

PERCEPTRÓN MULTICAPA (MLP)

Experimento número 0

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron
random_buffer_size	2048
freeze_embeddings	True
batch_size	128
lr	1,00E-03
weight_decay	1,00E-05

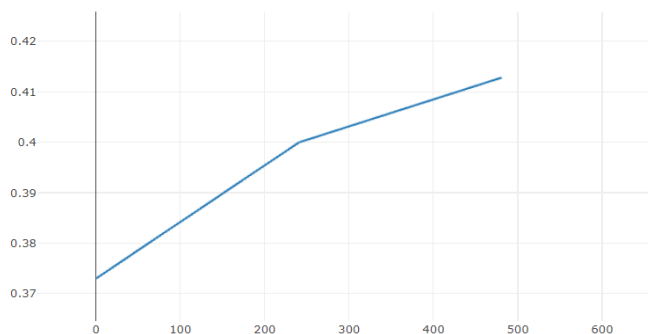
▼ Parameters

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron

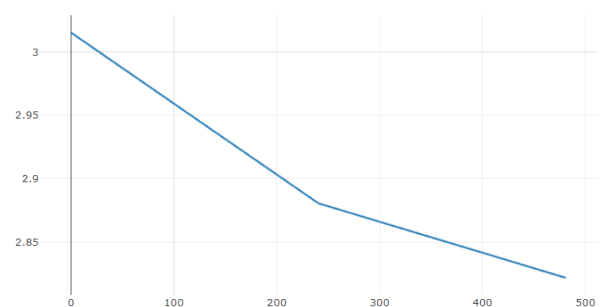
▼ Metrics

Name	Value
train_loss 	2.845
validation_bacc 	0.413
validation_loss 	2.822

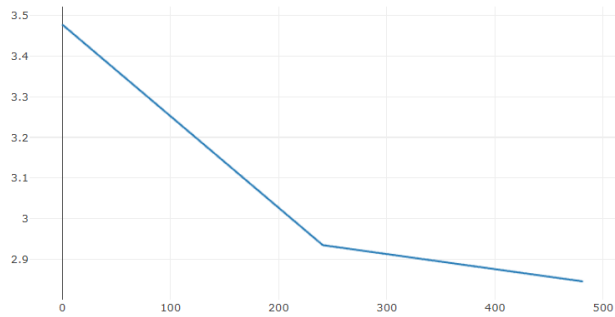
validation_bacc



validation_loss



Train_loss



Experimento número 1

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron
random_buffer_size	2048
freeze_embeddings	True
batch_size	128
lr	1,00E-05
weight_decay	1,00E-05

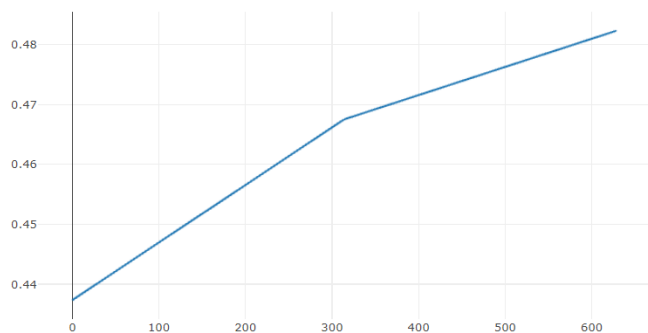
▼ Parameters

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
freeze_embeddings	True
hidden_layers	[256, 128]
model_type	Multilayer Perceptron

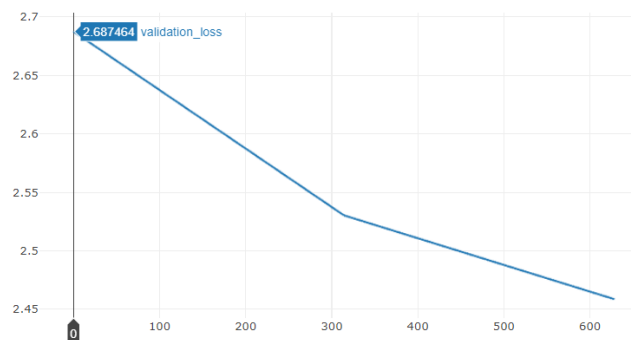
▼ Metrics

Name	Value
train_loss 📈	2.485
validation_bacc 📈	0.482
validation_loss 📈	2.459

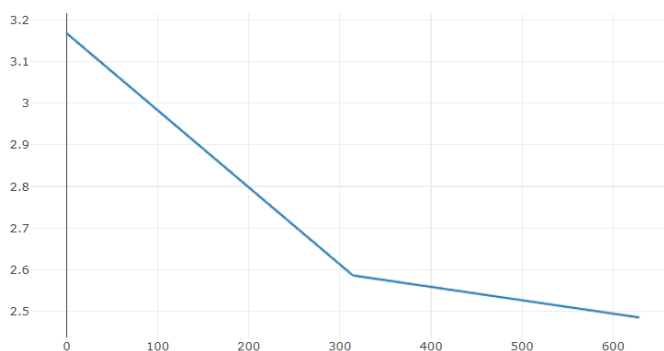
validation_bacc



validation_loss



Train_loss



Experimento número 2

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron
random_buffer_size	2048
freeze_embeddings	True
batch_size	128
lr	1,00E-10
weight_decay	1,00E-10

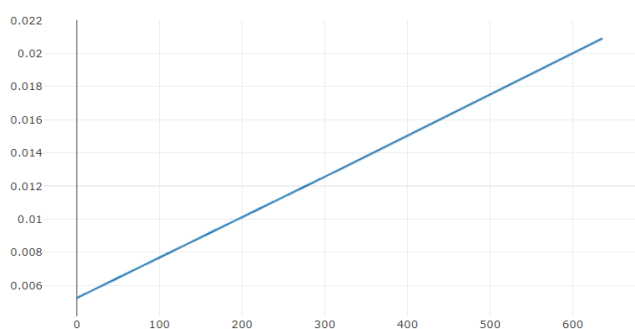
▼ Parameters

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
freeze_embeddings	True
hidden_layers	[256, 128]
model_type	Multilayer Perceptron

