Obtaining word embedding from existing classification model

Martin Sustek and Frantisek V. Zboril

FIT, Brno University of Technology, IT4Innovations Centre of Excellence,
Bozetechova 1/2, 612 66 Brno, Czech Republic,
isustek@fit.vutbr.cz

Abstract. This paper introduces a new technique to inspect relations between classes in a classification model. The method is built on the assumption that it is easier to distinguish some classes than others. The harder the distinction is, the more similar the objects are. Simple application demonstrating this approach was implemented and obtained class representations in a vector space are discussed. Created representation can be treated as word embedding where the words are represented by the classes. As an addition, potential usages and characteristics are discussed including a knowledge base.

Keywords: unsupervised learning, artificial intelligence, word embedding, word2vec, CNN

1 Introduction

Word embedding can be used to represent any entity in continuous vector space with hundreds of dimensions. The entity is typically a word obtained from a large dataset that had been constructed from publicly available corpus (e.g. Wikipedia). The word representations model the language and can be afterward used for different machine learning tasks such as speech recognition. As an addition, from operation over the words represented as dots (or vectors) in continuous vector space can be derived rich information about their relationship. The most popular analogy is mathematically represented sex (king to queen as man to woman) as was shown in [1]. When the corpus is used as dataset in combination with method such as *word2vec*, it creates word embedding reflecting the given text, therefore, it might be possible to deduce the information hidden in used text.

Even though the relations captured by this method can be considered surprisingly precise, they are based on the Distributional Hypothesis. Two words tend to have similar meaning if they share the same context according to the hypothesis. Some semantic information, however, is difficult to capture just with the help of the general text corpus and the Distributional Hypothesis. Recently, some authors [2–4] used visual information to overcome the lack of such semantic evidence in the textual data. Provided visual information is typically in the form of sentences describing pictures (e.g. “girl eats ice cream”) or the system is multimodal (uses both text and pictures). The authors afterward demonstrate the improvement on the chosen dataset, therefore, these methods are aimed to bring additional semantic information into encoded word representation. Because they target on natural language processing, it is reasonable to focus on visual information in the form of sentences or build upon existing word representations obtained by text corpus.

2 Goals of the paper

The goal is to present an approach of creating a class representation in a vector space. With this approach, we want to demonstrate the relations between classes on a simple example using convolutional neural networks (CNN). Moreover, we want to discuss important characteristics regarding the learning with the focus on unsupervised learning and the origin of the information (considered to be a knowledge). In the last part, we will propose methods benefiting from the word embedding as an potential future work.

We will present the way of obtaining word embedding from any existing classification model rather than from a corpus. We will focus on models working with pictures, because it can bring some additional benefits that will be discussed below. However, the method is capable of working with any model, not necessarily picture based. We choose convolution neural network (CNN) as an example of existing model. CNN was trained on a dataset in the form of pairs consisting of labelled class and example picture of this class.

It is typical to demonstrate that the approach is useful for some task. Moreover, obtained score can be compared with others. However, we do not present any task or provide any score since the method is intended to inspect the relations between classes in classification model and we do not demand or expect any strict relations between them.

3 Learning the similarity

We decided to use very simple method to create word embedding by assuming that a similarity is related to an error. This means that when CNN finishes learning phase, and given a picture of dog, it is theoretically expected that the probability of seeing a dog $\mathrm{P}(dog|picture) = 100\%$ and $\mathrm{P}(x|picture) = 0\%$ for any class x that is not a dog. In practice, results can be typically $\mathrm{P}(dog|picture) = 90\%$, $\mathrm{P}(cat|picture) = 9\%$, $\mathrm{P}(x|picture) < 1\%$ for any x that is neither dog nor cat. This can be illustrated in Figure 1.

For this method purpose it is actually beneficial that the theoretically desired state is not reached. We suppose that similar approach is natural for humans as well since the question “Does this dog look more like a cat or like a sheep?” can be asked. The answer will probably not be indifferent.

