Homework n°2

Yassine Kadiri

NLP and Representation Learning DS-GA 1011

More detail provided on the Jupyter Notebook that can be found at github.com/yasskad

Table des matières

1 Preparing the data

I tokenized the data using similar functions to the ones used in homework 1 and lab 3. The rest of my choices for preparing the data is detailed on the Jupyter notebook.

2 Hyperparameter exploration for GRU

The following hyperparameters are fixed and I didn't change them during my runs

Learning rate = 3×10^{-4} Number of epochs = 10 Criterion : Cross Entropy Loss
Optimizer : Adam

First I started exploring the influence on validation accuracy of hidden sizes. Then I tried two dropout rates to see the influence of dropout on the validation accuracy.

2.1 First run/Epochs hyperparameter search

Here I was mainly trying to confirm that the hyperparameters above were giving a good-enough accuracy, and it was the case. Also, I wanted to see if 10 epochs were enough or too much (overfit risk), the results conforted me in that choice. However, the loss plot shows that the validation loss goes up again, meaning that there's a risk of overfit after 10 epochs.

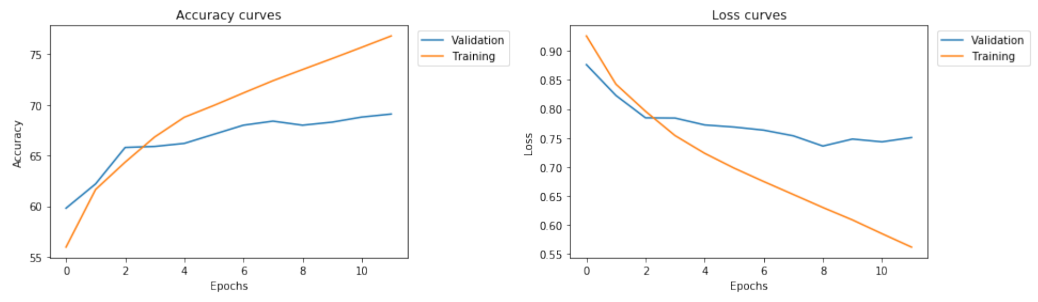

FIGURE 1 – Influence of epochs on validation and training loss and accuracy

2.2 Linear hidden size and GRU hidden size

I then started exploring the influence of the GRU hidden size, i.e. the hidden size used within the GRU, and then the linear hidden size, i.e. the hidden size used for the 2 fully connected layers. For that I covered the following ranges :

$$\text{GRU hidden sizes} = [128, 256, 512] \quad \text{Linear hidden sizes} = [16, 32, 64, 128]$$

For 512, due to computing and time constraints, I only could compute the linear hidden size 16.

Here are the results :

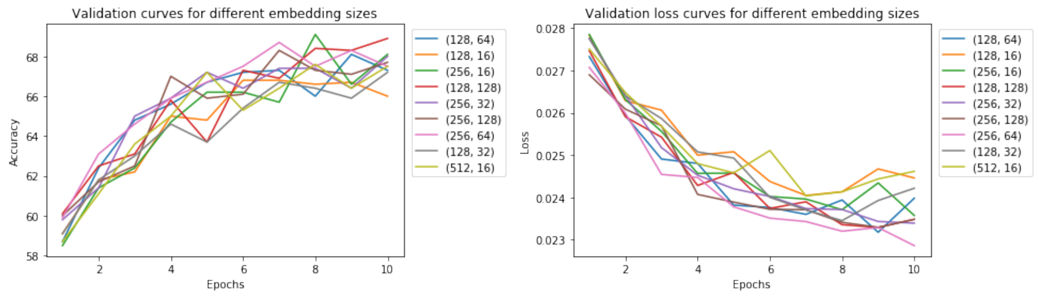

FIGURE 2 – Influence of hidden_sizes size on accuracy and loss

It is quite difficult to tell what makes the best performer but we can just see that the setting (128, 128) works quite well and after 9 epochs already outperforms all the other settings. More comparisons available on the notebook.

2.3 Dropout

Since I already tried by default a dropout of 0, I just wanted to check if taking another value would improve the results and it didn't. I chose to test dropout=0.5 and this is the comparison between both settings :

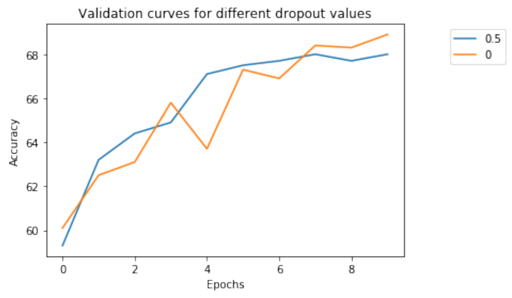

FIGURE 3 – Influence of dropout on accuracy for the GRU

We see that the curve is smoother but in the end, after 7 epochs, the 0 dropout is a better performer and I'll therefore continue with it.

