Study doc: embedding size experiment

how does the size of the embedding space effect the triplet loss

Fabian Gröger
fabian.groeger@stud.hslu.ch

Monday 11th May, 2020

Abstract

The experiment aims to show the effect of the size of the last dense layer from the embedding architecture, further called the **embedding size**.

1 Introduction

The size of the embedding space is essential for the performance, since choosing the wrong hyperparameter can lead to over- or underfitting of the model. The size defines how many dimensions the resulting embedding space has. Therefore if this parameter e is selected to be too big, the model almost certainly overfits, because the model has many options to project the input data onto the embedding space. However, if e is chosen to be too small, there is not enough room to project inputs in different regions. This experiment aims to search an optimal parameter for e.

2 Hyperparameters

The hyperparameters used for this experiment are shown in table 1. The experiment will be conducted using a state of the art ResNet18 architecture on the DCASE dataset. The hyperparameters in section *Feature representation* as well as the sample rate are the default ones proposed by the organisers of the DCASE challenge within the baseline project. The embedding size e will be evaluated for four different values `[2, 16, 32, 64]`.

Table 1: Hyperparameters used for the experiment

Hyperparameter	**value**
Dataset	DCASE
Model	ResNet18
Epochs	20-50
Batch size	64
Optimizer	Adam
Learning rate	1e-5
Margin	1.0
L2 regularisation amount	0.1
Embedding dimension	**[2, 16, 32, 64]**
Prefetch batches	Autotune (-1)
Random selection buffer	32
Shuffle dataset	True
Random seed	1234
Multi threading	
Number of generators	16
Number of parallel calls	16
Audio sample	
Sample rate	16000
Sample size	10
Sample tile size	5
Sample tile range	5
Convert to mono	True
Feature representation	
Feature extractor	LogMelExtractor
Frame length	480
Frame step	160
FFT size	1024
Number of Mel bins	128
Number of MFCC bins	13

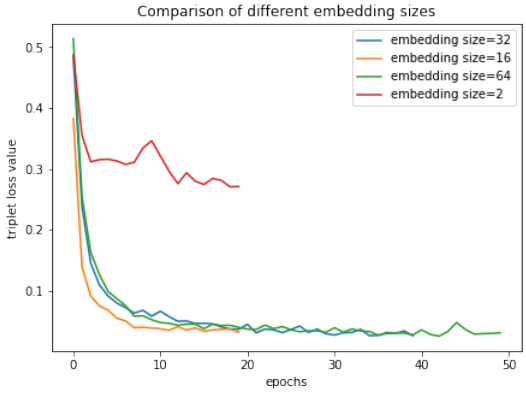

Figure 1: Plot of the triplet loss of the different embedding sizes

Figure 2: Plot of the resulting embedding space of size 16

3 Results

Four models with the same hyperparameters, shown in table 1, were trained for a different amount of epochs. The training was stopped when no more learning was observed.

Comparing different embedding sizes is pretty hard since most of the metrics in the thesis focus on distances between embedding points. In higher dimensional embedding spaces, distances have a different scale and different meanings. This is especially true if small embedding sizes, such as `2`, and large sizes, such as `64`, are compared with each other. Therefore a simple classifier was trained on the resulting embedding spaces, and the metrics of the classifier was compared to find the optimal parameter. To further compare the embedding spaces, they were visualised using the Tensorboard Embedding Projector and manually compared with each other.

The figure 1 shows the different triplet loss values of the embedding sizes. The embedding size `2` has a significantly higher value than other embedding sizes, which shows that in the two-dimensional embedding space, it is a lot harder to project the data points apart from each other. Whereas in high dimensional embedding spaces, the model can easier build clusters.

The plot further shows that the loss value of the embedding sizes `16, 32, 64` are relatively similar and are therefore further compared by examining their resulting embedding space. Which is shown in figure 2, 3 and 4. The result shows that there are vast differences in the embedding spaces, even though the triplet loss value is not that different. The embedding size of `16` shows only approximately four resulting clusters, which indicates a noisy embedding space where small classes are not well separated from each other. The embedding space of `32` and `64` show significant more resulting clusters and they result therefore in a better embedding space. However, it is to say that both embedding spaces further have more noise in it than the lower-dimensional space.

