Aprendizaje Profundo

Lab

Zarzosa Valdivia, Fernando [1] Mordidoni, Emilio [1] Matías Bettera Marcat [1] Lucas Didoné [1] Agustin Mauras Velez [1]

[1]Diplodatos 2020 (FAMAF-UNC)

Content

1. Intro
2. MLP, Multilayer Perceptron
3. CNN, Convolutional Neural Network
4. Conclusions

Resume

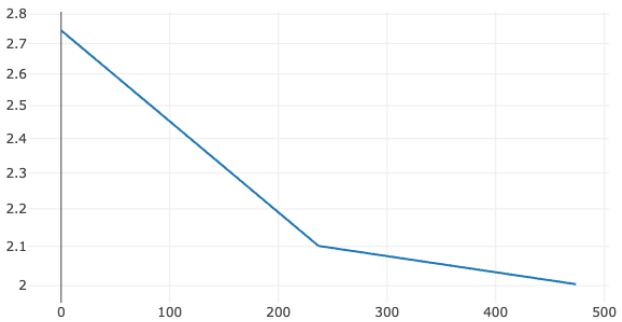
- We process various runs of two Multilayer Perceptron (MLP) models and two runs of convolutional neural network (CNN) models
- Due to limited resources we only show runs of the different models, and not the complete search of models
- It is not always true that additional runs improve accuracy or reduce losses (it obeys to the stochastic nature of the process)

Baseline MLP_1

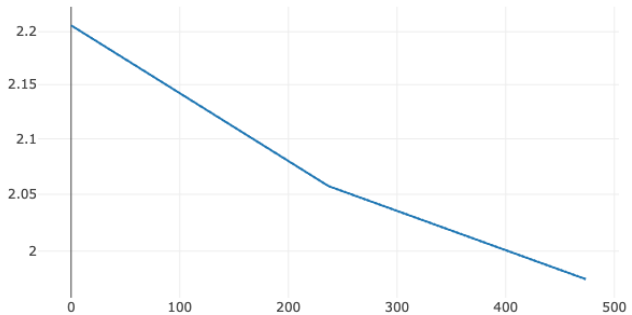
- The balanced accuracy metric, which take into account unbalanced data, shows a 57.6% accuracy (1st Modelo, MLP1) y 57.4% (2nd Model, MLP1a).
- The second model is less effective & efficient (higher loss & accuracy).

MLP_1

- Train Loss (logs)



Validation Loss (logs)



- MLFlow results

▾ Parameters

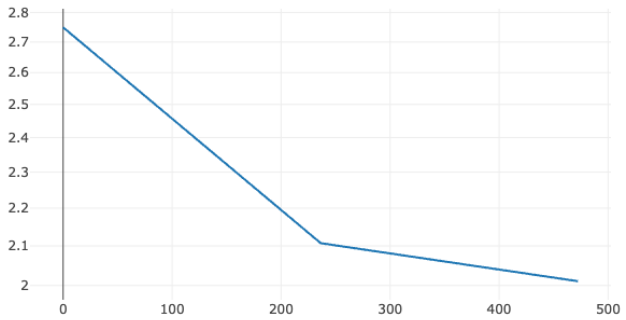
Name	Value
dropout	0.025
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron

▾ Metrics

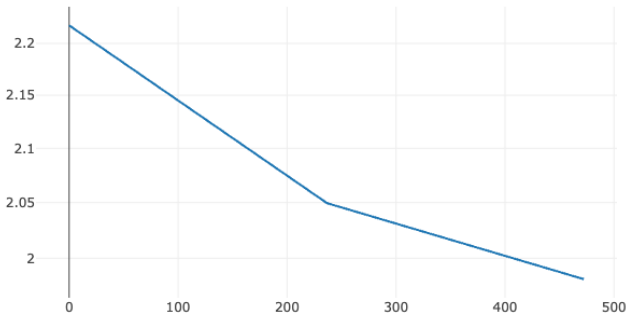
Name	Value
train_loss	2.004
validation_bacc	0.576
validation_loss	1.976

MLP$_1$ 2nd *round*

- Train Loss (logs)



Validation Loss (logs)