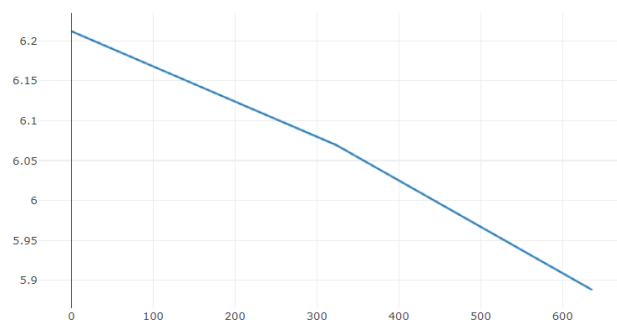
▼ Metrics

Name	Value
train_loss 	5.988
validation_bacc 	0.021
validation_loss 	5.888

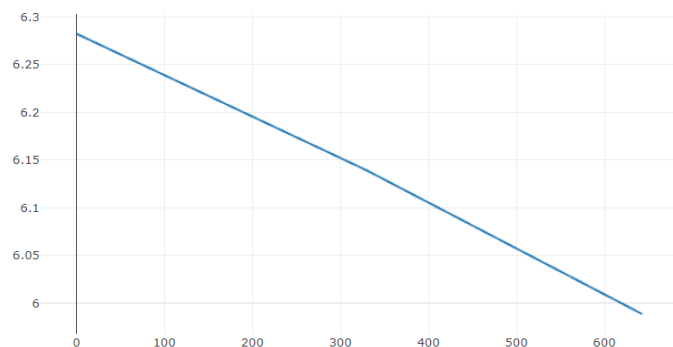
validation_bacc



validation_loss



Train_loss



Experimento número 3



Name	Value
dropout	0,3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	4
hidden_layers	[256, 128]
model_type	Multilayer Perceptron
random_buffer_size	2048

freeze_embeddings	True
batch_size	128
lr	1,00E-05
weight_decay	1,00E-05

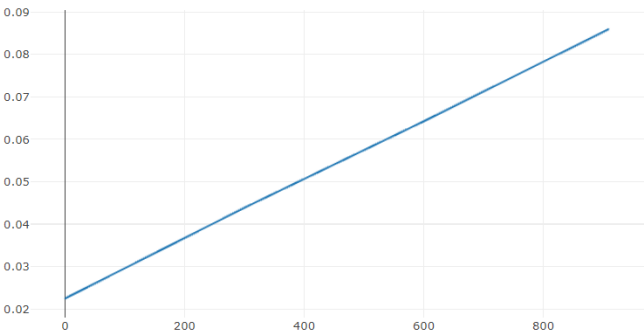
▼ Parameters

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	4
freeze_embeddings	True
hidden_layers	[256, 128]
lr	1e-05
model_type	Multilayer Perceptron
weight_decay	1e-05

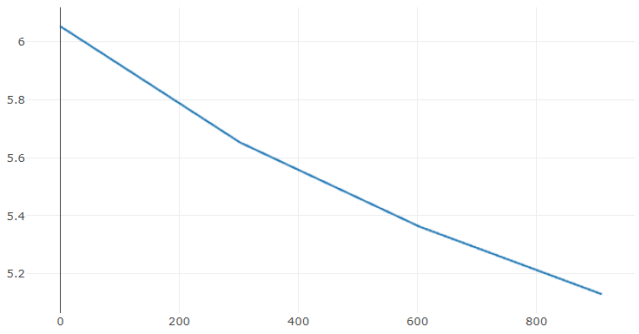
▼ Metrics

Name	Value
train_loss 	5.242
validation_bacc 	0.086
validation_loss 	5.13

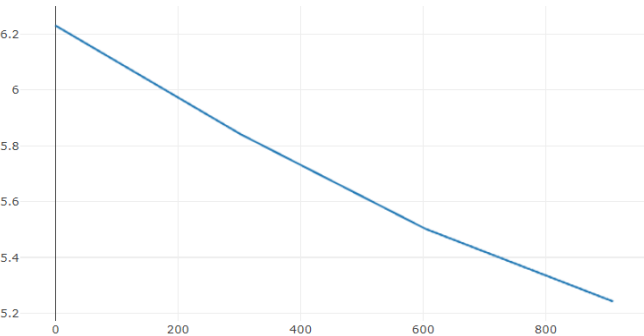
validation_bacc



validation_loss



Train_loss



RED CONVOLUCIONAL (CNN)




Experimento número 0

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	4
filters_count	100
filters_length	[2, 3, 4]
model_type	Convulsional Network
random_buffer_size	2048
freeze_embeddings	True
lr	1,00E-05
weight_decay	1,00E-05

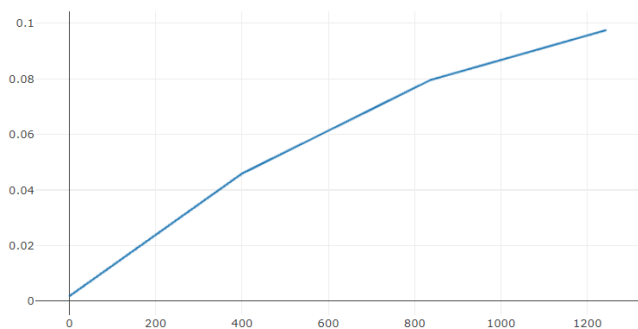
Parameters

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	4
filters_count	100
filters_length	[2, 3, 4]
freeze_embeddings	True
lr	1e-05
model_type	Convulsional Network
weight_decay	1e-05