The line plot 5 shows the resulting F1 score when a simple logistic classifier is trained on top of the resulting embedding space. Since the DCASE dataset is heavily unbalanced, the F1 score is compared. All of the classifiers are trained for 20 epochs using the same parameters as the one in table 1. The figure 5 shows that the F1 score of the embedding space `16` is the highest out of the three.

The line plot 6 shows the sparse categorical cross-entropy loss value of the embedding spaces `16` and `64`.

4 Conclusion

The figure 1 shows that the smallest embedding size can be omitted since it has the highest

Figure 3: Plot of the resulting embedding space of size 32

Figure 4: Plot of the resulting embedding space of size 64

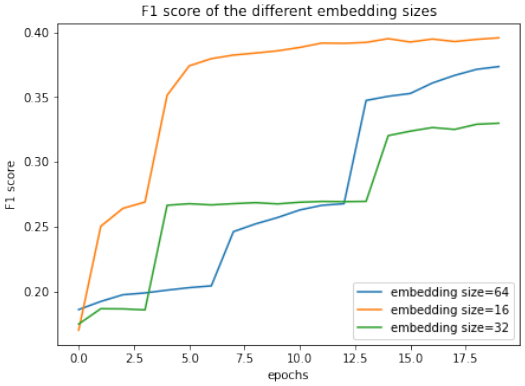

Figure 5: Line plot of the F1 score from the different classifiers trained on top of the resulting embedding spaces

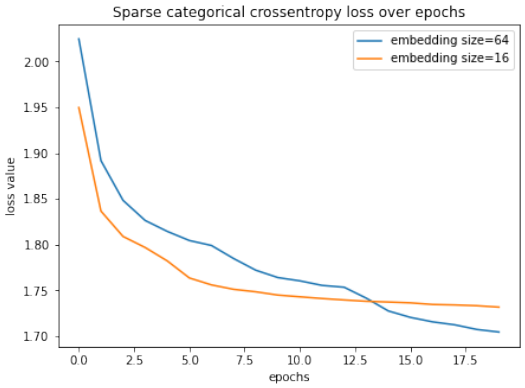

Figure 6: Line plot of the sparse categorical crossentropy loss of the embedding size 16 and 64

triplet loss value significantly. The other three embedding spaces have quite similar values and are, therefore, further compared. The trained classifier on top of the embedding space shows (figure 5) that the embedding size `32` can be omitted since it has a significantly lower score than the others. The figure 5 shows, that the embedding size `16` has achieved the highest F1 score of approximately `0.39`. However, the figure 6 shows that the loss value of the embedding size `16` converges, which indicates that the training process is finished and the classifier will not show any improvements when training longer. The embedding size `64` has a lower F1 score, but the loss value is still decreasing at the end of epoch 20, which indicates that the model can benefit from further training. Further training will increase the F1 score until it converges.

Because of this result, the optimal embedding size, out of the four, is `64`, since it has an optimal triplet loss value, a high enough F1 score and the loss value still decreases after 20 epochs of training the classifier.

5 Next steps

This experiment showed that changing the dimension of the embedding space results in sig-

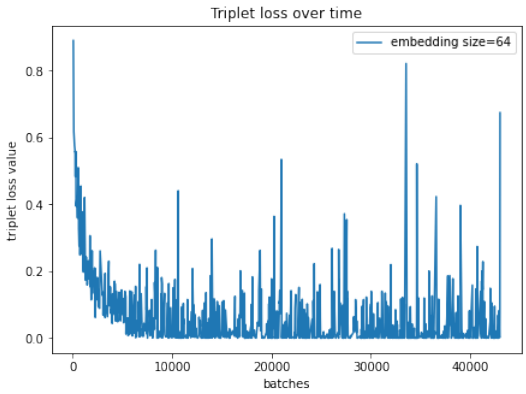

Figure 7: Plot of the triplet loss of the embedding space 64

nificant different embedding spaces and therefore resulting clusters. The experiment should further be conducted for the music dataset because this parameter highly depends on the underlying structure of the data.

For the next experiments, the embedding space size `64` is chosen. The embedding space should also be evaluated on significant higher spaces, such as `256` or `512`.

The experiment further showed another important conclusion, that the longer the embedding space is trained, the more the loss value oscillates which indicates that a learning rate decay should be used to reduce the learning rate over time (7).