LLM - HW Word2Vec Report

Alexandre NGAU - PSL Research University

November 8, 2023

Answers to "Preliminary Questions"

Question 1 : Similarity in Word Embeddings

Given a word w and a context c, we consider the similarity measure $\sigma(c \cdot w) = \frac{1}{1+e^{-(c \cdot w)}}$. Since we want to minimize the loss (1) :

$$L(M(w,c)) = -1_{c \in C^+} \log(\sigma(c \cdot w)) - 1_{c \in C^-} \log(1 - \sigma(c \cdot w)) \quad (1)$$

(a) For a positive context $c \in C^+$, we aim to **maximize** $\sigma(c \cdot w)$. This is in line with word2vec objectives, reducing the vector distance between word and context in the embedding space.

(b) For a negative context $c \in C^-$, we aim to **minimize** $\sigma(c \cdot w)$. This enhances the model's ability to distinguish a word from its unrelated contexts, augmenting the vector distance between word and unrelated context in the embedding space.

Geometrically, this corresponds to adjusting the vector representations such that positive contexts are pulled closer to the word vector, while negative contexts are pushed away, in the high-dimensional embedding space.

Question 2 : Chopra, Hadsell, and LeCun [1]

Contrastive Learning is, simply put :

(a) Contrastive Learning is the fact of learning a similarity metric to differentiate between similar and dissimilar data pairs, mapping inputs into a space where the distance mirrors semantic differences — the lower the distance, the more similar the data pair is, and *vice versa*.

Page 3 of [1], we can find the expression (2) :

$$L(W,(Y,X_1,X_2)^i) = (1-Y)L_G(E_W(X_1,X_2)^i) + Y L_I(E_W(X_1,X_2)^i) \quad (2)$$

Hereafter are the analogies that can be drawn with our setup :

(b) *Analog of* Y : A label indicating the context's nature (positive or negative) for the word (1-Y being the opposite label).

(c) *Analog of* E_W : The neural network weights, or embeddings.

(d) *Analogs of* L_G *and* L_I : Loss functions for positive and negative pairs, influencing the similarity scores in the embedding learning process.

Implementation of the Word2Vec Model

For the implementation of the Word2Vec model that can be found in the associated notebook `hw2_word2vec_alexandre_ngau.ipynb`, few global variables were chosen :

- `n_samples=5000` as suggested in the homework
- `batch_size=64` in order to have a batch size of a little over 10% and better learn from the data
- `n_epochs=100` (with checkpoints every 10 epochs) in order to observe possible overfitting and use a previous checkpoint model
- `R=4` and `K=6`, according to the 5-20 range stated by Mikolov et al. [2], with the intuiton that `K`>`R` would better separate the random words from the semantic clusters
- `embedding_dimension=150`, in order to have enough depth to cover the complexity of the data, without compromising the computation time, as stated by Hofstra et al. [3] for who 100, 200 and 300 produced robust results

Other than that, the global imposed code structured was followed. Specific coding choices are however commented within the associated ***`hw2_word2vec_alexandre_ngau.ipynb`*** *notebook.*

The training results of the Word2Vec model can be seen on Figure 1 :

(a) **training and validation metrics** vary very closely to one another

(b) **convergence rate** abruptly slows down after about 30 epochs

(c) **training improvement** reaches its asymptotic value (represented by the dotted red line) after about 50 epochs

These observations show that the training of the model is **efficient** (a, c), **rapidly converging** (b), and show **no sign of overfitting** after 100 epochs (a, c). The training is thus satisfying, and could even have been stopped at 50 epochs, following observation (b).