- MLFlow results

▾ Parameters

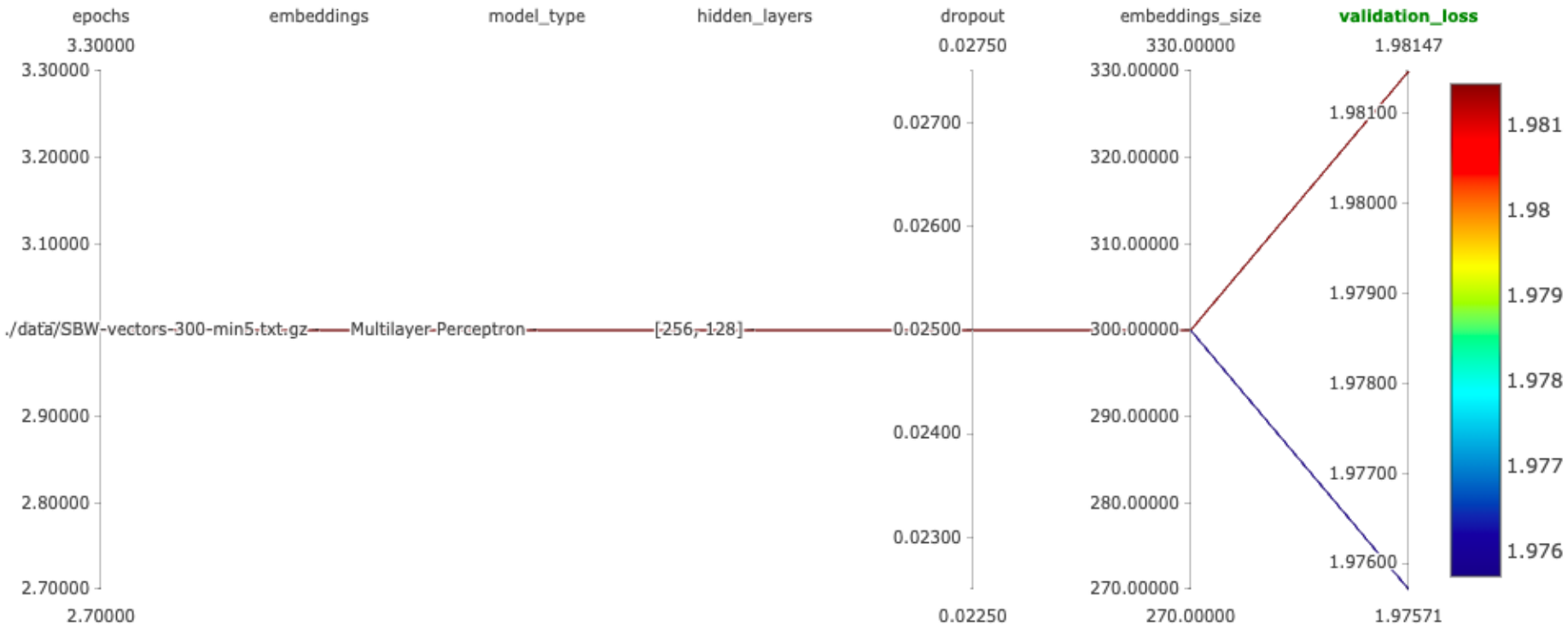
Name	Value
dropout	0.025
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
hidden_layers	[256, 128]
model_type	Multilayer Perceptron

