Introduction to Deep Learning

Vladimir Feinberg

July 17, 2017

1 Introduction

Many-layer neural networks networks, or deep neural networks, constitute a diverse form of machine learning strategies, which span a broad spectrum of unsupervised, semi-supervised, reinforced, and supervised learning. Inspired by biology, this class of methods is broadly referred to as artificial neural networks (ANNs) or deep learning. Occasionally the term deep neural network comes up, but deep is a relative term useful for distinguishing between two networks. I'll refrain from using such terminology because it creates a false dichotomy in certain situations.

At the highest level, for a given network architecture, an ANN forms a family of possibly nonlinear functions. Such functions may be mappings from a possibly variable-length tensor to another possibly variable-length tensor of real values.

An early, simple example of ANNs is the feed-forward neural network (FFNN), which focuses on fixed-size input and output tensors. This family defines a hypothesis class for empirical risk minimization (ERM) in supervised learning.

These notes follow Dr. Goodfellow's Deep Learning Book. As mentioned there, there are two essential components responsible and necessary for the success ANNs:

1. Expressiveness through depth: iterative compositions of linear functions followed by activations can efficiently model mostly continuous functions (specifics below).

2. Tractable learning: compositions of simple functions as a graph allow for linear-time evaluation, and, critically, linear-time gradients via back-propagation.

Interestingly, these two frequently-cited facts give rise to two perplexing questions about ANNs, which don't seem to yet have a good answer:

1. If DNNs are so expressive, then isn't their VC dimension extremely large? Wouldn't this make them extremely prone to overfitting, to the point that they would be unusable? Perhaps DNNs can model "real distributions" well, so the distribution-independent analysis of VC-based statistical learning theory is insufficient. How can we justify the previous sentence? (Zhang et al 2017, Krueger et al 2017)

2. Why exactly is learning so tractable? Efficient derivatives aren't good enough in a non-convex enviornment: we know we'll reach a local min, but what makes a local min good? (Lee et al 2016, Jin et al 2017, Choromanska et al 2014, Wilson et al 2017)

2 Definition

An ANN f of depth d is a a composition of d vector-valued functions:

$$f(\mathbf{x}) = f^{(d)} \circ f^{(d-1)} \circ \cdots \circ f^{(2)} \circ f^{(1)} (\mathbf{x})$$

Note that the input and output dimension of each $f^{(i)}$ may differ; the maximum such dimension is the width of the network. Each sub-function $f^{(i)}$, or layer, may differ, but has some known and well-structured parameterization. ANN methods then typically diverge with the training algorithm and what they consider their inputs and outputs to be. These iterative compositions of more fundamental layer-wise operations construct what is accurately labeled by Dr. Karpathy as “real valued circuits,” where simple building blocks construct complex output behaviors through tractable operations.

For instance, a FFNN treats inputs $\mathbf{x}$ as those inputs from a supervised learning problem, and attempts to find a mapping f between observed pairs $(\mathbf{x}, y)$. A multi-layer perceptron (MLP), a type of FFNN, is the composition of a vectorized activation function g and an affine function. The dimensionality of the affine functions and the choice of activation are hyperparameters.

$$f^{(i)}(\mathbf{x}) = g\left(W^{(i)}\mathbf{x} + \mathbf{b}^{(i)}\right)$$

$W^{(i)}, \mathbf{b}^{(i)}$ define parameters for f. The activation g is usually a parameter-free (but perhaps not hyperparameter-free) nonlinearity. The MLP differs from other ANNs in that it is fully connected (all $W^{(i)}$ are unconstrained, so there may be arbitrary linear relationships between the output of $f^{(i-1)}$ and the input to g). A network whose $W^{(i)}$ matrices correspond to convolutions is a convolutional neural network (CNN) and refers to a much more restricted family of functions.

Other networks, such as recurrent neural networks (RNNs) handle variable-length inputs. The Asimov Institute has compiled an extensive list of the networks and acronyms used over the years.