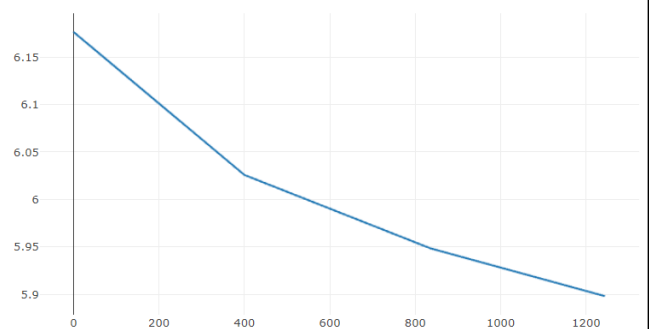
Metrics

Name	Value
train_loss 	5.921
validation_bacc 	0.098
validation_loss 	5.898

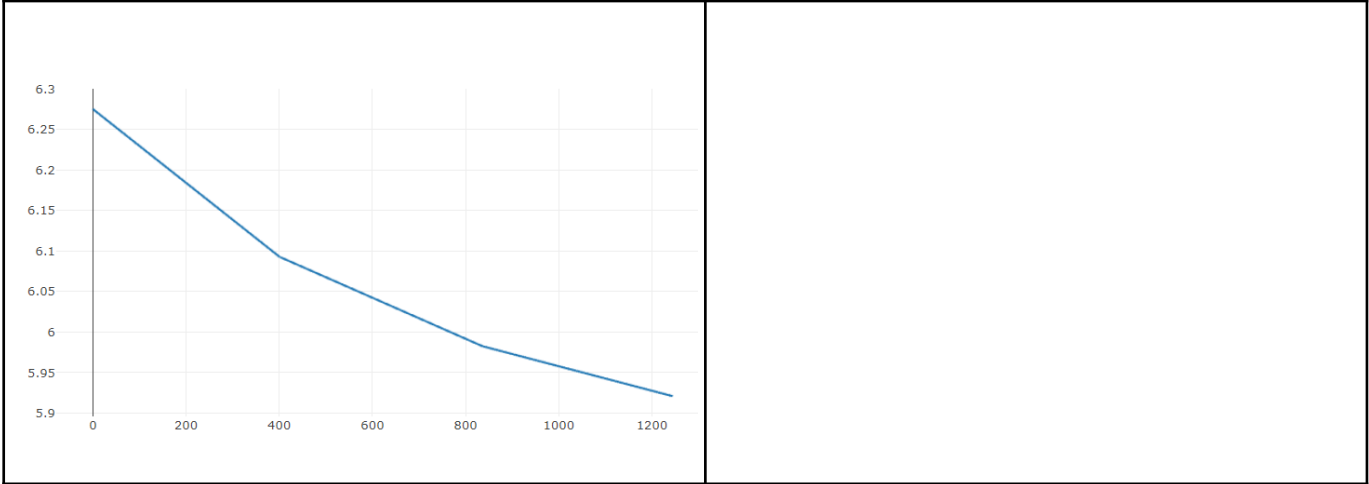
validation_bacc



validation_loss



Train_loss



Experimento número 1

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
filters_count	100
filters_length	[2, 3, 4]
model_type	Convulsional Network
random_buffer_size	2048
freeze_embeddings	True
lr	1,00E-03
weight_decay	1,00E-05

Parameters

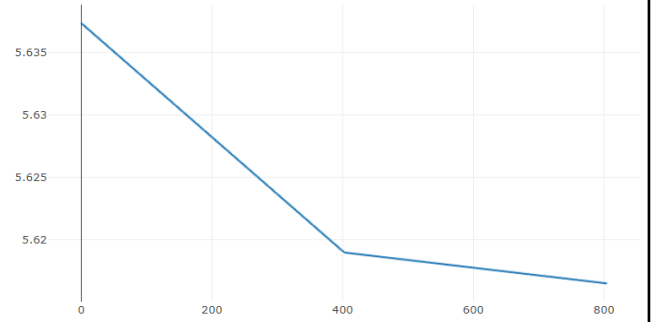
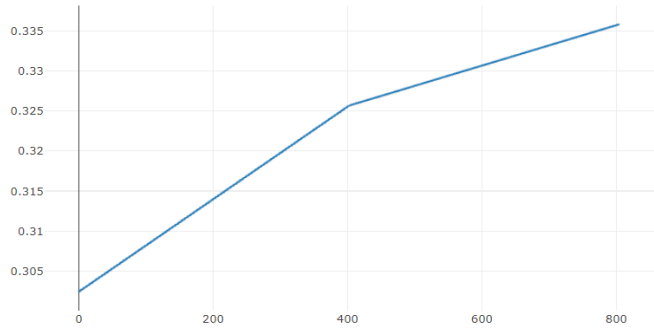
Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
filters_count	100
filters_length	[2, 3, 4]
freeze_embeddings	True
lr	0.001
model_type	Convulsional Network
weight_decay	1e-05

Metrics

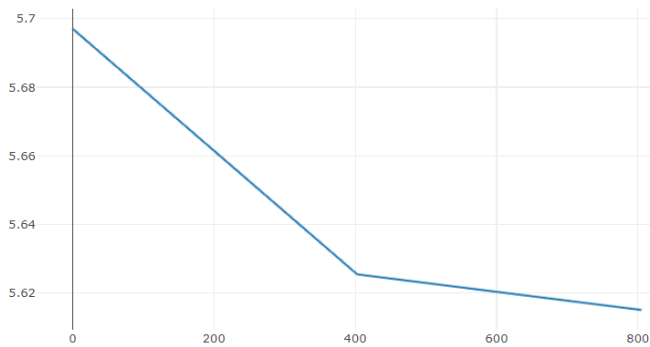
Name	Value
train_loss 	5.615
validation_bacc 	0.336
validation_loss 	5.616

validation_bacc

validation_loss



Train_loss



Experimento número 2

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	2
filters_count	100
filters_length	[2, 3, 4]
model_type	Convulsional Network
random_buffer_size	2048
freeze_embeddings	True
lr	1,00E-03
weight_decay	1,00E-03

