Text Classification with Deep Learning

National Technical University of Ukraine “Igor
Sikorsky Kyiv Polytechnic Institute
Speaker: D. L. Lazarenko
Supervisor: Ph.D. in Physico-mathematical
Science, docent A. Maltsev

Kyiv, 2018

Aim of this thesis is to build an effective model which would have high accuracy and appropriate speed for classification of advertisements at the e-commerce platform

Object of study is advertisements at e-commerce platform

Subject of study is classification model for advertisements:

Relevance of the problem

- e-commerce sales are quickly increasing
- large online e-commerce websites serve millions of users' requests per day
- make the processes of registrations and purchases as much convenient and fast as possible.
- users have to make a choice from more than hundred categories
- automatic category prediction is very important in terms of saving moderators' time and as a result, decreasing the number of necessary moderators to process them

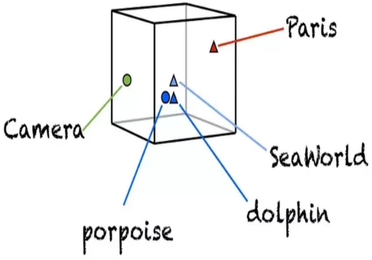
Structure of the data files

lvl2	titles	descriptions
29	Clean Toyota Camry 2008 Silver	Fairly used Toyota 08 Camry with no problems V4 engine fabric seats and interior
25	Look Unique	Nice, quality, adorable,unique dress available now, whatsapp me

Let's assume we have the following sentences:
["The sun is yellow", "The sky is blue"]
Encode words with the Bag-of-words method

Text	the	sun	is	yellow	sky	blue
T_1	1	1	1	1	0	0
T_2	1	0	1	0	1	1

1. Naive Bayes
2. Logistic Regression
3. Support Vector Machines (SVMs)
4. Decision Trees and Random Forests

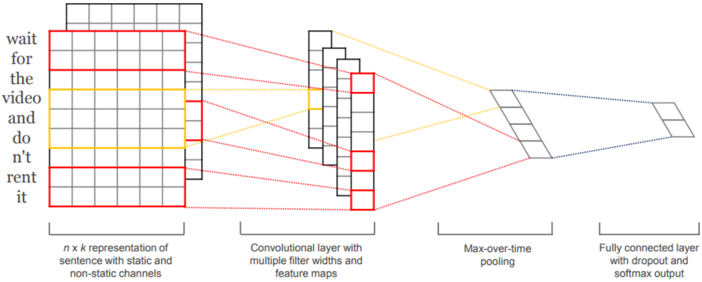


An **embedding** is a mapping from discrete objects, such as words, to vectors of real numbers. For example, a 300-dimensional embedding for English words could include:

blue: $(0.059, 0.7597, \ldots)$

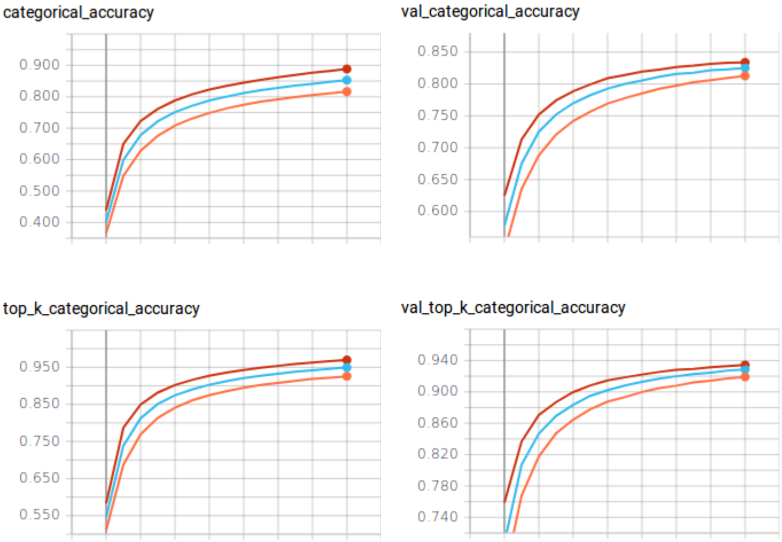
Metric	**Train**	**Test**
categorical accuracy	0.7975	0.8203
category cross entropy	0.8532	0.7478
top k accuracy	0.9189	0.9219.