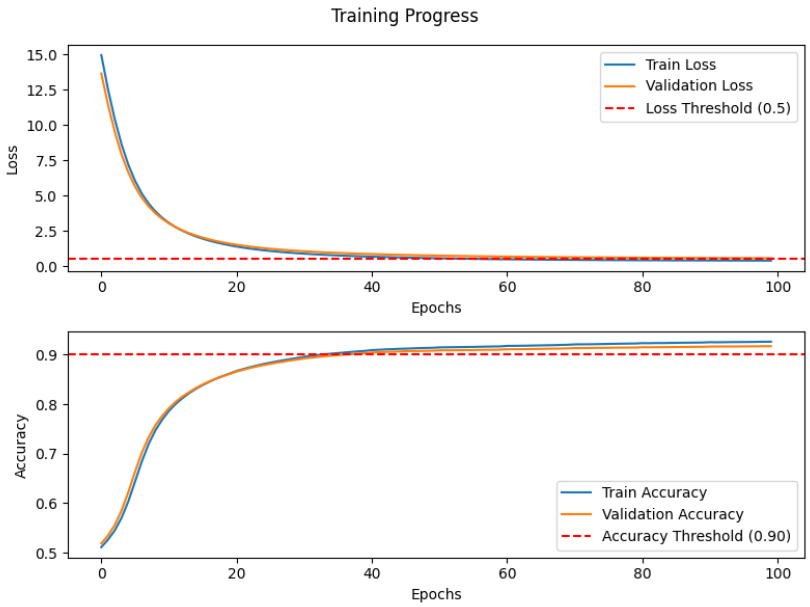


Figure 1: Training Progress of the Word2Vec Model as a Function of the Epochs

Implementation of the Classification Task

The implementation of the Classification Task can be found in the associated notebook `hw2_convolutionnal_classification_alexandre_ngau.ipynb`.

The training results of the Conv1D model initialized with the Word2Vec (w2v) embeddings, and with random ones, can be read on Figure 2 :

(a) for the two models, training and validation metrics vary in the same way

(b) the training loss of the w2v-initialized model decreases slower, but its training accuracy increases quicker (reaching asymptotic value after epoch 40)

(c) the validation metrics seem to tend to a rather equal asymptotic value between the two models (although for the accuracy it may take more epochs)

These observations show that the w2v-initialized model **learns faster** (b), although the two model seem to **asymptotically tend to the same value** after 50 epochs (a, b, c). Nonetheless, the fast converging training accuracy of the w2v-initialized model after 40 epochs strongly suggest that its performances are **better than the randomly-initialized one for a small number of epochs**.

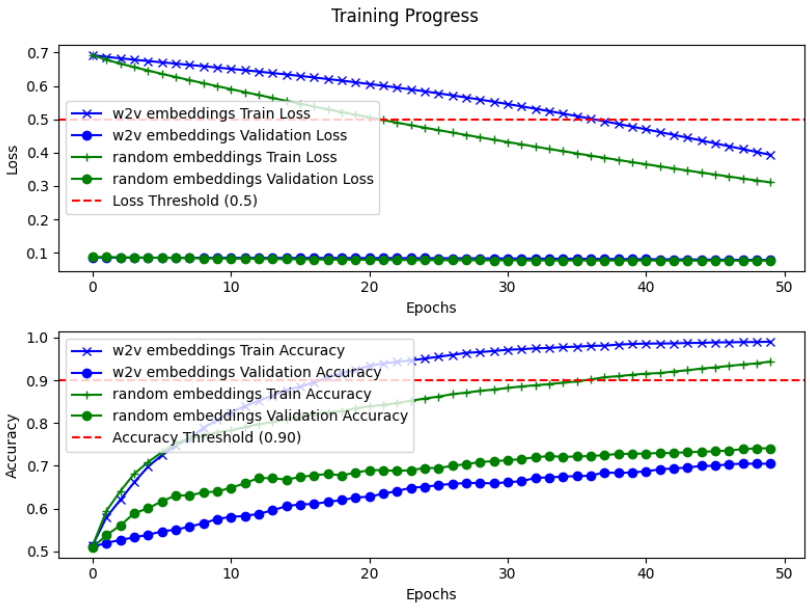


Figure 2: Training Progress of the Classification Models as a Function of the Epochs

Ablation Study on the Number of Epochs of the Training of the Word2Vec Model

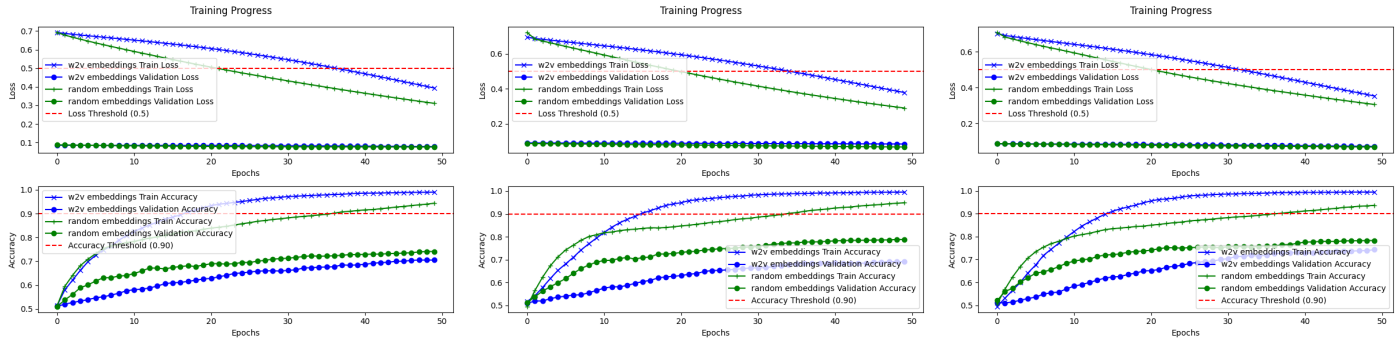
The aim is to observe the impact of the number of epochs in the training of the Word2Vec model – 30, 70 and 100 epochs – on the classification task, while preserving the other parameters fixed as in the original Word2Vec model trained in this report. Figure 1 shows that the convergence is first reached around 30 epochs, and well established around 70 epochs, thus the choice of these three values (100 epochs being the original model).