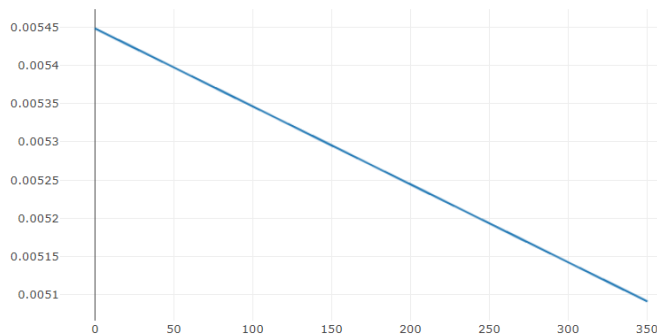
▼ Parameters

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	2
filters_count	100
filters_length	[2, 3, 4]
freeze_embeddings	True
lr	0.001
model_type	Convulsional Network
weight_decay	0.001

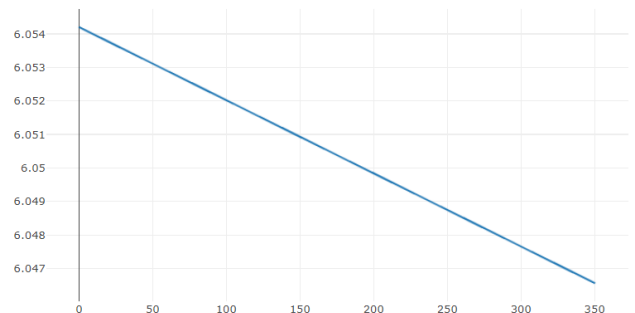
▼ Metrics

Name	Value
train_loss 	6.049
validation_bacc 	0.005
validation_loss 	6.047

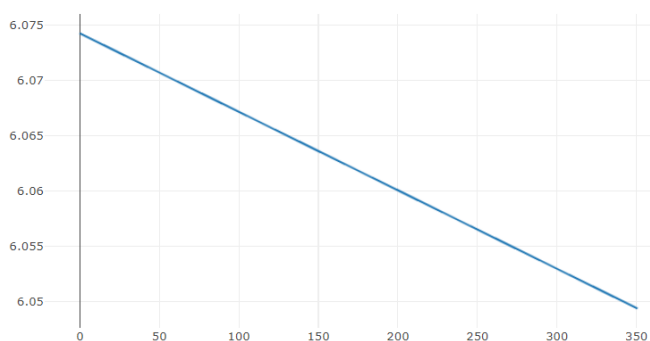
validation_bacc



validation_loss



Train_loss



Experimento número 3

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	10

filters_count	100
filters_length	[2, 3, 4]
model_type	Convulsional Network
random_buffer_size	2048
freeze_embeddings	True
lr	1,00E-03
weight_decay	1,00E-05

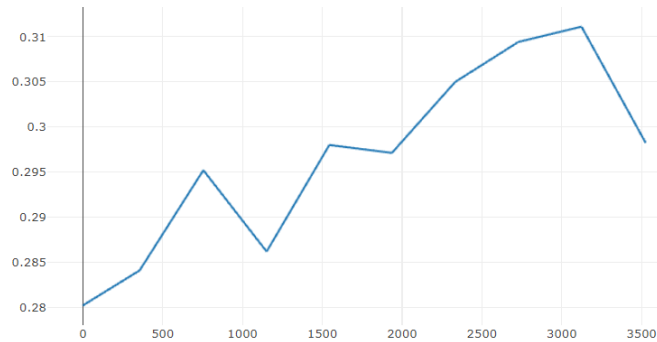
▼ Parameters

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	10
filters_count	100
filters_length	[2, 3, 4]
freeze_embeddings	True
lr	0.001
model_type	Convulsional Network
weight_decay	1e-05