Convolution Neural Network



- 128, 256, 512 filters
- size of filter: 3, 4, 5
- dropout equals to the rate 0.5

Overfitting

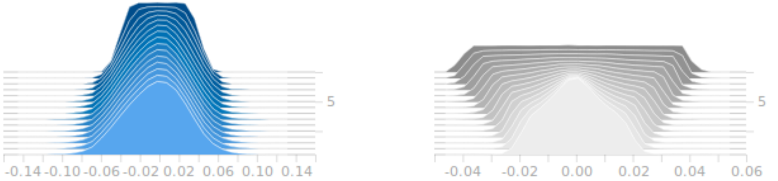


— 512 — 256 — 128 filters

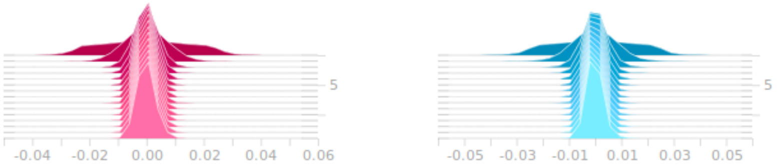
Modifications:

1. Dropout rate decreased in two times.
2. l2-regularization equals to 0.01 both for convolution layers and dense layer. Dropout = 0.5 both for dense and convolution layers. training algorithm - Adam with learning rate 1e-4.
3. l2-regularization equals to 1e-3. Dropout = 0.25 both for dense and convolution layers.
4. l2-regularization = 0.001 for convolution layers and 0.01 for dense layers. Dropout rate = 0.25 for convolution layers and 0.5 for dense layers.

Figure: *Histogram of dense layers*

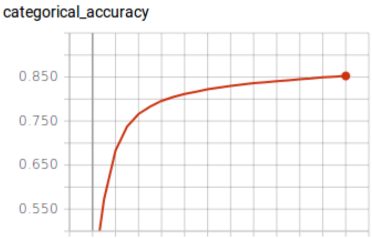


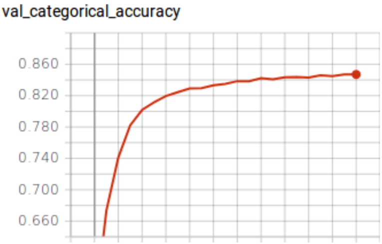
1, 2

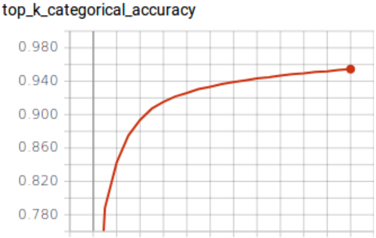


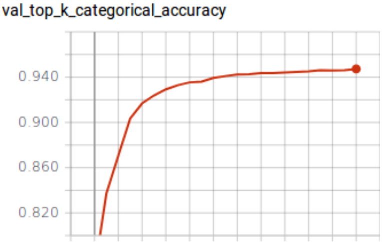
3, 4

Results


categorical_accuracy
0.850
0.750
0.650
0.550


val_categorical_accuracy
0.860
0.820
0.780
0.740
0.700
0.660


top_k_categorical_accuracy
0.980
0.940
0.900
0.860
0.820
0.780


val_top_k_categorical_accuracy
0.940
0.900
0.860
0.820

Results

Metric	Train	Test
categorical accuracy	0.8250	0.8307
category cross entropy	0.5800	0.6612
top k accuracy	0.9545	0.9473

category	precision	recall	f1-score	support
11	0.78	0.58	0.66	623
14	0.93	0.99	0.96	8070
15	0.82	0.68	0.74	362
16	0.93	0.95	0.94	1656
20	0.82	0.91	0.86	1151
25	0.75	0.9	0.82	1910
29	0.97	0.99	0.98	12346

Future work

- train networks with other training algorithms. For example, it is possible to try SGD or RMSprop with appropriate parameters;
- make an assemble of neural networks to best use each one's strong qualities;
- try to use different words sequences length for titles and descriptions;
- as the results on categories were not really impressive it is possible to add them into one large.

Thank you for your attention!