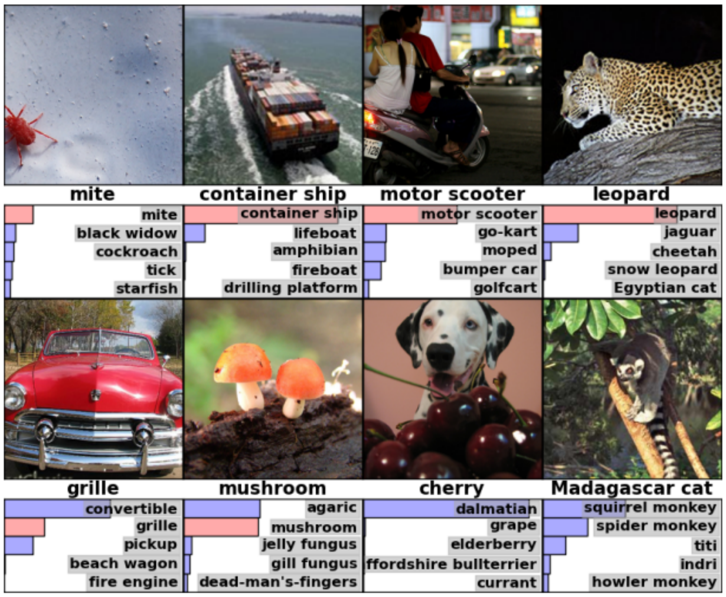

Fig. 1. An example response of existing trained CNN (picture taken from [5]). Below each picture is the label of the expected class and the distribution between classes. The width of the bar corresponds to the probability that CNN assigned to the class. The fact that the probability of seeing jaguar given the picture of leopard P$(jaguar|leopard)$ in the upper right picture is not negligible can be used as a witness of their similarity.

4 Experiments

Firstly, the convolutional neural networks was trained on the task to classify the image. *CIFAR-10* [6][1] was chosen as a dataset for CNN. The dataset consists of colour pictures of 32x32 size divided into following 10 categories (classes):

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck

[1] http://www.cs.toronto.edu/~kriz/cifar.html

Each class includes 5000 training and 1000 testing examples. In order to train CNN, we used publicly available example[2] and about 300 epochs.

After we had finished the learning phase, we used CNN as a classification model to create a new dataset containing pairs in the form of $(input, output)$. For each picture in dataset, 3 most probable classes were retrieved and pairs $(input, mp_1), (input, mp_2), (input, mp_3)$ were constructed, where $input$ is the class of the input picture and mp_k is the k-th most probable class.

A newly created dataset was afterward used as an input of the *word2vec Skip-Gram* architecture to create word embedding of used classification classes as introduced in [1]. The number of dimensions in word embedding was set to 5 since there are only 10 classes. The process is shown in Figure 2.

5 Evaluation

Word analogy is typical measurement for the evaluation of constructed word embedding. It can be good when modelling a language, therefore, learned embedding should hold the information about the relation between words regarding the sex, geographical facts or linguistic regularities. This information can be regain by certain operations over the vector representations.

We do not think that this method is ideal for task such as language modelling at the time. It can be, however, used as an introspection tool and should provide information about *how the model intercepts classes*, there is no ground truth to be expected, therefore, we do not use any method to evaluate results. However, we as humans feel that at least organic and inorganic division can be reflected. We also know that airplane shape was inspired by the shape of a bird. Since there are just 10 classes, the cosine similarity (1) can be computed for each vector pair and visualize the similarity between classes. The cosine similarity express how similar the vectors are and ranges from -1 for opposite vectors, to 1 for vectors sharing the same direction. The cosine similarity should theoretically reach almost 1 for very similar objects. Symbol θ stands for an angle between vectors.

$$\text{similarity}(\mathbf{A}, \mathbf{B}) = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \tag{1}$$

Table 1 and Table 2 display 4 organic and inorganic classes respectively. Each class (column) shows the rest of the categories in descending order based on their similarity. It can be noticed that certain relations are somehow reflected through the similarity. In particular, inanimate classes (airplane, automobile, ship, truck) are the least similar objects to animate classes such as dog, cat or deer. The distinction between animate and inanimate objects remains noticeable in all categories. Cat and dog as well as automobile and truck form a strong connection. Cosine similarity between pairs of four-legged animals (cat, deer, dog, horse) is above the average value. Already mentioned similarity between airplane and bird is reflected as well.