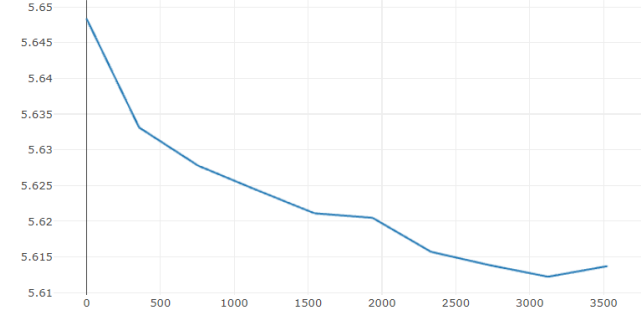
▼ Metrics

Name	Value
train_loss 	5.611
validation_bacc 	0.298
validation_loss 	5.614

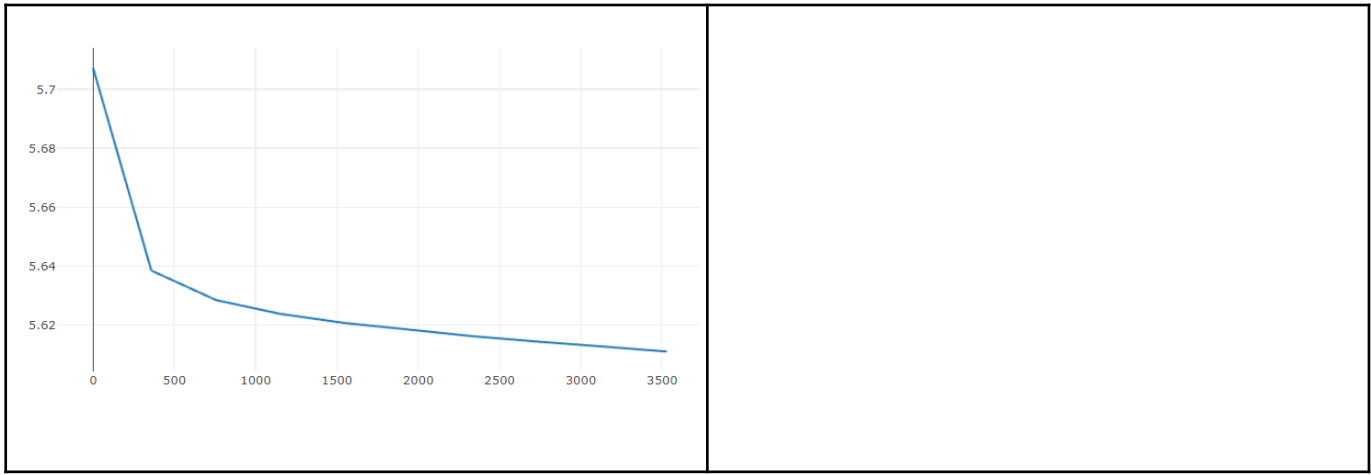
validation_bacc



validation_loss



Train_loss

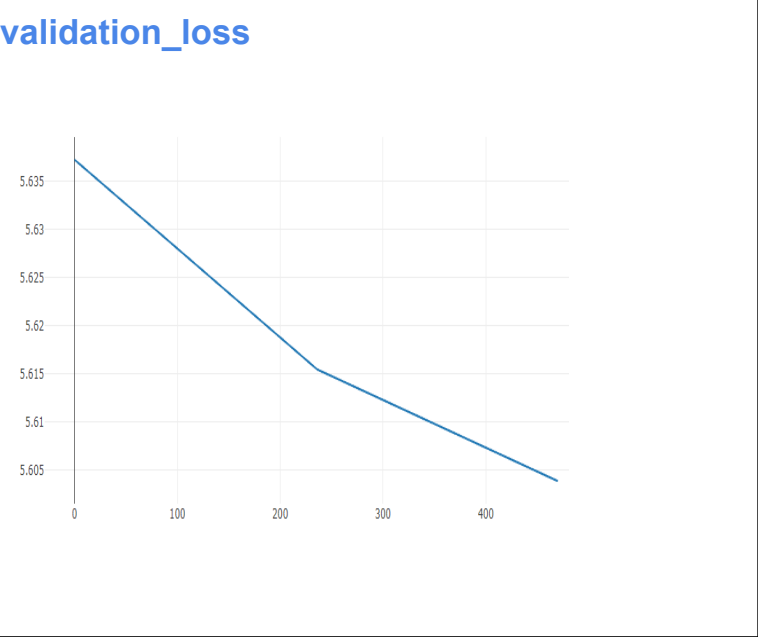
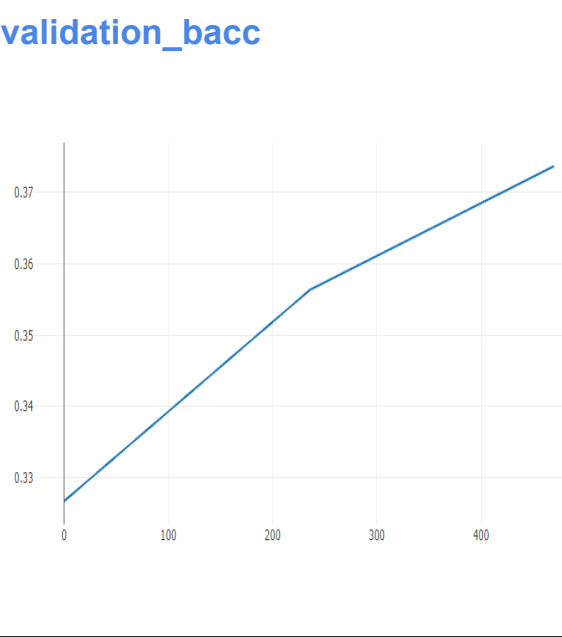
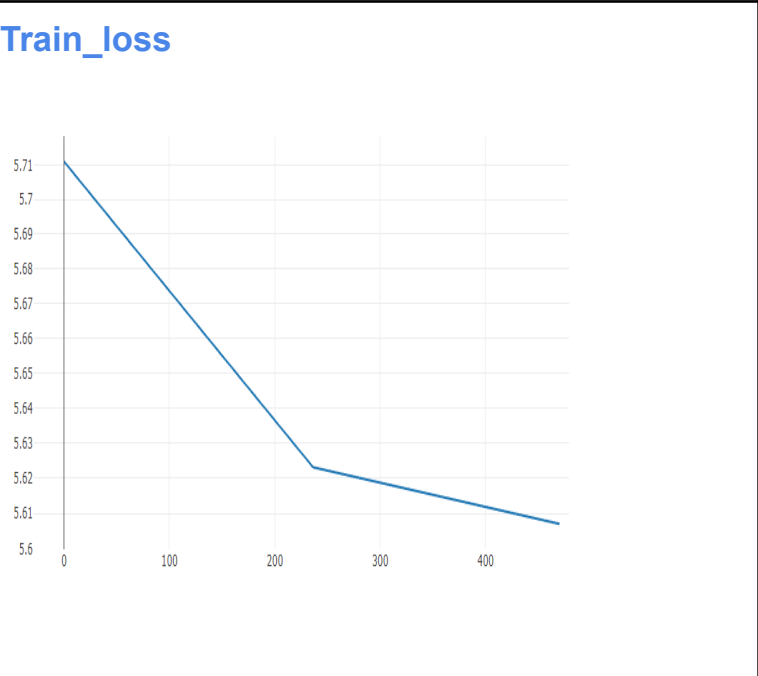


