Pull request for Multimodal Deep Learning

State-of-the-art and current trends in NLP (Topic 01)

Department of Statistics
Ludwig-Maximilians-Universität München

Cem Akkus

Munich, June 3th, 2022

Submitted in partial fulfillment of the requirements for the degree of M. Sc.
Supervised by Dr. Matthias Aßenmacher

rough structure planned:

- brief overview for state of NLP before 2013
- Word Embeddings (incl. description, illustrations showing words in coordinate system)
- Encoder-Decoder (incl. description, equations, illustration of structure)
- Attention (incl. description, picture illustrating soft-alignment, table showing comparisons to translations of basic encdec-structure)
- Transformer (incl. description, types of attention, model architecture)
- Transformer architectures: BERT, T5, GPT3 (illustration of number of parameters, showing capabilities with examples)
- ...
- Compare time lines of NLP / CV (regarding transformer architectures etc.)

notes for first pull request:
Natural Language Processing (NLP) has existed for about 50 years, but it is more relevant than ever. There have been several breakthroughs in this branch of machine learning that is concerned with spoken and written language. For example, learning internal representations of words was one of the greater advances of the last decade. Word embeddings (Mikolov et al., 2013, Bojanowski et al., 2016) made it possible and allowed developers to encode words as dense vectors that capture their underlying semantic content. In this way, similar words are embedded close to each other in a lower-dimensional feature space. Another important challenge was solved by Encoder-decoder (also called sequence-to-sequence) architectures (Sutskever et al., 2014), which made it possible to map input sequences to output sequences of different lengths. They are especially useful for complex tasks like machine translation, video captioning or question answering. This approach makes minimal assumptions on the sequence structure and can deal with different word orders and active, as well as passive voice.

A definitely significant state-of-the-art technique is Attention (Bahdanau et al., 2014), which enables models to actively shift their focus – just like humans do. It allows following one thought at a time while suppressing information irrelevant to the task. As a consequence, it has been shown to significantly improve performance for tasks like machine translation. By giving the decoder access to directly look at the source, the bottleneck is avoided and at the same time, it provides a shortcut to faraway states and thus helps with the vanishing gradient problem. One of the most recent sequence data modeling techniques is Transformers (Vaswani et al., 2017), which are solely based on attention and do not have to process the input data sequentially (like RNNs). Therefore, the deep learning model is better in remembering context-induced earlier in long sequences. It is the dominant paradigm in NLP currently and even makes better use of GPUs, because it can perform parallel operations. Transformer architectures like BERT (Devlin

et al., 2019), T5 (Raffel et al., 2019) or GPT-3 (Brown et al., 2020) are pre-trained on a large corpus and can be fine-tuned for specific language tasks. They have the capability to generate stories, poems, code and much more. With the help of the aforementioned breakthroughs, deep networks have been successful in retrieving information and finding representations of semantics in the modality text. In the next paragraphs, developments for another modality image are going to be presented.

notes from literature:
Mikolov et al. (2013)
p.1
-computing continuous vector representations of words (from very large data sets and measured in word similarity task)
→ large improvements in accuracy at much lower computational cost
-not treating words as atomic units (no notion of similar- ity between words): simplicity, robustness → at limit for many tasks: eg automatic speech recognition (size of high quality transcribed speech data - often just millions of words). Scaling up basic techniques doesn't result in significant process. Therefore, more advanced techniques needed.
p.2
-modest dimensionality of the word vectors between 50 - 100 -expectation that not only will similar words tend to be close to each other, but that words can have multiple degrees of similarity (vector("King") - vector("Man") + vec- tor("Woman") results in a vector closest to Queen)
p.5
-many different types of similarities between words, for example, word big is similar to bigger in the same sense that small is similar to smaller -To find a word that is similar to small in the same sense as biggest is similar to big, we can simply compute vector X = vector("biggest") vector("big") + vector("small"). Then, we search in the vector space for the word closest to X measured by cosine distance, and use it as the answer to the question
p.10
-quality of vector representations of words on syntactic and semantic language tasks - possible to train high quality word vectors using very simple model architectures -Because of the much lower computational complexity, it is possible to compute very accurate high dimensional word vectors from a much larger data set (even on corpora with one trillion words).
Bojanowski et al. (2016)
p.1
Popular models ignore the morphology of words, by assigning a dis- tinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words → skipgram model:each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations - fast, allow- ing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data
-Most of these techniques represent each word of the vocabulary by a distinct vector,