As can be seen on Figure 3, the overall appearances of the graphs are the same as on Figure 2. However, it seems that the more the number of epochs, the faster the convergence of the training accuracy of the w2v-initialized model towards its asymptotic value is achieved, suggesting better performances the more there are epochs in the training of the w2v model.

Ablation Study on the Embedding Dimension of the Word2Vec Model

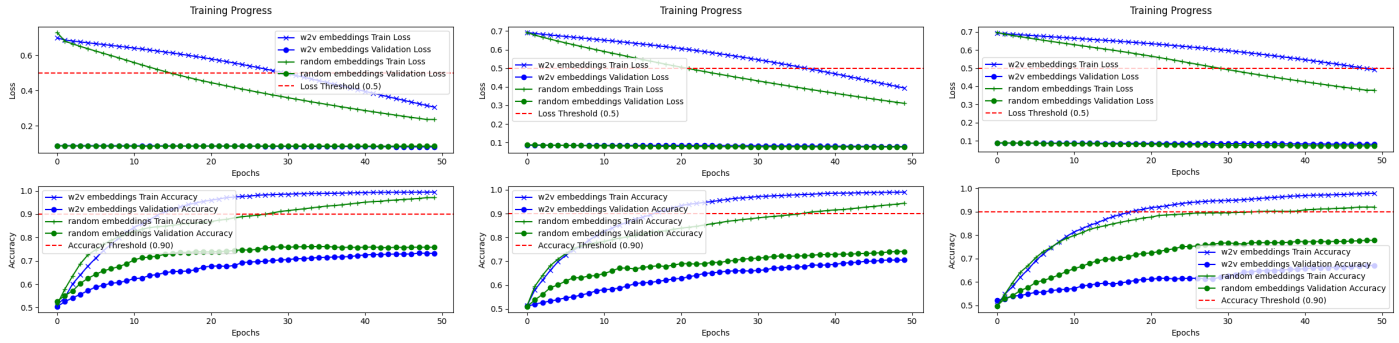
The aim is to observe the impact of the embedding dimension in the training of the Word2Vec model – 100, 150 and 200 – on the classification task, while preserving the other parameters fixed as in the original Word2Vec model trained in this report. Hofstra et al. [3] states that embedding dimensions between 100 and 300 produce robust results, thus the choice of these three dimensions around 150 (dimension of the original model).

As can be seen on Figure 4, the overall appearances of the graphs are the same as on Figure 2. However, it seems that the bigger the embedding size, the faster the convergence of the training and validation accuracy of the w2v-initialized model towards its asymptotic value is achieved, suggesting better performances the higher the embedding dimension of the w2v model.



(a) w2v Model Trained on 100 epochs (b) w2v Model Trained on 70 epochs (c) w2v Model Trained on 30 epochs

Figure 3: Training Progress of the Classification Models as a Function of the Epochs



(a) w2v Model Trained on 200 embedding dimensions (b) w2v Model Trained on 150 embedding dimensions (c) w2v Model Trained on 100 embedding dimensions

Figure 4: Training Progress of the Classification Models as a Function of the Epochs

What I learnt during this homework

During the course of this homework, I better understood how to use PyTorch, and how to structure the training procedure of a model, more specifically the rigorous attitude one has to have towards its code for efficient reading and running.

I also had the opportunity to refine the architecture of my `training` and `validation` functions, that I can now reuse whenever I want to train another model.

Finally, I got a better understanding of what embeddings were, and how to embed words and use these embeddings in classification tasks such as the one implemented in the homework.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[3] B. Hofstra, V. V. Kulkarni, S. M.-N. Galvez, B. He, D. Jurafsky, and D. A. McFarland, "The diversity–innovation paradox in science," *Proceedings of the National Academy of Sciences*, vol. 117, no. 17, pp. 9284–9291, apr 2020. [Online]. Available: https://doi.org/10.1073%2Fpnas.1915378117