Experimento número 4

Hiper-parámetros:

- random_buffer_size=2048
- batch_size=256
- lr=1e-3
- weight_decay=1e-5

▼ Parameters	
Name	Value
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
fc_size	128
filters_count	100
filters_length	[2, 3, 4]
model_type	Red convolucional
▼ Metrics	
Name	Value
train_loss 📊	5.607
validation_bacc 📊	0.374
validation_loss 📊	5.604

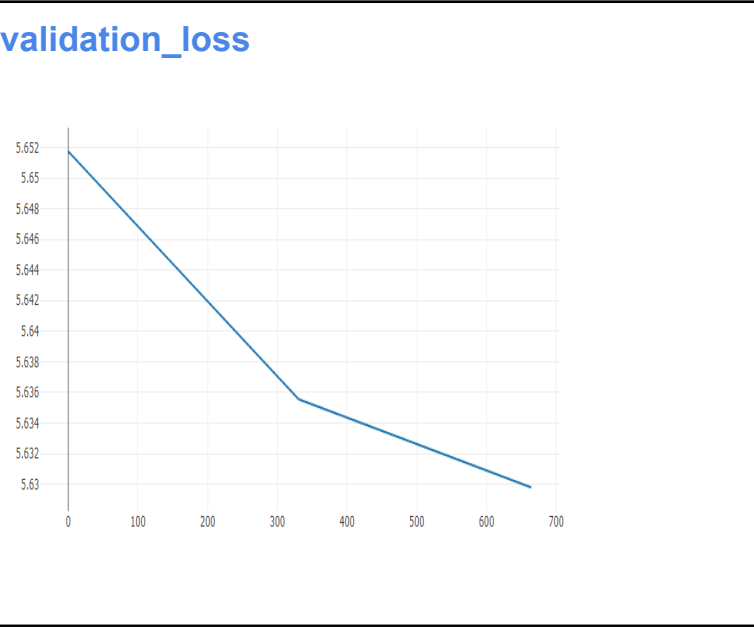


Experimento número 5

Hiper-parámetros:

- random_buffer_size=4096
- batch_size=128
- lr=1e-3
- weight_decay=1e-5

▼ Parameters	
Name	Value
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
fc_size	128
filters_count	200
filters_length	[3, 4, 5]
model_type	Red convolucional
▼ Metrics	
Name	Value
train_loss 📈	5.632
validation_bacc 📈	0.283
validation_loss 📈	5.63



Experimento número 6

Hiper-parámetros:

- random_buffer_size=4096
- batch_size=128
- lr=1e-3
- weight_decay=1e-5

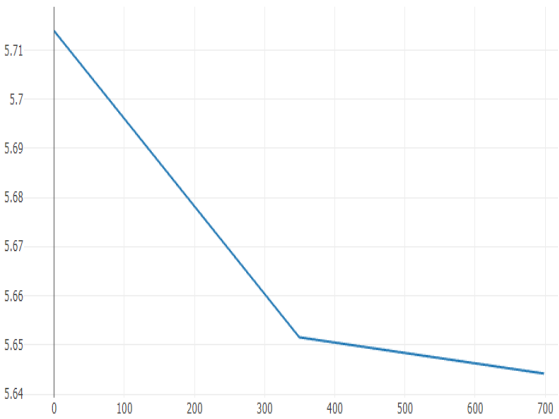
Parameters

Name	Value
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
fc_size	128
filters_count	300
filters_length	[3, 4, 5]
model_type	Red convolucional

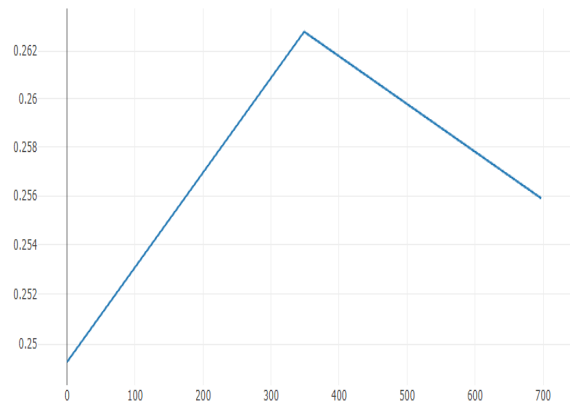
Metrics

Name	Value
train_loss 🔗	5.644
validation_bacc 🔗	0.256
validation_loss 🔗	5.642

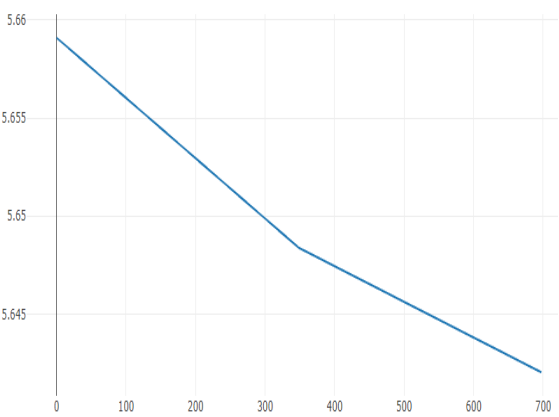
Train_loss



validation_bacc



validation_loss



Experimento número 7

Hiper-parámetros:

