REPRESENTATION LEARNING: A BRIEF OVERVIEW

(Bill) Yuchen Lin

University of Southern California
yuchen.lin@usc.edu

September 2, 2019

ABSTRACT

Learning good representations of data is a key factor of the success of deep learning. This lecture notes briefly reviews the basic concepts and methods in representation learning and connections with other related topics, such as network pre-training, multi-task learning, transfer learning, few-shot learning, etc. We also talks about some recent advances about disentangled representation learning and representation learning on graphs, which are not covered by the textbook.

***K*eywords** Representation Learning · Natural Language Processing · Deep Learning

1 Background: What is representation learning?

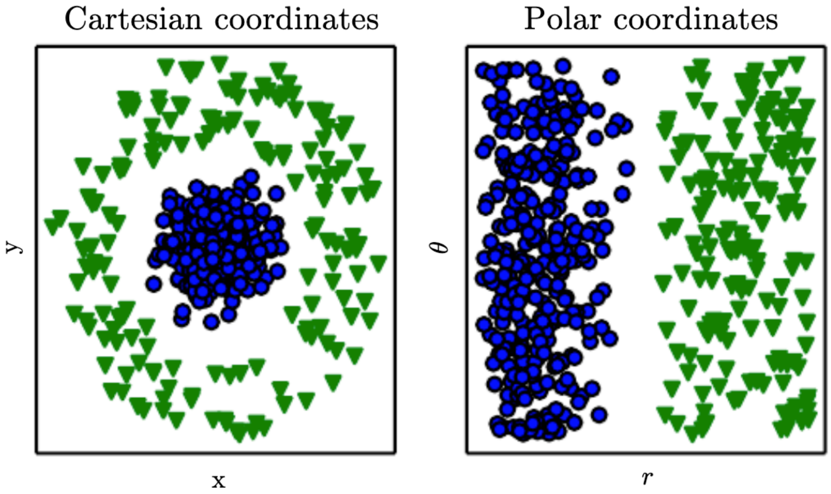

Figure 1: Classes that were not linearly separable in the input space may become linearly separable through representation learning (e.g. the transformation from Cartesian to Polar coordinates here in the example).

A good and suitable representation of data can be essential for computational tasks, such as machine learning. For example, "$210 \div 6 \ =?$" is a very easy task for a human to solve, but the same problem with *Roman numeral representations* ("$CCX \div VI =?$") can be less obvious. The performance of machine learning methods is also heavily dependent on the choice of data representation. Figure 1 shows a good representation of data can make learning (with linear classifier) much easier.

Feature engineering is a way to take advantage of *human ingenuity* and *prior knowledge* for transforming raw data input $\mathcal{X}$ (a sequence of tokens, a tensor of pixels, a time series of sound waves, etc.) to a representation $\mathcal{X}'$, which is

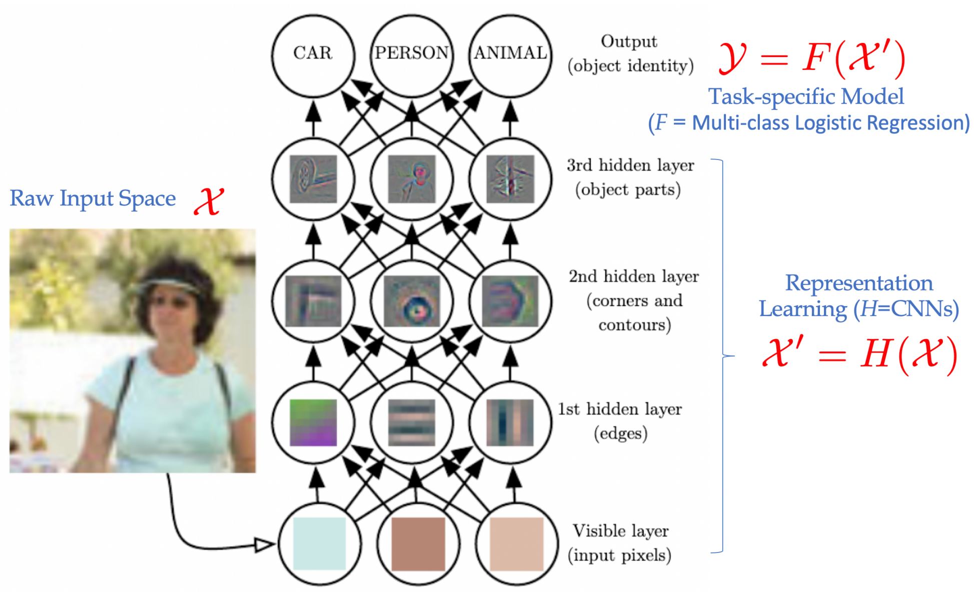

Figure 2: Using CNNs as the representation learning method for the object classification task.

usually a very labor-intensive process. Traditional machine learning methods usually focus on mapping such pre-defined data representations $\mathcal{X}'$ to task-specific output space ($\mathcal{Y}$), which is generated by feature engineering.

