Homework 3

Anna Braghetto - 1205200

16/02/2020

Introduction

The aim of the proposed homework is to build, train and test a recurrent neural network with PyTorch that generates a sequence of words given an arbitrary initial seed; the learning is performed at word level using the *Word2Vec* embedding. In particular, the structure and, in general, the hyperparameters of the considered network, are looked for by performing a grid search, implemented with the k-fold cross validation. Then, a smart method to establish when learning should stop is implemented. Since, the needed computational power to perform the grid search, and in general the training, is too high to use CPU, everything is perfomed with CUDA in Google Colaboratory.

Dataset

The dataset is composed by five different books written by Joseph Conrad, titled *Chance, The Mirror of the Sea, Heart of Darkness, Lord Jim* and *The Shadow-Line*.

In order to make the learning easier and faster, a preprocessing on the dataset is performed and then, the full text is divided into sequences whose number of words is equal to the crop length, that is fixed to 20.

Preprocessing

At first, the text of the five books are stack up in an unique one. Then, to avoid multiple definitions for the same word, all the letters have been transformed from upper case to lower case and almost all the punctaction is discarded, except for the periods and the commas.

Futhermore, in order to keep the wanted punctaution in the *Word2Vec* embedding, commas and points are separated from the text and translated into inveted words that describe them, as follows.

Punctuation	**Describing word**
,	*commapunct*
.	*pointpunct*
!	*exclapunct*
?	*questpunct*

Table 1: Change of punctuation.

After a brief preprocessing, the dataset is built by getting the list of paragraphs that have at least the crop length; for each paragraph, each sentence with length equals to the crop length is extracted, without overlaps: this permits to build a dataset with an high number of sequences (21665).

The dataset is then divided into training (80%) and validation (20%) sets, and it is trained using batches: the batch size is set to 512 for the whole training.

Word2Vec

Since the sequence modelling of the recurrent neural network is performed at word level, it is necessary to perfom the word embedding that converts words into vectors. This task is achieved by using the library provided by Gensim: *Word2Vec*. At first, each word is associated to a number, called *index*, that is mapped into a vector in a n-dimensional space, where neighboring vectors correspond to words with similar meaning. In particular, the dimension n correspond to the embedding size that is set to

100. After the training of the embedding model, it is interesting to show the results, in Fig. 1. It is applied the PCA to the vectors, keeping just two components in order to observe the embedded words in a two-dimensional plane.

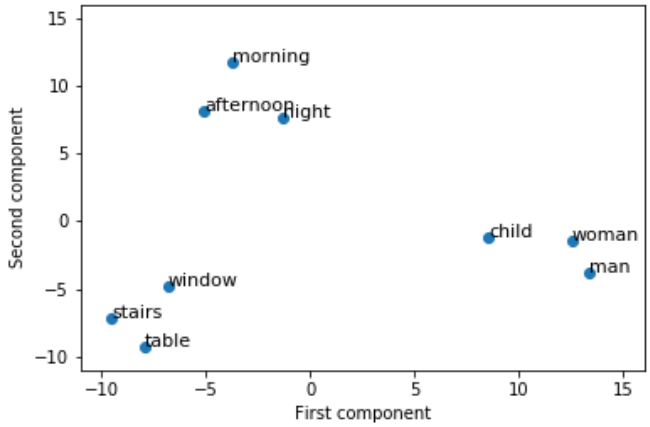

Figure 1: PCA on the *Word2Vec* embedding.

In Fig. 1, it is possible to notice nine words belonging to three different sample arguments: moments of the day (*morning, afternoon, night*), objects of the house (*table, window, stairs*) and people (*woman, man, child*). Thus, neverthless the PCA just keeps 2 components, that is very smaller than the embedding size (100), the result is satisfactory, showing, as expected, that the words are clustered based on their meaning.

Model

The model to train, is built by definying the embedding layer, the *LSTM* module and then the linear layer.