- random_buffer_size=2048
- batch_size=128
- lr=1e-5
- weight_decay=1e-7

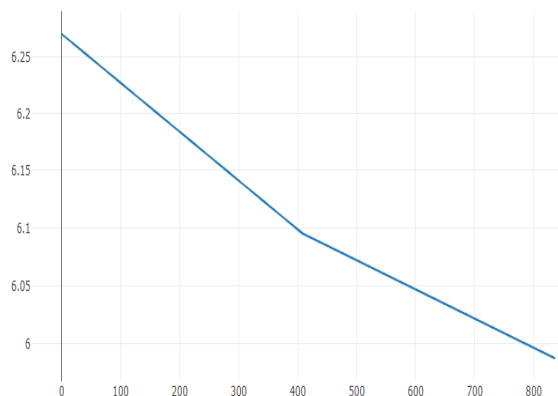
▼ Parameters

Name	Value
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
fc_size	128
filters_count	100
filters_length	[2, 3, 4]
model_type	Red convolucional

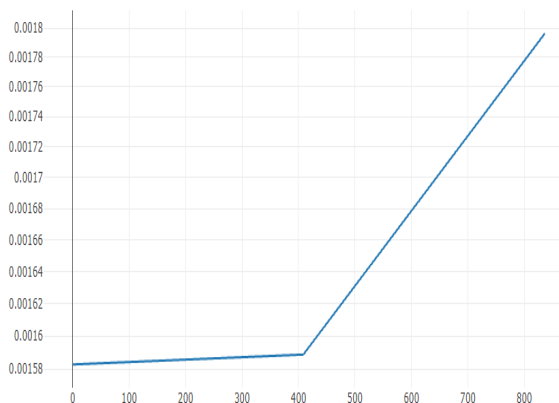
▼ Metrics

Name	Value
train_loss 🔗	5.988
validation_bacc 🔗	0.002
validation_loss 🔗	5.951

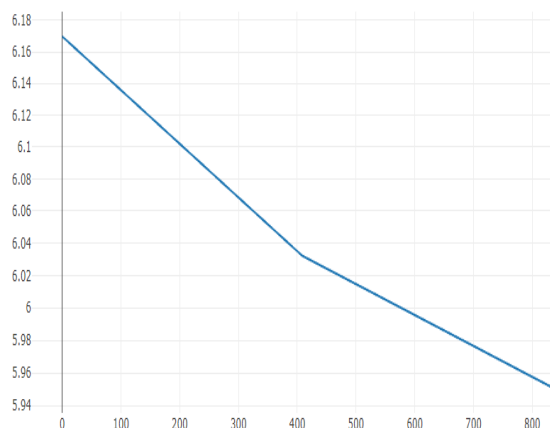
Train_loss



validation_bacc



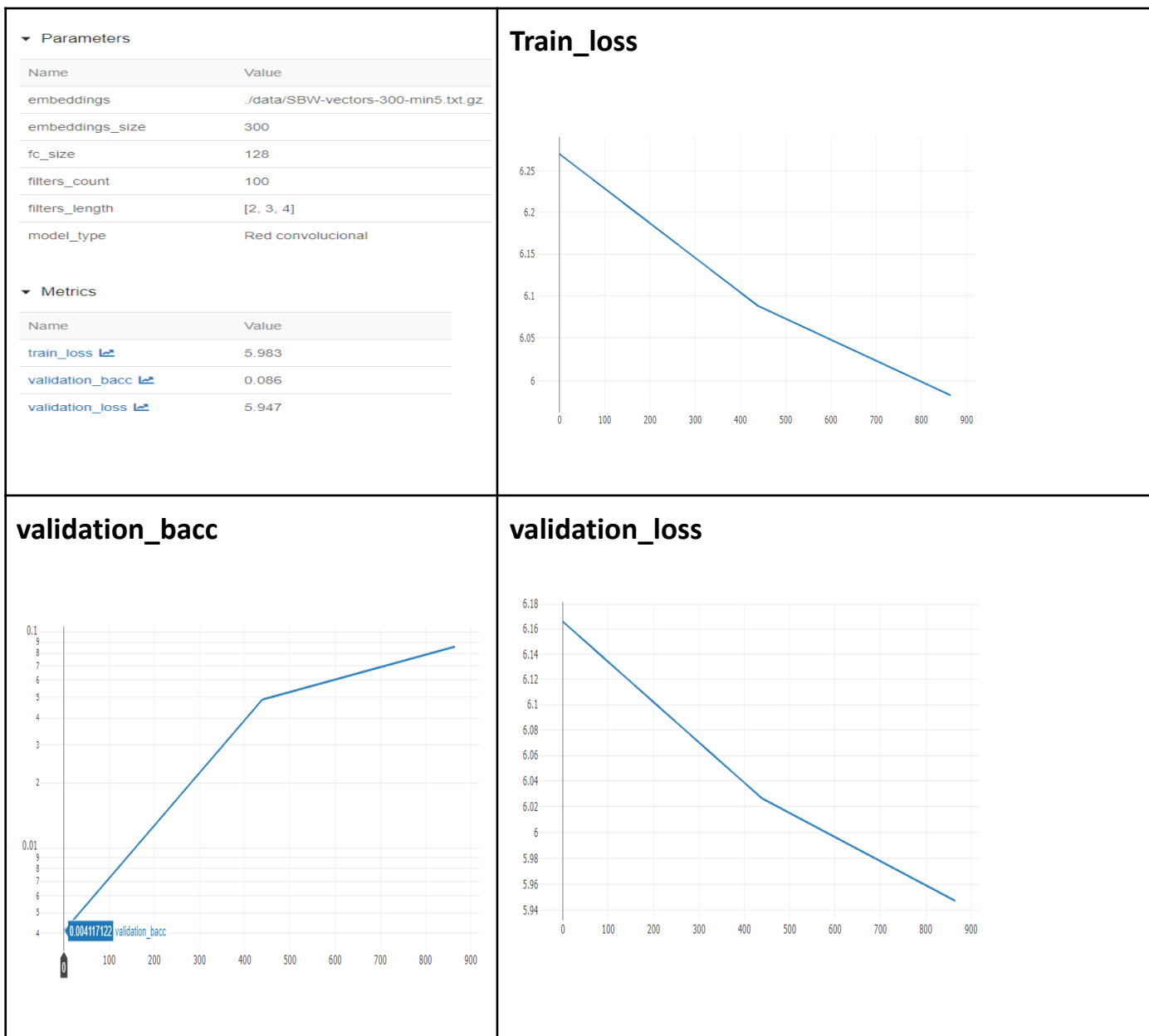
validation_loss



Experimento número 8

Hiper-parámetros:

