2

Some Historical Context of Deep Learning

Until recently, most machine learning and signal processing techniques had exploited shallow-structured architectures. These architectures typically contain at most one or two layers of nonlinear feature transformations. Examples of the shallow architectures are Gaussian mixture models (GMMs), linear or nonlinear dynamical systems, conditional random fields (CRFs), maximum entropy (MaxEnt) models, support vector machines (SVMs), logistic regression, kernel regression, multi-layer perceptrons (MLPs) with a single hidden layer including extreme learning machines (ELMs). For instance, SVMs use a shallow linear pattern separation model with one or zero feature transformation layer when the kernel trick is used or otherwise. (Notable exceptions are the recent kernel methods that have been inspired by and integrated with deep learning; e.g. [9, 53, 102, 377]). Shallow architectures have been shown effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound and language, and natural image and visual scenes.

Human information processing mechanisms (e.g., vision and audition), however, suggest the need of deep architectures for extracting complex structure and building internal representation from rich sensory inputs. For example, human speech production and perception systems are both equipped with clearly layered hierarchical structures in transforming the information from the waveform level to the linguistic level [11, 12, 74, 75]. In a similar vein, the human visual system is also hierarchical in nature, mostly in the perception side but interestingly also in the "generation" side [43, 126, 287]). It is natural to believe that the state-of-the-art can be advanced in processing these types of natural signals if efficient and effective deep learning algorithms can be developed.

Historically, the concept of deep learning originated from artificial neural network research. (Hence, one may occasionally hear the discussion of "new-generation neural networks.") Feed-forward neural networks or MLPs with many hidden layers, which are often referred to as deep neural networks (DNNs), are good examples of the models with a deep architecture. Back-propagation (BP), popularized in 1980s, has been a well-known algorithm for learning the parameters of these networks. Unfortunately BP alone did not work well in practice then for learning networks with more than a small number of hidden layers (see a review and analysis in [20, 129]. The pervasive presence of local optima and other optimization challenges in the non-convex objective function of the deep networks are the main source of difficulties in the learning. BP is based on local gradient information, and starts usually at some random initial points. It often gets trapped in poor local optima when the batch-mode or even stochastic gradient descent BP algorithm is used. The severity increases significantly as the depth of the networks increases. This difficulty is partially responsible for steering away most of the machine learning and signal processing research from neural networks to shallow models that have convex loss functions (e.g., SVMs, CRFs, and MaxEnt models), for which the global optimum can be efficiently obtained at the cost of reduced modeling power, although there had been continuing work on neural networks with limited scale and impact (e.g., [42, 45, 87, 168, 212, 263, 304].

The optimization difficulty associated with the deep models was empirically alleviated when a reasonably efficient, unsupervised learning algorithm was introduced in the two seminar papers [163, 164]. In these papers, a class of deep generative models, called deep belief network (DBN), was introduced. A DBN is composed of a stack of restricted Boltzmann machines (RBMs). A core component of the DBN is a greedy, layer-by-layer learning algorithm which optimizes DBN weights at time complexity linear to the size and depth of the networks. Separately and with some surprise, initializing the weights of an MLP with a correspondingly configured DBN often produces much better results than that with the random weights. As such, MLPs with many hidden layers, or deep neural networks (DNN), which are learned with unsupervised DBN pre-training followed by back-propagation fine-tuning is sometimes also called DBNs in the literature [67, 260, 258]. More recently, researchers have been more careful in distinguishing DNNs from DBNs [68, 161], and when DBN is used to initialize the training of a DNN, the resulting network is sometimes called the DBN–DNN [161].

Independently of the RBM development, in 2006 two alternative, non-probabilistic, non-generative, unsupervised deep models were published. One is an autoencoder variant with greedy layer-wise training much like the DBN training [28]. Another is an energy-based model with unsupervised learning of sparse over-complete representations [297]. They both can be effectively used to pre-train a deep neural network, much like the DBN.

In addition to the supply of good initialization points, the DBN comes with other attractive properties. First, the learning algorithm makes effective use of unlabeled data. Second, it can be interpreted as a probabilistic generative model. Third, the over-fitting problem, which is often observed in the models with millions of parameters such as DBNs, and the under-fitting problem, which occurs often in deep networks, can be effectively alleviated by the generative pre-training step. An insightful analysis on what kinds of speech information DBNs can capture is provided in [259].