[2] https://www.tensorflow.org/tutorials/deep_cnn

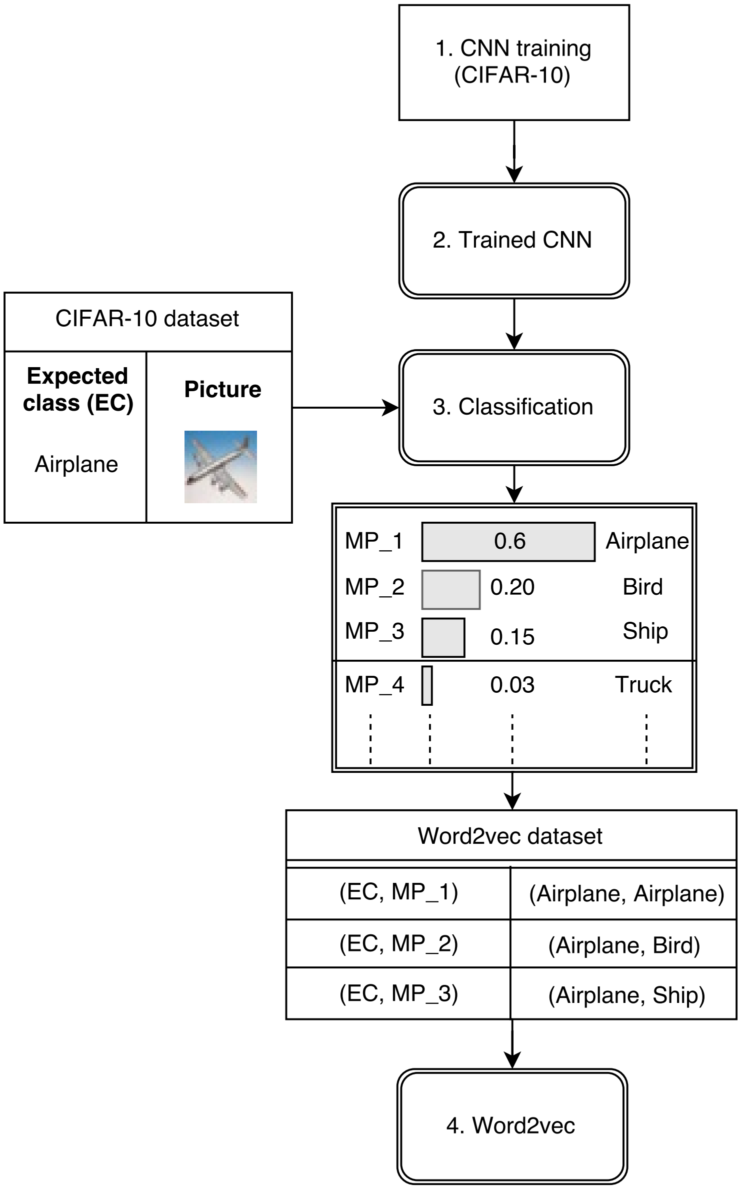

Fig. 2. The illustration of the process of creating word embedding. At first, CNN is trained. Afterward, it is used as classification model on each picture from the *CIFAR-10* dataset to construct a new dataset for word2vec method. Word2vec is learned using the pairs, where the first element is the input and the second is the output.

Table 1. Similarities between organic classes. Each column shows classification class and remaining classes in descending order sorted by the similarity.

dog		cat		deer		bird	
cat	0.9715	dog	0.9715	horse	0.6744	airplane	0.6377
horse	0.4560	frog	0.5029	bird	0.5978	deer	0.5978
frog	0.2944	horse	0.2886	frog	0.3080	frog	0.5888
deer	0.2542	bird	0.2698	dog	0.2542	cat	0.2698
bird	0.1761	deer	0.2455	cat	0.2455	horse	0.2166
ship	0.1397	ship	0.1208	airplane	0.1208	dog	0.1761
automobile	-0.0359	automobile	-0.0258	automobile	0.0726	ship	0.1719
truck	-0.1054	truck	-0.0948	truck	0.0653	truck	-0.1957
airplane	-0.1537	airplane	-0.1357	ship	-0.3449	automobile	-0.2750

Table 2. Similarities between inorganic classes. Each column shows classification class and remaining classes in descending order sorted by the similarity.

automobile		truck		ship		airplane	
truck	0.9877	automobile	0.9877	airplane	0.7608	ship	0.7608
ship	0.2725	ship	0.3723	truck	0.3723	bird	0.6377
frog	0.1371	airplane	0.2558	automobile	0.2725	truck	0.2558
airplane	0.1104	frog	0.1228	bird	0.1719	horse	0.1667
horse	0.0870	horse	0.0888	dog	0.1397	frog	0.1288
deer	0.0726	deer	0.0653	cat	0.1208	deer	0.1208
cat	-0.0258	cat	-0.0948	horse	0.0552	automobile	0.1104
dog	-0.0359	dog	-0.1054	frog	-0.0210	cat	-0.1357
bird	-0.2750	bird	-0.1957	deer	-0.3449	dog	-0.1537

5.1 Learned intelligence

We believe that each model must have some kind of intelligence to be able to give good results. Even if a model would take into account just the colour of the object or in the corner of each picture would be written object class and the model would learn to recognize the text. Different approaches can yield different word embedding. We assume that it might not be able to strictly compare different approaches ("intelligence") in all cases but they might depend on circumstances.