3 CNN hyperparameters

3.1 Elementwise multiplication vs Concatenation

Here, I compared the influence of the method used on the encoded sequences before making them an input for the two connected layers.

Results here :

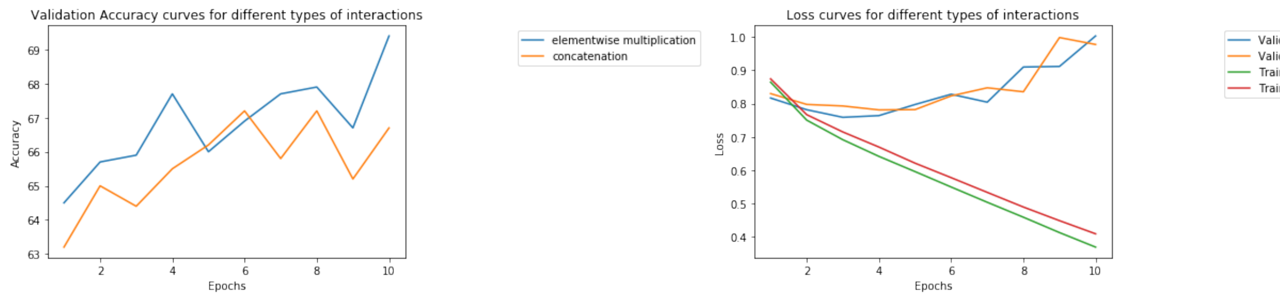

FIGURE 4 – Influence of method used on encoded sequences

We see that elementwise multiplication yields better results.

3.2 Number of epochs

While for GRU 10 epochs seemed appropriate, here we notice that more than 7 epochs leads to overfitting the training data since the validation loss increases after having decreased in the first 4-5 epochs. Therefore, we'll stick with 7 epochs.

3.3 Hidden sizes

Here, I didn't use a gridsearch kind of approach like for GRU but rather tried each separately taking the best at each try.

I tried the following three values, while linear hidden size was set to 128 :

$$\text{CNN\_hidden\_size} = [32, 64, 128]$$

We see in the first 5 epochs that increasing this parameter increases accuracy, then the 64 choice outperforms the others and this is the one we'll use for remaining tests.

With the CNN hidden size set to 64, I just tried the following linear hidden sizes :

$$\text{Linear hidden size} = [32, 64, 128]$$

And here are the results :

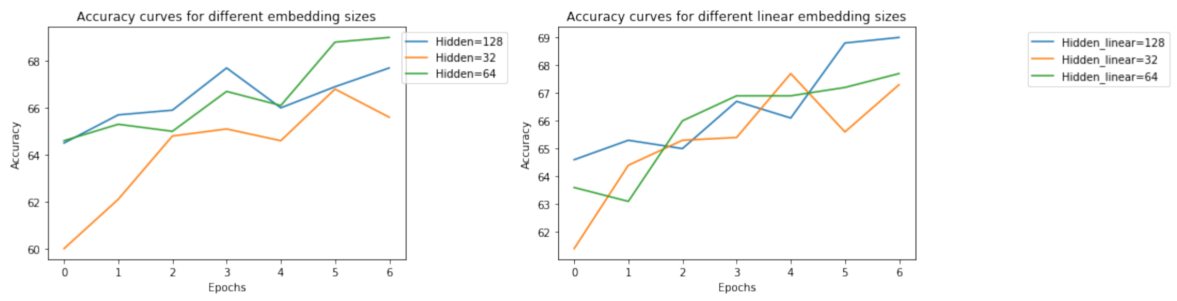

FIGURE 5 – Influence of CNN (left) and linear (right) hidden sizes on the model with CNN

Therefore, the choice is to go for the setting $(64, 128)$

4 Testing it on MultiNLI

4.1 CNN encoder

Here are the results for the best CNN encoder on the validation data for MultiNLI.

	Fiction	Telephone	Government	Slate	Travel
Accuracy (in %)	45.33	43.18	43.31	42.02	43.08
Loss	1.27	1.33	1.94	1.60	1.67
Lift	1.36	1.30	1.30	1.26	1.3

We see that the results are not that good, however, considering random guessing as the baseline, we still do better. (Lift of approximately 1.3) Most of all, the model performs similarly across the genres, which seems logical since we have a model trained to detect the label better than the random baseline, however, this model isn't trained for specific genres.

4.2 GRU encoder

Here are the results for the best GRU encoder on the validation data for MultiNLI.

	Fiction	Telephone	Government	Slate	Travel
Accuracy (in %)	44.92	43.58	43.41	41.92	43.58
Loss	1.21	1.23	1.20	1.24	1.24
Lift	1.35	1.31	1.30	1.26	1.31

We see that the results are a almost the same in terms of lift to the ones achieved with the CNN encoder.