"**Representation learning** is to learn representations of the data that make it easier to extract useful information when building classifiers or other predictors.", defined by Bengio et al.(2012)[1]. The key motivation here is to let models not only discover the mapping between representations/features and outputs $\mathcal{Y} = F(\mathcal{X}')$, **but also representations itself** ($\mathcal{X}' = H(\mathcal{X})$). Modern deep learning models usually can be seen as a neural network-based representation learning module (e.g. MLP/CNNs/RNNs) followed by a simple predictor model (e.g. logistic regression):

$$\mathcal{Y} = F(\mathcal{X}') = F(H(\mathcal{X})).$$

Among the various ways of learning representations, this lecture focuses on deep learning methods: those that are formed by **the composition of multiple non-linear transformations**, with the goal of yielding **more abstract – and ultimately more useful** - representations. Figure 2 shows an example of using CNNs as the representation learning method for the object classification task.

2 Priors for Representation Learning: What makes a representation good?

"One reason why *explicitly dealing with representations* is interesting is because **they can be convenient to express many general priors about the world around us**, i.e., priors that are not task-specific but would be likely to be useful for a learning machine to solve AI-tasks." (Bengio and LeCun, 2007) In this section, we discuss some of such general factors are necessary for learning a good general representation.

1. **Smoothness**: assuming the function to be learned H is s.t. $x_1 \approx x_2 \rightarrow H(x_1) \approx H(x_2)$
2. **Multiple explanatory factors**: the data generating distribution is generated by different underlying factors, and for the most part what one learns about one factor generalizes in many configurations of the other factors.
3. **Hierarchical organizations**: the concepts that are useful for describing the world around us can be defined in terms of other concepts, in a hierarchy, with more abstract concepts higher in the hierarchy, defined in terms of less abstract ones.
4. **Semi-supervised learning**: representations that are useful for modeling $P(X)$ tend to be useful when learning $P(Y|X)$, allowing sharing of statistical strength between the unsupervised and supervised learning tasks.
5. **Transferable/shareable across tasks**: with many Ys of interest or many learning tasks in general, tasks are explained by factors that are shared with other tasks, allowing sharing of statistical strengths across tasks.
6. **Natural clustering**: different values of categorical variables such as object classes are associated with separate manifolds; i.e., $P(X|Y = i)$ for different i tend to be well separated and not overlap much.

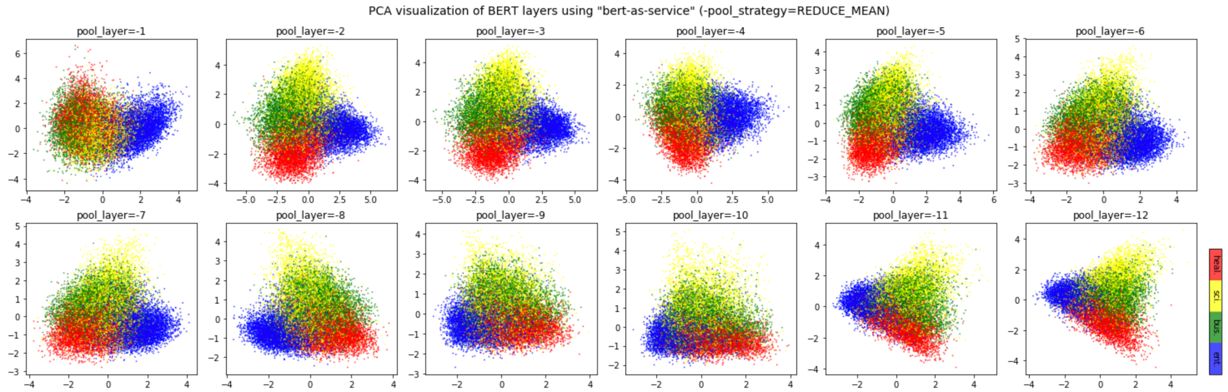

Figure 3: Visualizing via PCA on BERT representations for 20k news titles of four classes.

7. **Sparsity**: Most features should presumably not be relevant to describing most inputs—there is no need to use a feature that detects elephant trunks when representing an image of a cat. It is therefore reasonable to impose a prior that any feature that can be interpreted as "present" or "absent" should be absent most of the time.

