Deep Learning

Yuan An

Today, I read a review article " Deep Learning" written by LeCun *et al*. [5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.

Machine learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. It can also be used to identify objects in images, transcribe speech into text, match news items and so on.

But conventional machine learning techniques were limited in their ability to process natural data in their raw form. So deep learning is increasingly used in these applications. For decades, constructing a pattern recognition or machine learning system required careful engineering and considerable domain expertise to design a feature extractor.

Representation learning is a set of methods that allow a machine to be fed with raw data and automatically discover the representations needed for detection or classification. Deep learning method are representation learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level into a representation at a higher slightly more abstract level. With the composition of enough transformations, very complex functions can be learned.

The key aspect of deep learning is that these layers of features are not designed by human engineers: they learned from data using a general-purpose learning procedure.

Deep learning has made major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has produced extremely promising results for various tasks in natural language understanding [3], particularly topic classification sentiment analysis, question answering [1] and language translation.

Deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increase in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this process.

Then the authors talked about supervised learning and using backpropagation to train multilayer architectures.

The most common form of machine learning, deep or not, is supervised learning. In the typical deep learning system, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine. To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

In practice, most practitioners use a procedure called stochastic gradient descent (SGD). This simple procedure usually finds a good set of weights surprisingly quickly when compared with far more elaborate optimization techniques [2].

And problems such as image and speech recognition require the input-output function to be insensitive to irrelevant variations of the input, such as variations in position, orientation or illumination of an object, or variations in the pitch or accent of speech, while being very sensitive to particular minute variations (for example, the difference between a white wolf and a breed of wolf-like white dog called a Samoyed).

In Figure 1a, a multilayer neural network (shown by the connected dots) can distort the input space to make the classes of data (examples of which are on the red and blue lines) linearly separable. In Figure 1b, the chain rule of derivatives tells us how two small effects (that of a small change of x on y, and that of y on z) are composed. Figure 1c shows the equations used for computing the forward pass in a neural net with two hidden layers and one output layer. In Figure 1b, here's the equations used for computing the backward pass.

The key insight is that the derivative (or gradient) of the objective with respect to the input of a module can be com-

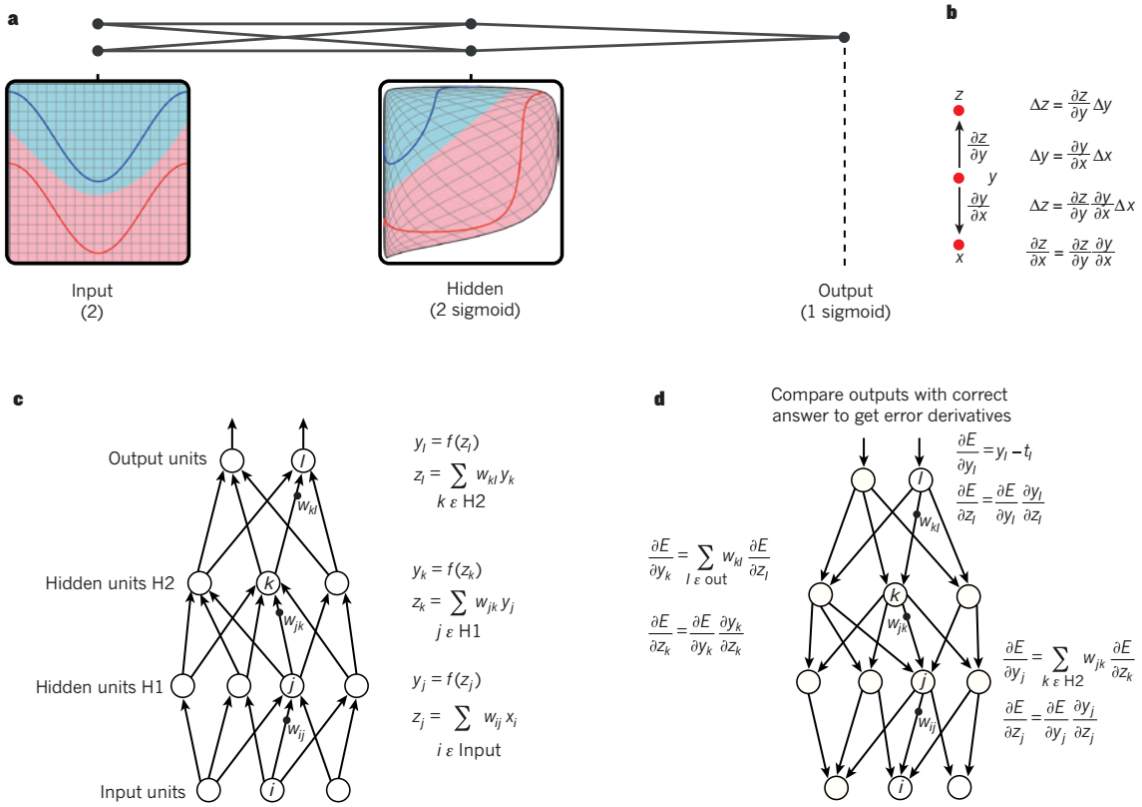

Figure 1. Multilayer neural networks and backpropagation

puted by working backwards from the gradient with respect to the output of that module (or the input of the subsequent module). The backpropagation equation can be applied repeatedly to propagate gradients through all modules, starting from the output at the top all the way to the bottom. Once these gradients have been computed, it is straightforward to compute the gradients with respect to the weights of each module.

There was, however, one particular type of deep, feedforward network that was much easier to train and generalized much better than networks with full connectivity between adjacent layer. This was the convolutional neural network (ConvNet) [4]. It achieved many practical successes during the period when neural networks were out of favor and it has recently been widely adopted by the computer vision community.

References

[1] A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014. 1

[2] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *NIPS*, pages 161–168, 2007. 1

[3] R. Collobert, J. Weston, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 2011. 1

[4] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Handwritten digit recognition with a back-propagation network. In *NIPS*, pages 396–404, 1990. 2

[5] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015. 1