From Word2Vec to Chord2Vec

Sephora Madjiheurem

July 1, 2016

1 Skip-gram Model: Recap

The skip-gram allows to efficiently learn high-quality distributed vector representations that capture precise syntactic and semantic word relationships [1]. We give here a short reminder of how the skip-gram model works.

We define a text as a sequence of words drawn from a finite vocabulary of size W. A word can be described as a "one-hot" vector $w_t \in \{0,1\}^W$, where exactly one entry is non-zero and the subscript t represent the position of the word in the text. Given a word w_t in a text, define the context of word w_t by $C(w_t) = \{w_{t+j}, -m \leq j \leq m, j \neq 0\}$, where m is the size of the context. We consider the conditional probability of a context given a word $p(w_{t+j}|w_t)$. The goal is to find word representations that are useful for predicting the surrounding words in a sentence. Formally, given a corpus of words of size T and the context of word w_t given by $C(w_t)$, the objective of the skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{w\in C(w_t)} \log p(w|w_t). \tag{1}$$

The parametrization of the skip-gram model uses the architecture depicted in Figure 1 (from Mikolov et al. [1]). In this model, each output is computed using softmax to obtain the posterior distribution of context words:

$$p(w_{t+j}|w_t) = \frac{\exp(v_{w_{t+j}}^T v_{w_t})}{\sum_{w=1}^{W}\exp(v_w^T v_{w_t})}, \tag{2}$$

where $-m \leq j \leq m, j \neq 0$, m is the size of the training context, v_w is the vector representation for w, and W is the number of words in the vocabulary.

Detailed derivations and explanations of the parameter learning for this original skip-gram model can be found in [3].

Because the computation cost of objective (1) is proportional to W which can be very large when working with text training data, Mikolov et al. [1] use instead an efficient approximation, known as negative sampling (see [2] for details).

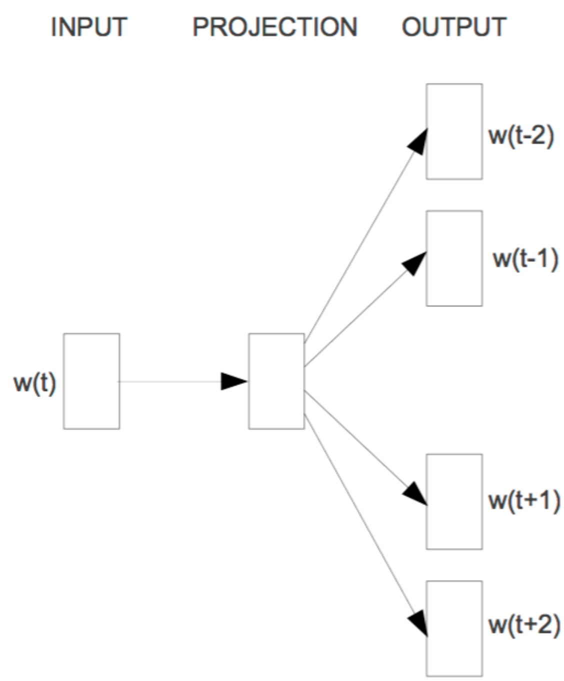

Figure 1: Skip-gram (Mikolov et al. [1])

2 Chord2Vec

Similarly to a text, we define a piece of music as a series of chords. A chord is a subset of notes drawn from a finite set of size N and can be represented by a binary vector $c \in \{0,1\}^N$.

To adapt the skip-gram model to music data there are a few points that need to be considered:

1. A text can be represented as a sequence of words, where each word can be represented as a "one-hot" vector. In the case of music, we need a "many-hot" vector to represent a chord, as more than one note can be heard simultaneously.

2. The set of notes is smaller than the vocabulary considered when working with text data.

The first point implies that the softmax layer in the original skip-gram model is no longer appropriate, as we need to allow more than one note to be active in a chord. A naive adaptation is to use a separate sigmoid function to predict each note in a chord. This makes the (very bad) independence assumption between the notes in a context chord, but we'll start from here and see how the model can be improve later. Under this model, the posterior distribution of a context chord c_{t+j} given a chord c_t is given by:

$$p(c_{t+j}|c_t) = \prod_{n=1}^{N} \frac{1}{1+\exp(-v_{c_{t+j}}^T v_{c_t})} \tag{3}$$

where $-m \leq j \leq m, j \neq 0$, m is the size of the training context, N is the number of different notes and v_c is the vector representation for chord c.

The second point, together with the fact that the computation of the objective is no longer dependent of number of possible context chords (since we are not using softmax, the sum over all possible context chords disappears) suggest that there might not be the need to use efficiency optimizations tricks as negative sampling.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[3] Xin Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014. URL `http://arxiv.org/abs/1411.2738`.