STATS 306B: Unsupervised Learning **Spring 2014**

Lecture 15 — May 19

Lecturer: Richard Socher *Scribe: Natalie Telis, Ludi Rehak*

Note: These notes serve to supplement the accompanying lecture slides.

15.1 Deep learning

Representation learning is another name for the task of finding a useful (often reduced) feature representation for each datapoint in a dataset. In this lecture, we will explore **deep learning** an approach to representation learning based on hierarchical latent variable modeling that has recently enjoyed great practical success.

15.1.1 A deep architecture: deep belief networks

Markov Random Fields with multiple layers and various types of multiple-layer neural networks make use of multiple stacked hidden variable layers to model the relationship between inputs and outputs.

15.1.2 Resurgence of deep learning

Many of these techniques have been around since the eighties. Why is this particularly relevant today?

1. Handcrafting features is very hard. The features are often both over-specified and incomplete.
2. The feature representation work must be redone for different domains and different tasks, so we cannot automate it over different fields.
3. Humans develop these representations for learning and reasoning. Why cannot we train machines to do the same automatically?

An example from NLP: Parsing parts-of-speech

Many existing machine learning and natural language processing (NLP) systems are fragile because their standard feature representations are incredibly sparse. We are presented with the example of a system that can parse the sentence, "My dog also eats oranges," but has trouble parsing "My dog also eats bananas," since the sentence, "She went bananas," also appeared in the corpus. The inability to disambiguate based on context can be traced back to the sparse feature representation.

The curse of dimensionality

Learned distributed representations allow us to cope with the curse of ambient dimensionality.

Unsupervised learning vs. supervised learning

Many of the most successful NLP and ML carry out supervised learning with labeled training data. However, we know that obtaining labeled data is costly, so there is a strong incentive to leverage widely available unlabeled data and to develop models that integrate the two sources in a semisupervised fashion.

Multiple layers of representations

Most complex data sets have many layers of hidden representations, which we can observe. See the example of face recognition in the slides.

In summary, why now?

Many of these techniques are old but did not work well in the past. Before 2006, training deep architectures was basically unsuccessful due in part to overfitting. What has changed since then?

1. We have much faster machines, and many of the techniques to be described can be easily parallelized.
2. New, more successful methods for unsupervised pre-training have been developed.
3. More efficient parameter estimation methods are available.
4. We have better understanding of model regularization.
5. The learned representations have found great success in modern speech, vision, and language tasks.

15.2 Neural networks: feed-forward and auto encoders

15.2.1 Introduction

An interesting method is for learning the weights of a neural network is Adagrad. Adagrad proceeds as in (stochastic) gradient descent but separately rescales each component of the (stochastic) gradient: for example, for a parameter vector θ, the update for each entry θ_i takes the form

$$\theta_i^{t+1} = \theta_i^t - \frac{\alpha}{\sqrt{\sum_{s=0}^t g_i^{s2}}} g_i^t$$

where g^t is the (stochastic) gradient vector at iteration t, and α is a step size. The coordinate-specific scaling is especially valuable for settings in which some features are much rarer than others, as rare feature coordinates will have a larger effective step size.

15.2.2 Practical examples

Learning word representations with a simple unsupervised model

An example given in the lecture slides is neural word embeddings. These are around 50 to 100 dimensional vectors which capture syntactic similarities, e.g., context and part-of-speech tags. For example, president, chairperson, and senator are very similar.

More meaningful neural word embeddings can be obtained by introducing a form of supervision into our unsupervised task. For instance, sentiment is often not captured by unsupervised word embeddings but can be captured through training. We can train a neural network on a huge body of text, for example Wikipedia. Let's assume any example of word usage is a positive training example - for example, "cat chills on a mat". Negative training examples are created by replacing the word observed (for example, "on") with a random word, such as an island name - "cat chills Jeju a mat". Essentially, we have transformed our unsupervised learning task into a supervised learning problem. For other examples of this unsupervised to supervised reduction, see ESL 14.2.4.

15.2.3 Auto-encoders

An auto-encoder is a multilayered neural net with the target output being the input. The reconstruction is essentially decoder(encoder(input)). We begin with the manifold learning hypothesis that examples will concentrate near a lower dimensional 'manifold', a region of high density where small changes are only allowed in certain directions. Auto-encoders learn salient variations, similar to a non-linear PCA. They focus on minimizing reconstruction error. This forces latent representation of similar inputs to stay on the manifold.

15.2.4 Stacking auto-encoders

Auto-encoders can be stacked successfully to model highly non-linear inputs, such as speech encoding. In essence, one auto-encoder layer L_i feeds into another auto-encoder layer L_{i+1}, very similarly to the neural net description above. By doing this multiple times, we can improve the overall model by taking as input salient features of the second layers and continuing to abstract further.