- random_buffer_size=2048
- batch_size=128
- lr=1e-5
- weight_decay=1e-5



PERCEPTRÓN MULTICAPA (MLP)	
Modelo con conjunto de test	
Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron
random_buffer_size	2048
freeze_embeddings	True
batch_size	128
lr	1,00E-05

weight_decay

1,00E-05

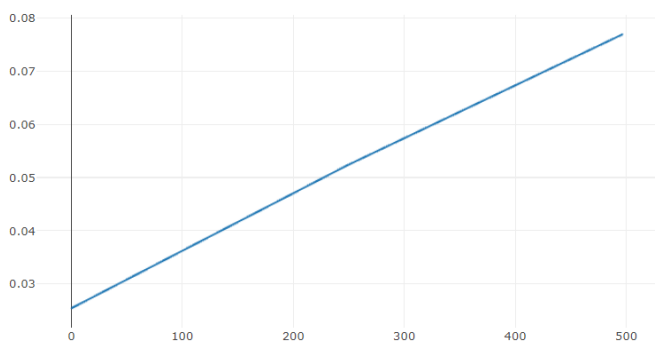
▼ Parameters

Name	Value
dropout	0.2
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
freeze_embeddings	True
hidden_layers	[256, 128]
lr	1e-05
model_type	Multilayer Perceptron
weight_decay	1e-05

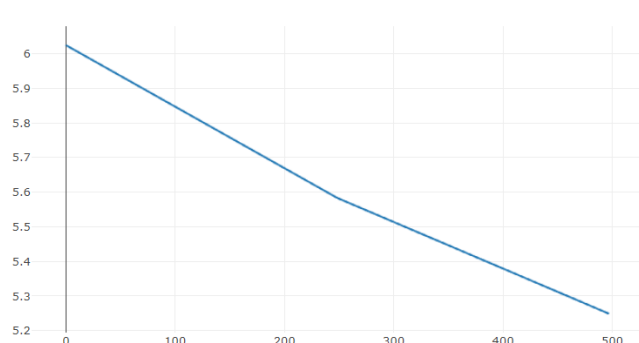
▼ Metrics

Name	Value
test_bacc	0.083
test_loss	5.58
train_loss	5.408
validation_bacc	0.077
validation_loss	5.249

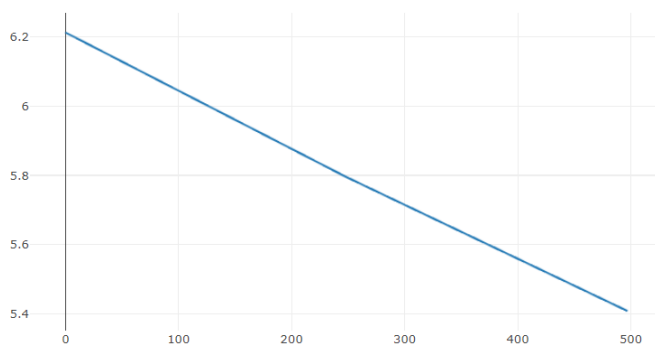
validation_bacc



validation_loss



Train_loss



Consideraciones sobre los hiperparámetros utilizados:

Utilizamos el parámetro **freeze_embeddings** en True, para que los embeddings que ya se encuentran preprocesados queden congelados, esto tiene por objetivo principal encontrar similitudes entre ambos conjuntos de datos sin influir en el análisis que realiza el modelo. La representatividad de los embeddings utilizados nos será útil para el procesamiento de nuestro dataset si se encuentran similitudes, de lo contrario obtendremos muchas palabras en el vector UKN/unknown.

También utilizamos **dropout** en redes MLP como método de regularización, el cual consiste en apagar la salida de determinadas neuronas durante el entrenamiento, lo cual permite al modelo una mejor generalización y evitar el sobreajuste.

Consideraciones sobre los resultados obtenidos:

Una función de pérdida o loss representa un valor que indica cuán mala fue la predicción del modelo en un sólo ejemplo. Una pérdida o loss alto hace referencia a una mala predicción para cualquier modelo. La pérdida se calcula sobre la base del entrenamiento y la validación y representa cuán bien está funcionando el modelo para estos dos conjuntos. A diferencia de la precisión, una pérdida no es un porcentaje.

También hay que tener en cuenta que si el validation loss es mayor que training loss es posible que existe algún tipo de sobreajuste en el modelo, en cambio si validation loss es menor que training loss puede que exista desajuste en el mismo. Por lo tanto, el objetivo es hacer que la validation loss sea lo más pequeña posible.

Además, si recordamos el Balanced Accuracy se calcula como $(\text{Sensitivity} + \text{Specificity}) / 2$. Esta métrica puede variar y tornarse un poco imprecisa si las clases no están balanceadas, para lo cual se puede utilizar otro tipo de métrica con mayor índice de asertividad. Por último, el **lr** y el **weight_decay** que son parámetros de optimizador tienen una muy marcada influencia en el valor del Balanced Accuracy.

De los experimentos realizados concluimos que el mejor modelo en las fase de training y validation es el experimento nro. 1, el cual utiliza redes Perceptrón Multicapa o MLP.