We can view many of the above priors as ways to help the learner discover and disentangle some of the underlying (and a priori unknown) factors of variation that the data may reveal.

3 Unsupervised Pre-training in Natural Language Processing

Most of recent deep learning approaches do not learn from the scratch, but instead follow a "pre-training and then fine-tuning" fashion. Unlike images that naturally have meaningful numeric representations, the raw format of the language data is highly symbolic. Thus, representing textual units (e.g. character, word, sentence, paragraph, document) is the most fundamental research in NLP, which connects symbolic spaces to neural network computations.

For images, there is no clear way to segment pixels such that each unit is a meaningful part and composition of them can reproduce the semantics (i.e. a set of pixels has no stable semantics across instances). Thus, the common way of learning image representations has to be dependent on the human annotations (e.g. ImageNet dataset). Models learned on such human labels (e.g. Mask R-CNN) can be directly useful for pre-training the models for other tasks. Recently, He et al. (2018) [2] even found that such "supervised pre-training and fine-tuning" schema is not useful in many cases.

In contrast, language data is naturally segmented as sequences of semantic units with stable meaning. Also, with abundant existing language corpora, unsupervised pre-training is much more promising.

Unsupervised pre-training combines two different ideas:

- **Initial parameters** for a deep NN can have a significant **regularizing effect** on the model.

  Initializing the model in a location that would cause it to approach one local minimum rather than another.

- Learning about the **input distribution** can help with learning about the mapping from inputs to outputs.

How can we learn representations for textual units without supervision?

1. **Meaningful unsupervised tasks:** Mask some parts of the data as inputs, and thus the task is to recover the complete structure. For example, in Word2Vec, we use the neighboring words to predict central words or vice versa. In BERT, we also train a Transformer model to predict whether two sentences are continuous.

2. **Comprehensive data:** Using the whole Wikipedia and many other additional corpora, the pre-trained representations are shown to be powerful in many domain-general tasks.

Such unsupervisedly learned word/sentence representations shows good generalization in many down-stream tasks and further benefits transfer learning, multi-task learning, and few-shot learning research in NLP.

4 Learning transferable representations for transfer/multi-task learning

Transfer learning, in general, focuses on how we can make use of learned knowledge in one setting and apply for another unseen setting. In fact, fine-tuning pre-trained representation for new tasks is already a perfect example of transfer learning. When we are training the word embeddings or the BERT model, we actually learning the representations for the unsupervised tasks. The target tasks (e.g. sentiment classification) can be very different from the source tasks (e.g. predicting neighboring words). Thus, most modern models for various downstream new tasks can be seen as a special form of transfer learning.

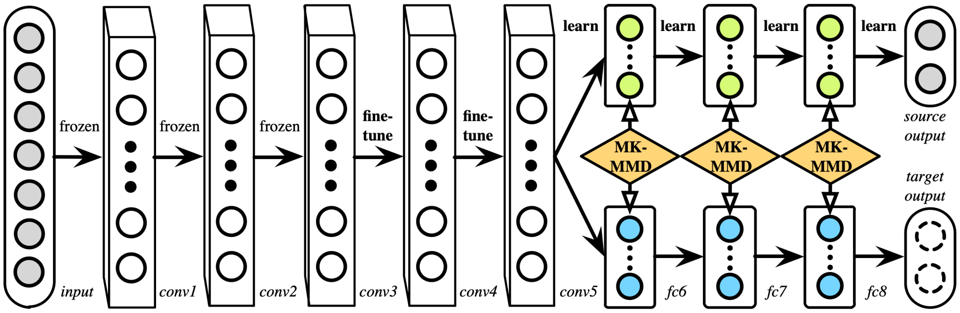

Figure 4: The DAN architecture [3] for learning transferable representations.

Learning with such differences between the distribution of input and output spaces in two or more tasks covers a lot of difficult but practical scenarios:

- **Domain adaptation:** usually, the input distribution is different from source to target domain, but the task and the output space remains the same. For example, sentiment classification (Negative/Neutral/Positive) in the Movie domain and the Computer domain. [1]

  In some harder cases, the output spaces can also change (e.g. NER in news articles and in biomedical papers). A special case in NLP is **cross-lingual transfer**.
- **Multi-task learning:** we assume the input distributions is the same or we choose to ignore the differences between distributions, but the target space is task-specific (e.g. classification, sequence tagging, seq2seq, etc.)
- **Few-shot learning:** we assume in a new task/domain, there are only a very small number of examples for learning to transfer. **Meta-learning**, which focuses on rapidly learning representations for new tasks, is usually evaluated in this problem setting.
- **Zero-shot learning:** a more extreme case where we only have description of the target domain/task but no labels at all. The research in this direction should focus on the design of incorporating **task representations**, such that new tasks can be adapted and inferred without labeled data.