▾ Metrics

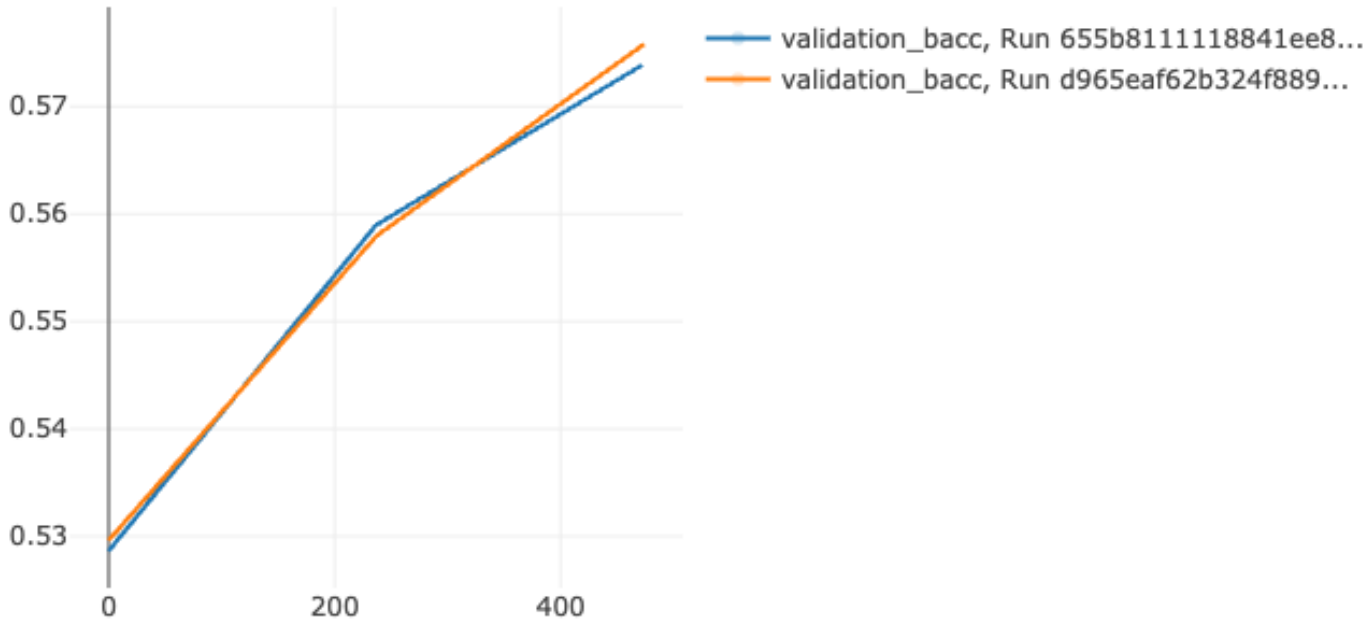
Name	Value
train_loss	2.011
validation_bacc	0.574
validation_loss	1.981

Benchmarking both MLP_1 *models*

Second round less effective & efficient



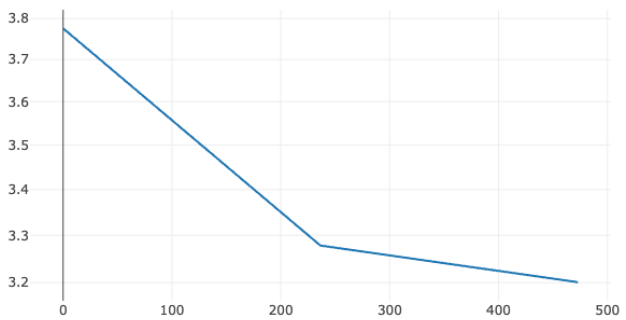
Validation BACC for both MLP_1 *models*



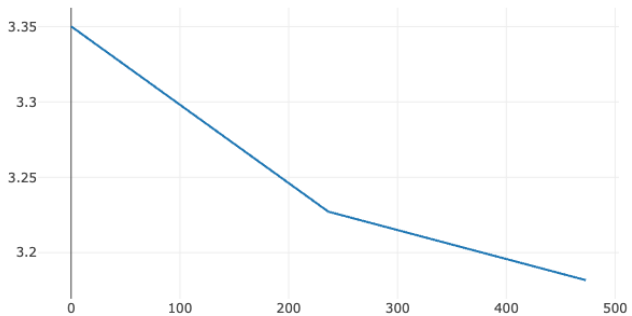
- Blue line refers to the 2^{nd} round model
- Although the comparative figure shows that the second round model is not always inefficient

Larger Dropout (from 0.025 to 0.35) MLP_2

- Train Loss (logs)



Validation Loss (logs)



- As expected, lower efficacy/ciency
- Additional rounds,better results,
- Does not assure it can be get better w.r.t models without dropouts
- MLFlow metrics

▼ Metrics

Name	Value
train_loss	3.201
validation_bacc	0.33
validation_loss	3.182

Comparative: First and Second round with high dropout

- More efficient second round, lower loss (train and validation) and slightly higher balanced accuracy