Using hidden layers with many neurons in a DNN significantly improves the modeling power of the DNN and creates many closely

optimal configurations. Even if parameter learning is trapped into a local optimum, the resulting DNN can still perform quite well since the chance of having a poor local optimum is lower than when a small number of neurons are used in the network. Using deep and wide neural networks, however, would cast great demand to the computational power during the training process and this is one of the reasons why it is not until recent years that researchers have started exploring both deep and wide neural networks in a serious manner.

Better learning algorithms and different nonlinearities also contributed to the success of DNNs. Stochastic gradient descend (SGD) algorithms are the most efficient algorithm when the training set is large and redundant as is the case for most applications [39]. Recently, SGD is shown to be effective for parallelizing over many machines with an asynchronous mode [69] or over multiple GPUs through pipelined BP [49]. Further, SGD can often allow the training to jump out of local optima due to the noisy gradients estimated from a single or a small batch of samples. Other learning algorithms such as Hessian free [195, 238] or Krylov subspace methods [378] have shown a similar ability.

For the highly non-convex optimization problem of DNN learning, it is obvious that better parameter initialization techniques will lead to better models since optimization starts from these initial models. What was not obvious, however, is how to efficiently and effectively initialize DNN parameters and how the use of large amounts of training data can alleviate the learning problem until more recently [28, 20, 100, 64, 68, 163, 164, 161, 323, 376, 414]. The DNN parameter initialization technique that attracted the most attention is the unsupervised pretraining technique proposed in [163, 164] discussed earlier.

The DBN pretraining procedure is not the only one that allows effective initialization of DNNs. An alternative unsupervised approach that performs equally well is to pretrain DNNs layer by layer by considering each pair of layers as a de-noising autoencoder regularized by setting a random subset of the input nodes to zero [20, 376]. Another alternative is to use *contractive* autoencoders for the same purpose by favoring representations that are more robust to the input variations, i.e., penalizing the gradient of the activities of the hidden units with respect to the inputs [303]. Further, Ranzato et al. [294] developed the

sparse encoding symmetric machine (SESM), which has a very similar architecture to RBMs as building blocks of a DBN. The SESM may also be used to effectively initialize the DNN training. In addition to unsupervised pretraining using greedy layer-wise procedures [28, 164, 295], the supervised pretraining, or sometimes called discriminative pretraining, has also been shown to be effective [28, 161, 324, 432] and in cases where labeled training data are abundant performs better than the unsupervised pretraining techniques. The idea of the discriminative pretraining is to start from a one-hidden-layer MLP trained with the BP algorithm. Every time when we want to add a new hidden layer we replace the output layer with a randomly initialized new hidden and output layer and train the whole new MLP (or DNN) using the BP algorithm. Different from the unsupervised pretraining techniques, the discriminative pretraining technique requires labels.

Researchers who apply deep learning to speech and vision analyzed what DNNs capture in speech and images. For example, [259] applied a dimensionality reduction method to visualize the relationship among the feature vectors learned by the DNN. They found that the DNN's hidden activity vectors preserve the similarity structure of the feature vectors at multiple scales, and that this is especially true for the filterbank features. A more elaborated visualization method, based on a top-down generative process in the reverse direction of the classification network, was recently developed by Zeiler and Fergus [436] for examining what features the deep convolutional networks capture from the image data. The power of the deep networks is shown to be their ability to extract appropriate features and do discrimination jointly [210].

As another way to concisely introduce the DNN, we can review the history of artificial neural networks using a "hype cycle," which is a graphic representation of the maturity, adoption and social application of specific technologies. The 2012 version of the hype cycles graph compiled by Gartner is shown in Figure 2.1. It intends to show how a technology or application will evolve over time (according to five phases: technology trigger, peak of inflated expectations, trough of disillusionment, slope of enlightenment, and plateau of production), and to provide a source of insight to manage its deployment.