All such challenging learning scenarios are only possible in deep learning models with a very general, easy-to-adapt representation learning method. The key idea used in deep transfer learning is to disentangle task-specific representations and then maximally share task-general representation. Figure 4 shows the idea of deep adaptation networks by Long et al. (2015) [3], where the first part of layers are highly transferable (conv1-3), and then slightly less transferable (conv4-5), and finally (fc6-8) not transferable at all.

5 Semi-Supervised Disentangling of Causal Factors

This section we study when and how representation learning can help in semi-supervised settings. A common and popular assumption for good representations is that “an ideal representation is one in which the features within the representation correspond to the underlying causes of the observed data”. A causal factor is like an explanatory, high-level feature. This is to say, the representations should separate features caused by different factors in the feature space. The hypothesis encourages the semi-supervised learning by:

1. Learning a good representation of $p(x)$.

[1] One can imagine that there is an underlying function that tells whether any statement is positive, neutral, or negative, but of course the vocabulary and style may vary from one domain to another, making it more difficult to generalize across domains.

2. Assuming it is a disentangled representation that separates all casual factors of generating the data x.
3. Assuming y is among the most salient causal factors.
4. Learning $p(y|x)$.

"If y is closely associated with one of the causal factors of x , then $p(x)$ and $p(y|x)$ will be strongly tied, and unsupervised representation learning that tries to disentangle the underlying factors of variation is likely to be useful as a semi-supervised learning strategy." For example, in a person image x, a causal factor can be whether the person is bearded, y is to say if the person is a man or woman.

If we assume "y is one of the causal factors of x", let h represent all those factors, and then the true generative process can be conceived as structured according to this directed graphical model, with h as the parent of x:

$$p(h, x) = p(x|h)p(h); \quad p(x) = \mathbb{E}_h p(x|h)$$

.

If y is closely related to h, then $p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$ is much easier to learn with partially labeled data.

An important research problem regards the fact that most observations are formed by an extremely large number of underlying causes. Suppose $y = h_i$, but the unsupervised learner does not know which h_i. Thus, we need to define the casual factors by assuming any structured pattern that the feed-forward network can recognize is highly salient. Generative adversarial networks (GANs) utilize a game-based learning schema to learn how to determine what factors are salient and thus more effectively find casual factors related to y.

6 Distributed Representation

Good representations are expressive, meaning that a reasonably-sized learned representation can capture a huge number of possible input configurations. **How many parameters does it require compared to the number of input regions (or configurations) it can distinguish?**

Learners of one-hot representations, such as traditional clustering algorithms, Gaussian mixtures, k-NN algorithms, decision trees, or Gaussian SVMs all require $O(N)$ parameters (and/or $O(N)$ examples) to distinguish $O(N)$ input regions. One could naively believe that one cannot do better.

However, RBMs, sparse coding, auto-encoders or multi-layer neural networks can all represent up to $O(2^k)$ input regions using only $O(N)$ parameters (with k the number of non-zero elements in a sparse representation, and $k = N$ in non-sparse RBMs and other dense representations).

Distributed representations are powerful because they can use n features with k values to describe k^n different concepts. Many deep learning algorithms are motivated by the assumption **that the hidden units can learn to represent the underlying causal factors that explain the data**. Distributed representations are natural for this approach, because **each direction in representation space can correspond to the value of a different underlying configuration variable**.

When and why can there be a statistical advantage from using a distributed representation as part of a learning algorithm? Distributed representations can have a statistical advantage when an apparently complicated structure can be compactly represented using a small number of parameters.

From Figure 5-7, we can see how distributed representations can be more example/parameter-efficient, and can represent more relational information in/out of domains.

References

[1] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2012.

[2] Kaiming He, Ross B. Girshick, and Piotr Dollár. Rethinking imagenet pre-training. *ArXiv*, abs/1811.08883, 2018.