Metrics

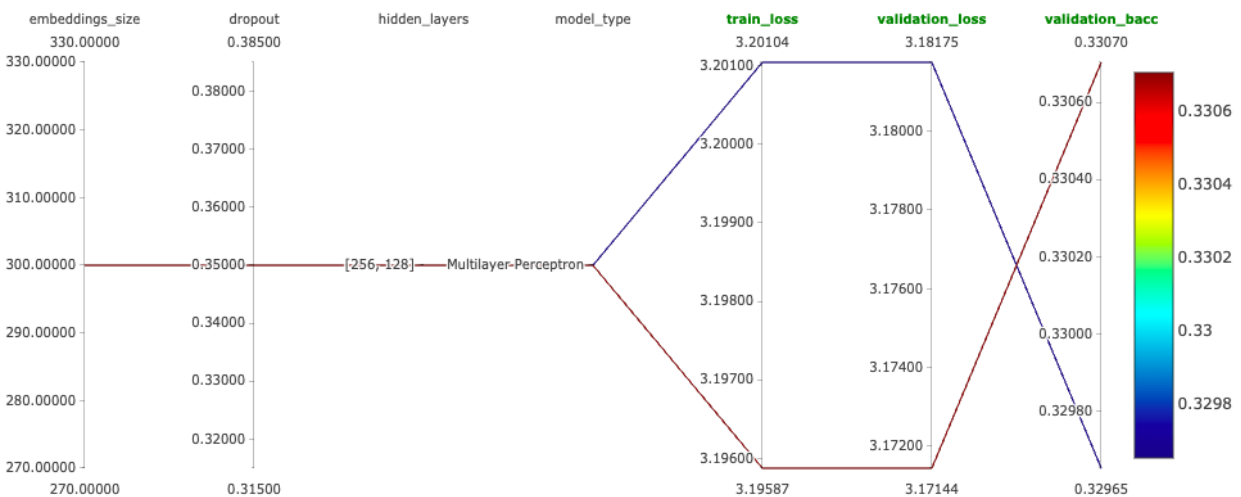
train_loss	3.196	3.201
validation_bacc	0.331	0.33
validation_loss	3.171	3.182

Scatter Plot | Contour Plot | Parallel Coordinates Plot

Parameters: embeddings_size × dropout × hidden_layers × model_type ×

Metrics: train_loss × validation_loss × validation_bacc ×

- train_loss
- validation_bacc
- validation_loss

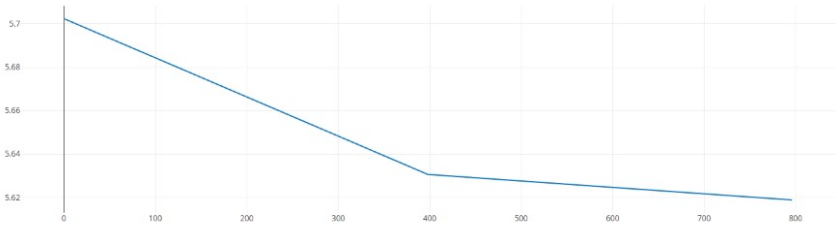


CNN

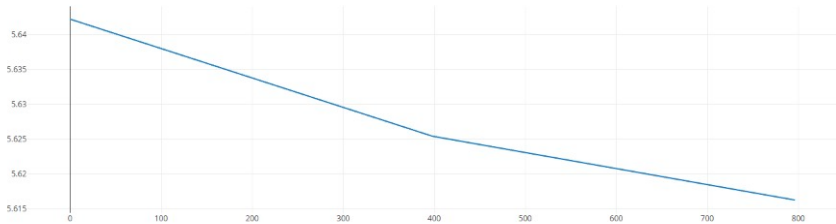
- We run various CNN models, but only one was finished
- The accuracy of this run was not much better than the ones of MLP
- The validation and train losses do not improve with CNN either

CNN_1

Train Loss



Validation Loss



MLFlow results

▼ Parameters

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
model_type	CNN

▼ Metrics

Name	Value
train_loss	5.619
validation_bacc	0.346
validation_loss	5.616

Unfinished (process killed before we got the better of it

- Accuracy and losses are worst than the finished CNN version, and, therefore, than of the MLP models previously presented

▾ Parameters

Name	Value
dropout	0.3
embeddings	./data/SBW-vectors-300-min5.txt.gz
embeddings_size	300
epochs	3
model_type	CNN

▾ Metrics

Name	Value
train_loss	5.702
validation_bacc	0.299
validation_loss	5.642

A lot of work to do

- The Lab, as well as the course, opens the mind to a new world
- Access to more resources could have improved our findings