A cost function $J(\boldsymbol{\theta})$ must be defined over the vector of aggregate parameters $\boldsymbol{\theta}$. MLE informs the shape of J in most cases: for $(\mathbf{x}, \mathbf{y})$ distributed according to some data distribution, the log likelihood is the cross entropy $\mathbb{E}\, p_{\text{model}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. With $p_{\text{model}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = N(f_{\boldsymbol{\theta}}(\mathbf{x}), I)$, for instance, MLE is equivalent to minimizing MSE. This kind of approach has the ANN specify the parameters for an output distribution, instead of predicting the output directly: this enables us to encode additional statistical knowledge about the inputs.

Training is typically performed with first-order optimization of $J(\boldsymbol{\theta})$; this leverages back-propagation for efficient derivatives.

3 Activations

Activations like g above are an important design choice for all ANNs. General well-performing ones are listed, though problem-specific considerations will take priority.

ReLU (Glorot et al 2011). The ReLU $g(\mathbf{x}) = \max(\mathbf{x}, \mathbf{0})$ induces activation sparsity, which is more biologically plausible. Their use improves accuracy when compared to smoother activations in practice, and their linear regime keeps magnitude scaling behavior reasonable as we traverse the net, which is a boon to optimization.

Maxout (Goodfellow et al 2013, Goodfellow et al). A maxout layer is a generalized ReLU g with input dimension n and fixed p and k as hyperparameters has fixed subsets $G^{(i)} = \{ik + j \mid j \in [p]\}$ such that $g(\mathbf{z})_i = \max_{j \in G^{(i)}} z_j$. In addition to being more theoretically expressive (allowing for fewer units in the next layer), maxout enables units to be driven by several “filters” (activations in the previous layer), preventing catastrophic forgetting. Moreover, maxout is complimentary to dropout.

4 Theory Overview

Universal Approximation Theorem. A feed-forward network with a linear output layer, at least one hidden layer, and the logistic sigmoid activation can approximate any Borel measurable function and its derivatives, where they exist, arbitrarily well, given sufficient hidden units.

Deep Rectifier Network Efficiency Theorem (Montúfar et al 2014). A deep rectifier network of constant width w, input dimension i and depth d may have up to $\Omega\left(\left({}^{w}\!/_{i}\right)^{di} i^{i}\right)$ distinct linear regions and a maxout network with k filters per unit may have up to $\Omega(k^{i+l-1})$. Visually demonstrated in Figure 1.

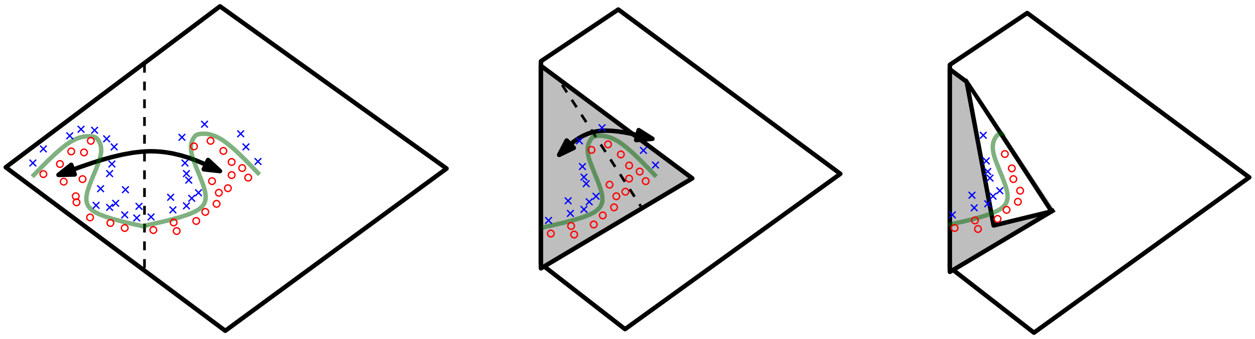

Figure 1: Visual demonstration of how composing an absolute value rectifier, which in effect allows for a higher-level function to be mirrored over some hyperplane in the input space, can result in exponentially many different linear regions by relying on (learned) symmetry.