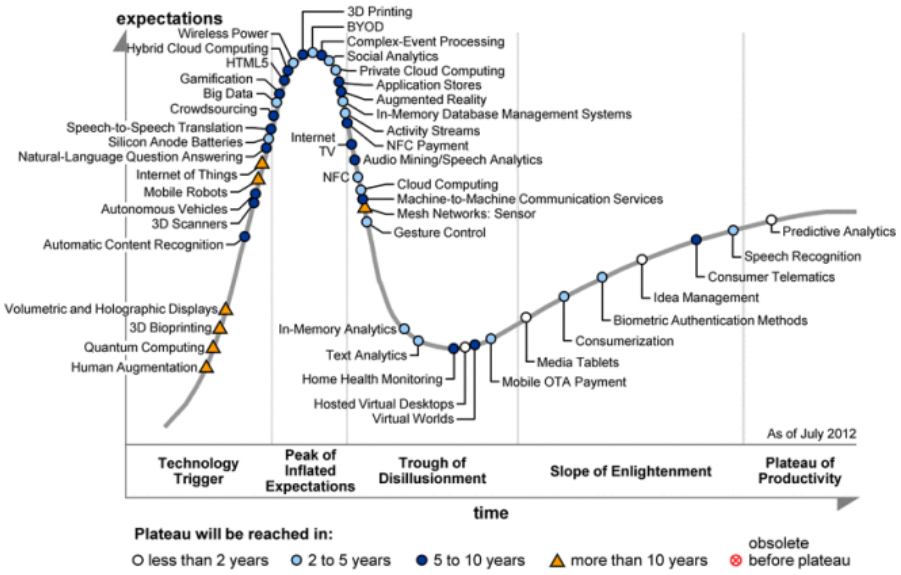

Figure 2.1: Gartner hyper cycle graph representing five phases of a technology (http://en.wikipedia.org/wiki/Hype_cycle).

Applying the Gartner hyper cycle to the artificial neural network development, we created Figure 2.2 to align different generations of the neural network with the various phases designated in the hype cycle. The peak activities ("expectations" or "media hype" on the vertical axis) occurred in late 1980s and early 1990s, corresponding to the height of what is often referred to as the "second generation" of neural networks. The deep belief network (DBN) and a fast algorithm for training it were invented in 2006 [163, 164]. When the DBN was used to initialize the DNN, the learning became highly effective and this has inspired the subsequent fast growing research ("enlightenment" phase shown in Figure 2.2). Applications of the DBN and DNN to industry-scale speech feature extraction and speech recognition started in 2009 when leading academic and industrial researchers with both deep learning and speech expertise collaborated; see reviews in [89, 161]. This collaboration fast expanded the work of speech recognition using deep learning methods to increasingly larger successes [94, 161, 323, 414],

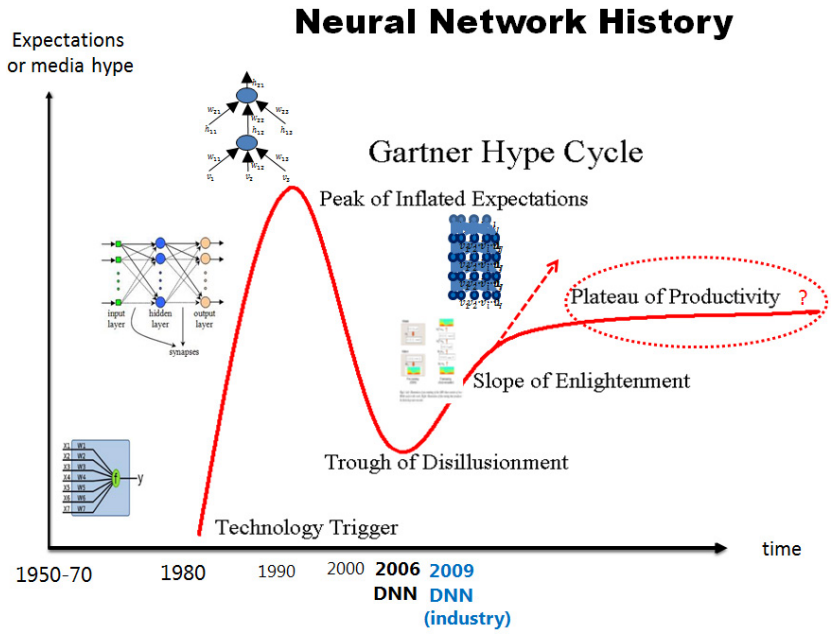

Figure 2.2: Applying Gartner hyper cycle graph to analyzing the history of artificial neural network technology (We thank our colleague John Platt during 2012 for bringing this type of "Hyper Cycle" graph to our attention for concisely analyzing the neural network history).

many of which will be covered in the remainder of this monograph. The height of the "plateau of productivity" phase, not yet reached in our opinion, is expected to be higher than that in the stereotypical curve (circled with a question mark in Figure 2.2), and is marked by the dashed line that moves straight up.