[3] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

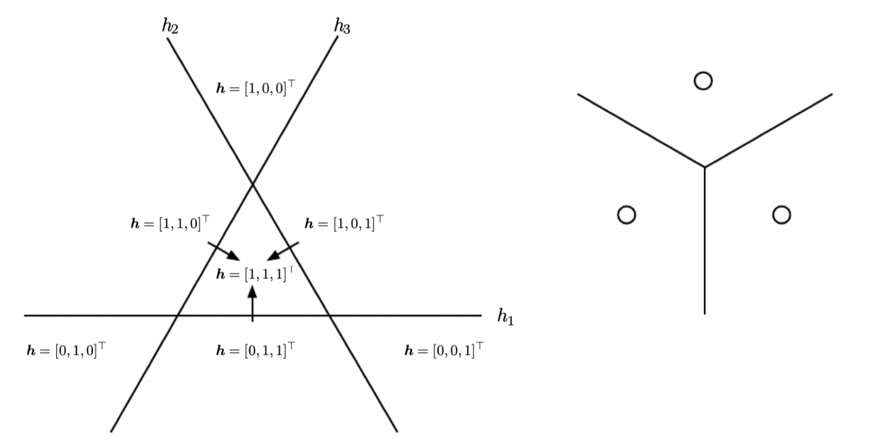
h_2
h_3
h_1
$\boldsymbol{h} = [1,0,0]^\top$
$\boldsymbol{h} = [1,1,0]^\top$
$\boldsymbol{h} = [1,0,1]^\top$
$\boldsymbol{h} = [1,1,1]^\top$
$\boldsymbol{h} = [0,1,0]^\top$
$\boldsymbol{h} = [0,1,1]^\top$
$\boldsymbol{h} = [0,0,1]^\top$

Figure 5: Left: Illustration of how a learning algorithm based on a distributed representation breaks up the input space into regions.; Right: Illustration of how the nearest neighbor algorithm breaks up the input space into different regions.

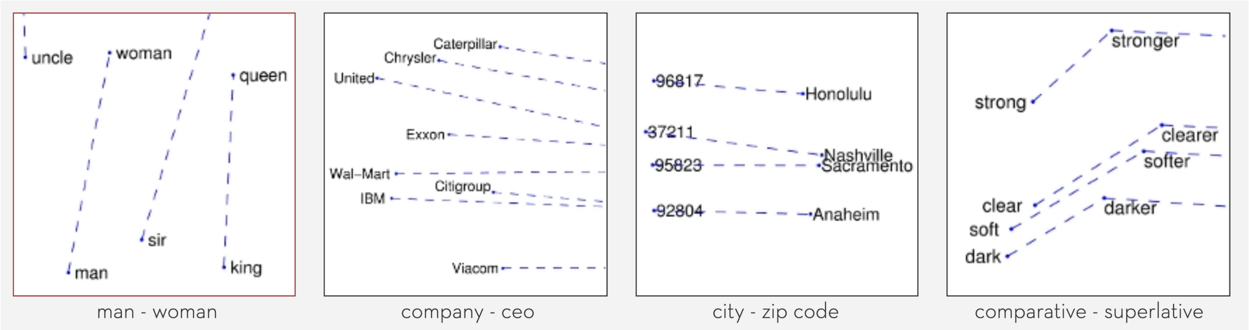

Figure 6: The distributed representations underlying concept that distinguishes may be equivalently specified by various other word pairs.

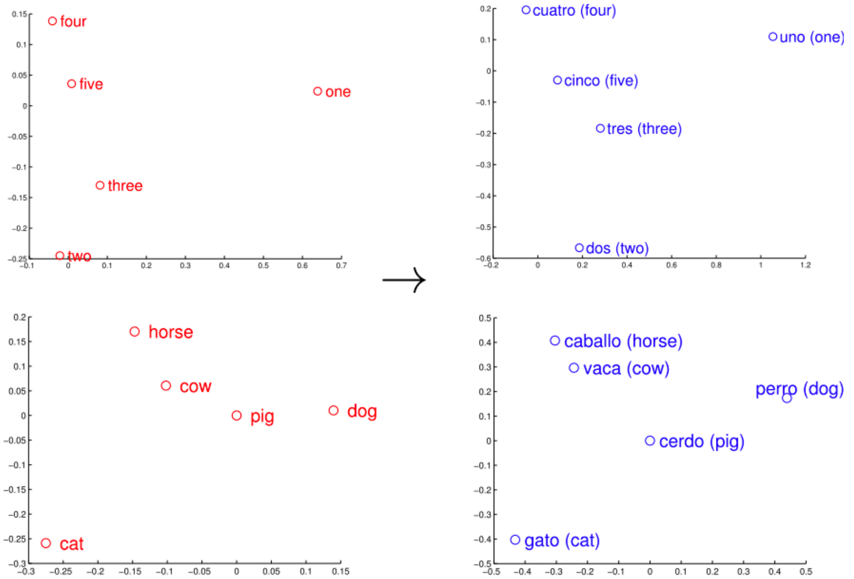

Figure 7: Distributed representations can hold more stable concept relational information.