6 Discussion

Learned embedding has some interesting characteristics regarding the process of obtaining the knowledge, moreover, potential knowledge rooted in the word embedding could be learned semi-automatically.

6.1 Unsupervised learning and the information origin

Word2vec method is often considered to be one of unsupervised learning methods. This assumption is based on the fact that the input is in the form of unlabeled data (corpus). However, the origin of information is in the language that were created by humans, therefore the word representation is learned from humans (authors of used corpus). This is inevitable when modeling languages that were created by humans. The similarity of two word representations is then simply based on word distribution.

We consider presented method of learning relations between classes also as unsupervised, moreover, independent of humans in terms of provided information, even though the classes were assigned by humans and data were labeled. The class name is just some identifier grouping multiple objects, so no information about the relation between these classes is provided.

Building on the fact that pictures simply reflect the reality, artificial intelligence is the consequence of the CNN architecture itself. Admitting that the artificial intelligence can be the consequence of the learning, it can be suitable to deduce the knowledge from for example classification task by the model itself instead of providing the knowledge to the model to learn it. In other words, instead of forcing a model to learn to describe picture by sentences in natural language (annotations made by humans), we want it to “understand” that it is beneficial to involve it (not in natural language) automatically as it could improve some task and later discover how to acquire this information. An example annotation “cat sits on the mat” could be learned as a fact that the model is somehow able to define “cat sits” if the body position affects the task. This could satisfy the *learning to learn* condition as presented in [7].

6.2 Learning more complex word embedding

Since the pictures in the dataset always show only one object (class), learned embedding mainly reflects object’s visual similarity. However, embedding could contain even more information regarding the relation between objects. Learning from the real word pictures with multiple objects on single caption could be the way. Therefore, with the ability to distinguish objects and to observe, more knowledge can be potentially produced.

We supplied annotated pictures to learn to distinguish basic classes. It would be interesting to have an automatic decision system that could be based on for example clustering. Then it would be possible to decide whether a newly observed object belongs to one of the known classes or is unknown, therefore, new class is to be created. Satisfying this condition, learning would be independent of humans. Probably the most difficult task *how to extract encoded knowledge from the word embedding* was not discussed.

7 Conclusion

Even though more sophisticated methods of learning word embedding representing the classification classes could be brought, we proposed a simple method based on errors just to demonstrate the approach of representing the knowledge by the model itself. Moreover, we built strictly on information from pictures because they reflect the real world and we think that learning from the real world can be easier task than learning any language since the ability to speak in any language is only human-specific while any animal needs to classify objects.

We demonstrated that some kind of intelligence can be derived automatically from the ability to distinguish two objects or their classes. Moreover, we assume that the intelligence in some form is necessary in order to obtain acceptable results. Further experiments are needed to decide whether this approach could be sufficient as for example knowledge representation for general artificial intelligence.

Acknowledgment

This work was supported by the BUT project FIT-S-17-4014 and the IT4IXS: IT4Innovations Excellence in Science project (LQ1602).

References

1. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
2. Kottur, S., Vedantam, R., Moura, J.M.F., Parikh, D.: Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. CoRR **abs/1511.07067** (2015)
3. Xu, R., Lu, J., Xiong, C., Yang, Z., Corso, J.J.: Improving word representations via global visual context. In: NIPS Workshop on Learning Semantics. (2014)
4. Lazaridou, A., Baroni, M., et al.: Combining language and vision with a multimodal skip-gram model. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2015) 153–163
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 1097–1105
6. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. (2009)
7. Baroni, M., Joulin, A., Jabri, A., Kruszewski, G., Lazaridou, A., Simonic, K., Mikolov, T.: Commai: Evaluating the first steps towards a useful general AI. CoRR **abs/1701.08954** (2017)