We show in Figure 2.3 the history of speech recognition, which has been compiled by NIST, organized by plotting the word error rate (WER) as a function of time for a number of increasingly difficult speech recognition tasks. Note all WER results were obtained using the GMM–HMM technology. When one particularly difficult task (Switchboard) is extracted from Figure 2.3, we see a flat curve over many years using the GMM–HMM technology but after the DNN technology is used the WER drops sharply (marked by the red star in Figure 2.4).

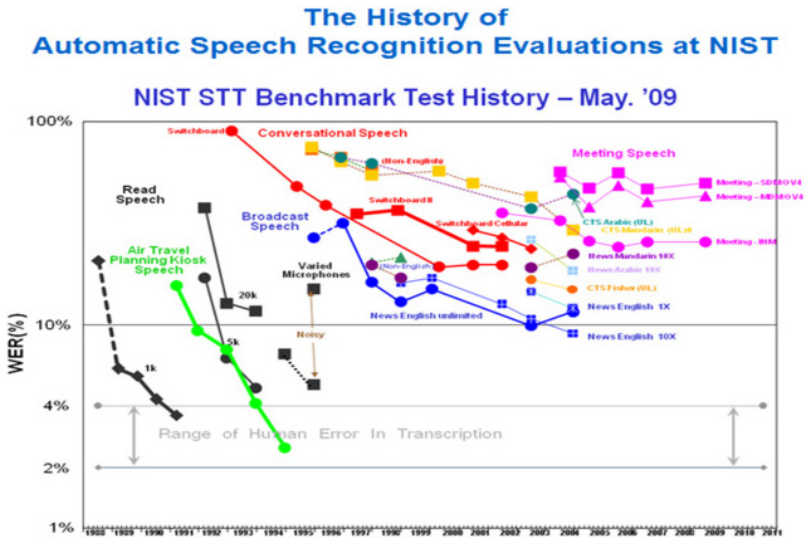

Figure 2.3: The famous NIST plot showing the historical speech recognition error rates achieved by the GMM-HMM approach for a number of increasingly difficult speech recognition tasks. Data source: http://itl.nist.gov/iad/mig/publications/ASRhistory/index.html

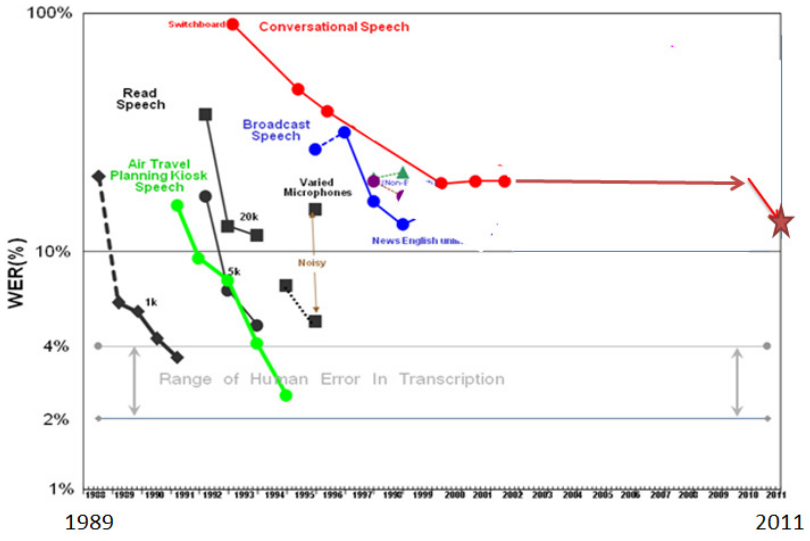

Figure 2.4: Extracting WERs of one task from Figure 2.3 and adding the significantly lower WER (marked by the star) achieved by the DNN technology.

In the next section, an overview is provided on the various architectures of deep learning, followed by more detailed expositions of a few widely studied architectures and methods and by selected applications in signal and information processing including speech and audio, natural language, information retrieval, vision, and multi-modal processing.