The embedding layer permits to perform the conversion from the word (in particular the decoded one, i.e. the index) into the vector obtained throught the *Word2Vec* model. Furthermore, the weights of this layer are set as no-trainable, meaning that their values are kept fixed and never updated during the training.

The *LSTM* module has the following hyperparameters: number of layers, number of hidden units and dropout probability. In particular, just the number of layers and the number of hidden units are looked for with the grid search, while the dropout probability is set to 0.3.

The final layer corresponds to the output layer with a number of units equal to the vocabulary lenght (17246). The output of this layer is then used to make the prediction and build the predicted text.

Training

The training is performed using, as loss function, the *Cross Entropy Loss*, computed taking into account the prediction not just on the last words but on all, and the *Adam Optmizer* with the weight decay for the L_2 *regularization*, set to $5 \cdot 10^{-4}$, while the learning rate is kept fixed to 10^{-3}.

Furthermore, to avoid overfitting, an early stopping is also implemented: if the validation loss increases for many consecutive epochs, the training stops.

Prediction

Given the initial seed, the predicted next word is obtained by sampling from the softmax distribution over the output layer.

Choice of the Model

In order to choose the best structure, a grid search with a k-fold cross validation is performed. In particular, the hyperparameters left to vary are the number of layers and the number of hidden units of the recurrent neural network: the considered values are the following.

- Number of layers: 2 and 3
- Number of hidden units: 32, 64 and 128

The search of the best model is limited to a low number of parameters due to the computational power needed to train this kind of networks and due to the size of the dataset, bigger structures are not taken into account.

The dataset is divided into 3 different folds: for each fold the model is trained, for 100 epochs, in the union of the other 2 and validated in the selected one. The best model is chosen by looking both at the average validation loss and at the prediction. In particular it is observed that by increasing the number of layers the performance does not improve so much (losses with different layers but same hidden units are very similar): the simplest structure with less layers is chosen in order to reduce the computational cost, as reccommended by the Occam's Razor principle.

Thus, the parameters of the best model are: 2 layers with 128 hidden units.

Results

Defined the best structure obtained through the grid search, implemeted with the k-fold cross validation, it is possible to perform the final training.

In particular, the final training is performed with the parameters defined above and summarized in Tab. 2.

Layers	Hidden Units	Learning Rate	Weight Decay	Epochs	Crop Length
2	128	0.001	0.0005	500	20

Table 2: Parameters of the final training.

In Fig. 2, the loss for the training and validation dataset are shown and it is possible to observe that the loss decreases and there is not overfitting.

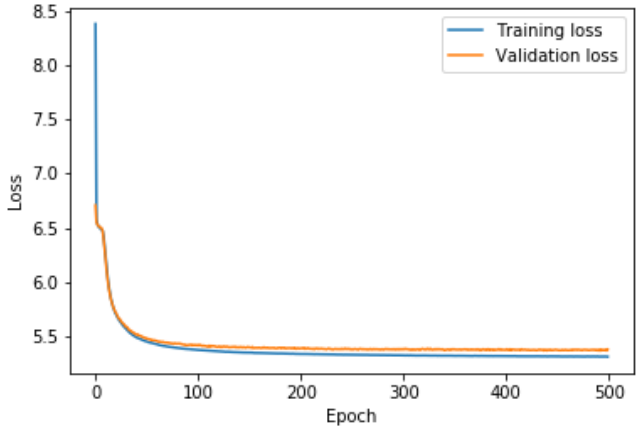

Figure 2: Training and validation loss.

The results of the prediction are shown below, given the input highlighted seeds and n number of predicted word set to 30.

- **The people want** to have to only the fynes, that she had restrained in him. she saw it a late elected because he had not been assumed. a very time near the
- **There were** something in the truth of lease. extended him its eyebrows open me to be eavesdropper. it was mrs too, burned, good effacing, senselessly and provided to the
- **She was going to** say its pacific during the ground. he was in the house. he said well that he asked me when he could seen. low hand from the arms of the

As shown in the results above, sometimes the prediction is not meaningfull but contains a correct syntactic structure and the result is satisfactory.