without param- eter sharing. → ignore the internal structure of words, which is an important limitation for morphologically rich languages -while the Finnish language has fifteen cases for nouns. These languages contain many word forms that occur rarely (or not at all) in the training corpus, making it difficult to learn good word rep- resentations. Because many word formations follow rules, it is possible to improve vector representations for morphologically rich languages by using charac- ter level information. -this work is extension of the continuous skip- gram model (Mikolov et al., 2013b), which takes into account subword information
p.2
We model morphology by considering subword units, and representing words by a sum of its charac- ter n-grams.
p.3
The problem of predicting context words can in- stead be framed as a set of independent binary clas- sification tasks. Then the goal is to independently predict the presence (or absence) of context words.
By using a distinct vector representation for each word, the skipgram model ignores the internal structure of words. Each word w is represented as a bag of character n-gram. We add special boundary symbols ¡ and ¿ at the beginning and end of words, allowing to distinguish prefixes and suffixes from other character sequences. We also include the word w itself in the set of its n-grams, to learn a representation for each word (in addition to character n-grams). Taking the word where and n = 3 as an example, it will be represented by the character n-grams: ¡wh, whe, her, ere, re¿ and the special sequence ¡where¿.
We represent a word by the sum of the vector representations of its n-grams. We thus obtain the scoring function: This simple model allows sharing the representations across words, thus allowing to learn reliable representation for rare words.
p.10
simple method to learn word representations by taking into account subword information
Sutskever et al. (2014)

p.1
-DNNS powerful models that achieved excellent performances on difficult learning tasks: work great w/ large labeled training sets -cannot be used to map seq2seq -enc dec pre- sented: end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure -model learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the pas- sive voice -reversing the order of the words in all source sentences (but not target sentences) im- proved the LSTM's performance markedly , because of short term dependencies between source and target which made optimisation problem easier
DNNs: -extremely powerful machine learning models that achieve ex- cellent performance on difficult problems such as speech recognition and visual object recog- nition , because they can perform arbitrary parallel computation for a modest number of steps -So, while neural networks are related to conventional statistical models, they learn an intricate computation if there exists a parameter setting of a large DNN that achieves good results -DNNS can only be applied to problems whose inputs and targets can be sensibly encoded

with vectors of fixed dimensionality -significant limitation, since many important problems are best expressed with sequences whose lengths are not known a-priori. (e.g. sequential problems: speech recognition and machine translation (, question answering))
p.2
→ domain-independent method that learns to map sequences to sequences would be useful -straightforward application of the Long Short-Term Memory (LSTM) architecture can solve general sequence to sequence problems -The idea is to use one LSTM to read the input sequence, one timestep at a time, to obtain large fixed- dimensional vector representation, and then to use another LSTM (essentially a rnn lm) to extract the output sequence from that vector
-insert figure of encdec structure
p.4
-Interesting: discovered that the LSTM learns much better when the source sentences are reversed (the target sentences are not reversed)
p.6 interesting figure, maybe include something similar
Bahdanau et al. (2014)

p.1
-Unlike traditional statistical machine translation, neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. -most recently proposed belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. -Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition
-traditional phrase-based translation system: many small sub-components that are tuned separately -neural machine translation: single, large nn that reads a sentence and outputs correct translation -most of nmt models - encoder-decoder: encoder nn reads and encodes source-sentence into fixed-length vector, decoder then outputs from encoded vector (it's jointly trained)
-potential issue with this encoder–decoder approach: neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector (could cause difficulty coping w/ long sentences, especially when greater then length of input sentences) extension to inc-dec model which learns to align and translate jointly: Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.
p.2
-most important distinguishing feature of this approach from the basic encoder–decoder is that it does not attempt to encode a whole input sentence into a single fixed-length vector -instead: encodes input sequence into sequence of vectors and chooses a subset of

these vectors adaptively while decoding the translation. This frees a neural translation model from having to squash all the information of a source sentence, regardless of its length, into a fixed-length vector. We show this allows a model to cope better with long sentences.
-From a probabilistic perspective, translation is equivalent to finding a target sentence y that max- imizes the conditional probability of y given a source sentence x, i.e., arg maxy p(y — x). In neural machine translation, we fit a parameterized model to maximize the conditional probability of sentence pairs using a parallel training corpus. Once the conditional distribution is learned by a translation model, given a source sentence a corresponding translation can be generated by searching for the sentence that maximizes the conditional probability.
-look at page 2-3 for equations describing encoder-decoder and insert pictures for encoder-decoder
p.7
-provides an intuitive way to inspect the (soft-)alignment between the words in a generated translation and those in a source sentence -insert figure 3 of page 6 -alignments largely monotonic. We see strong weights along the diagonal of each matrix. However, we also observe a number of non-trivial, non-monotonic alignments
-An additional benefit of the soft align- ment is that it naturally deals with source and target phrases of different lengths, without requiring a counter-intuitive way of mapping some words to or from nowhere ([NULL])
-maybe add example of sentence RNNsearch-50 produced compared to RNNencdec-50

References

Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2016). Enriching word vectors with subword information.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space.

Sutskever, I., Vinyals, O. and Le, Q. V. (2014). Sequence to sequence learning with neural networks.