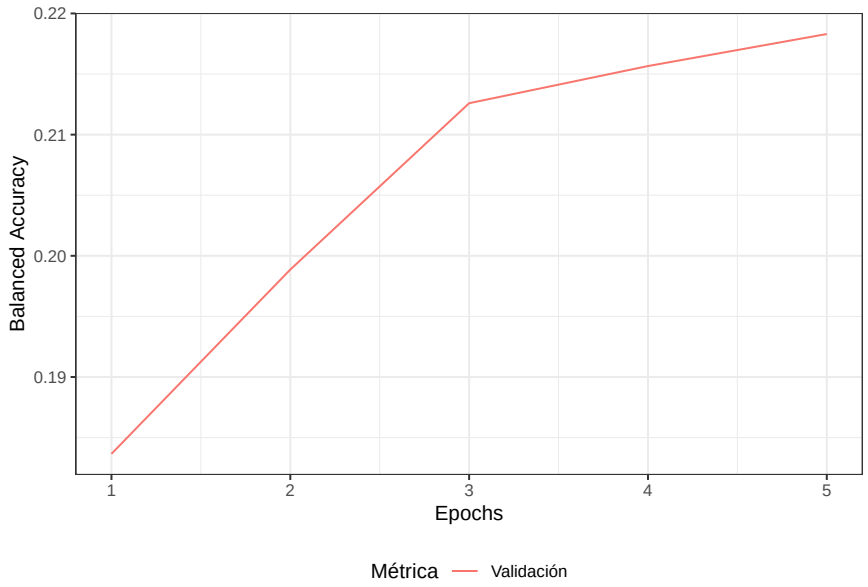

Figure 22: Resultado del Experimento 5

La Figura 23 muestra el resultado de la función de pérdida para los sets de validación y train.

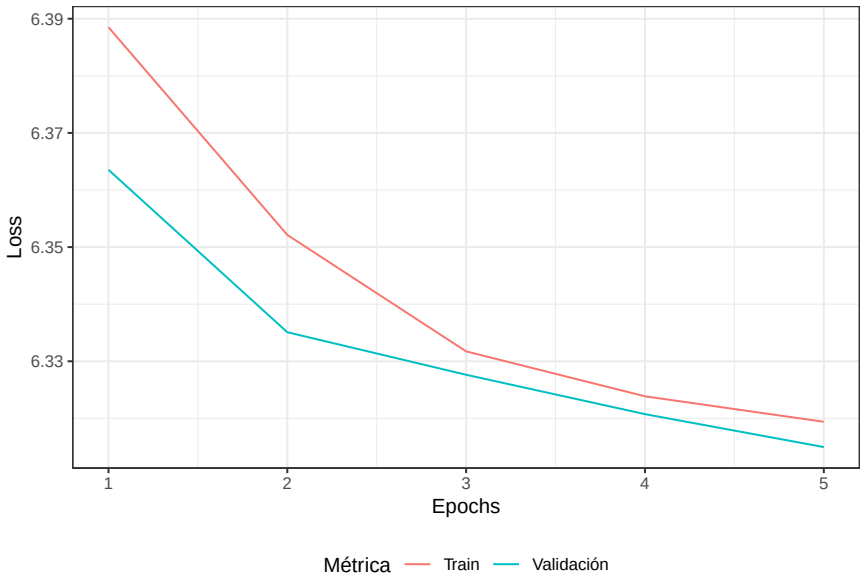

Figure 23: Resultado de la función de pérdida del Experimento 2