Deep Learning

Chapter 12 - Applications

Ike Okonkwo - @ikeondata

Applications

This chapter explains how we use deep learning to solve problem across different problem types like computer vision, speech recognition and natural language processing. The chapter is broken down as follows :

- 12.1 Large Scale Deep Learning
- 12.2 Computer Vision
- 12.3 Speech Recognition
- 12.4 Natural Language Processing
- 12.5 Other Applications

12.1.1 Fast CPU Implementations

Deep Learning based on connectionism. Individual neuron or feature in an ML model is not intelligent but a large population of these neurons and features acting together exhibit intelligent behavior.

We've seen network sizes grow dramatically in the past 2 decades and we've also seen some acceleration since 2012 because of new idea and cheap compute

AlexNet(2012) - 10 layers, ZF Net(2013) , VGGNet(2014) - 19 layers , GoogLeNet(2015) - 22 layers, ResNet(2015) - 152 layers

https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html

12.1.1 Fast CPU Implementations

NN were traditionally trained on single machine CPUs. Today we mostly use GPUs or clusters of CPUs.

Custom NN implementation on huge CPU families yielded huge improvements , eg 3x speed up when using fixed-point implementation vs strong floating point system

Optimizing data structures to avoid cache misses and using vector instructions also yielded huge payoffs

12.1.2 GPU Implementations

GPUs are specialized hardware components originally built for graphics / gaming applications. Video games rendering requires performing many operations in parallel quickly which turns out to be beneficial to training NN

GPUs perform simple matrix math on many vertices in parallel to convert 3D coordinates to 2D on-screen coordinates. They must also perform simple computations at each pixel in parallel to determine the color of each pixel.

Computations also involve processing massive buffers of memory containing bitmaps of objects to be rendered.This results in GPUs being design to have high parallelism and memory bandwidth at the expense if clock speed.

12.1.2 GPU Implementations

NN algorithms require the same performance characteristics of rela time graphics algorithms.NN have lots of parameters, activations and gradient value which must be updated at each step of training.

Since NN neurons in the same layer can be processed independently from each other, they easily benefit from GPU parallelism.

GPUs are more flexible today. GP-GPUs can execute arbitrary code not just rendering subroutines thanks to NVIDIA CUDA language.

Writing efficient code for these GP-GPUs can be challenge

12.1.2 GPU Implementations

It recommended to use library with high performance matrix and convolution operations vs writing custom GPU code, eg Pylearn2 calls to Theano and cuda-convnet.This allow same Theano code to run on both CPU and GPU

12.1.3 Large Scale Distributed Implementations

We can use data parallelism to distribute inference across many machines. We can also get model parallelism when multiple machines work on different parts of a single data point.

Data parallelism during training solved by async SGD and using a parameter server to manage the parameters vs store in shared memory

Distribution async GD remains the primary strategy for training large deep networks

12.1.4 Model Compression

Here, we want to replace the original expensive model with a smaller model that requires less memory and runtime to store and evaluate

Learning one large model is prefered to an ensemble for compression because the large model learns a function with many more parameter than are necessary for task as hand

12.1.5 Dynamic Structure

We can accelerate data processing systems by building those with dynamic computation graphs. The can dynamically determine which subsets of many NN can be run on a given input.

Conditional computation computes features on when they are needed.

Using a cascade of classifiers may be applied to detecting a rare object.Training a sequence of classifiers with low capacity / high recall is a better strategy. The final classifier in the cascade is trained to have high precision.

Cascade systems can be engineered to have high capacity by either by making the later members of the cascade have high capacity or making each individual member with low capacity by the system as a whole is high capacity due to the combination of very small models

12.1.5 Dynamic Structure

In the cascade structure the early members could help localize an objects while the later members perform further pre-processing, eg Google uses a two cascade to locate an address number frn Street View and then transcribe it.

An interesting union of decision trees and deep learning would be to put a NN at each node in the decision tree to determine to make splitting decisions

Dynamic structures can also be achieved by using a gater or switch

One major drawback to dynamically structured systems is the decreased degree of parallelism on both CPUs and GPUs

12.1.6 Specialized Hardware Implementation of Deep Networks

Specialized hardware like ASICs, hybrid chips and FPGAs have been used to implement deep learning models

Recent improvements have come from parallelization from multiple cores

Using fixed point rather than floating point representations reduces the hardware surface area, power requirements and computing time needed for performing multiplications which is the most demanding of the operations needed to use or train a modern deep network with back propagations

12.2 Computer Vision

Computer Vision has been one of the most active areas of research and many of the standard benchmarks are image problems like MNIST, CIFAR, ImageNet

Computer Vision is a broad field with a huge diversity of application.The vision interface is become the next big platform

Most current deep learning for Computer Vision is centered around Object Detection

12.2.1 Preprocessing

Computer Vision requires minimal processing of input besides standardized inputs. Images may be cropped to get this but it’s not always necessary.

Dataset augmentation is preprocessing routine and is an excellent way to reduce generalization error in computer vision models. Another way is image showing the model different version of the same input.

Reducing the amount of variation in the data can reduce both generalization error and the size of the model needed to fit the training data and simpler models tend to generalize well.

For large models, model - determined invariant factors is the suggested approach

12.2.1.1 Contrast Normalization

Contrast is the magnitude of the difference between the light and dark pixels in an image. With respect to deep learning, contrast refers to the stdev of the pixels in an image or a region of an image

GCN aims to prevent images from varying amount of contrast by normalizing the images (subtract mean and then rescale)

GCN can be understood to map examples to a spherical shell thus reducing ea example to just a direction. GCN is similar to sphering. Sphering rescale the principal components to have equal variance. Also called whitening

12.2.1.1 Contrast Normalization

GCN often fails to highlight image features such as edges and corners

LCN however ensures that contrast is normalized across a small window rather than the whole image.

12.2.1.2 Dataset Augmentation

We can typically improve classifier generalization by increasing the training set by feeding it copies of training examples that have been modified with some transformations with changing the class (rotating, cropping, random translations, random perturbations of the colors in an image,geometric distortions)

12.3 Speech Recognition

Speech recognition maps an acoustic signal containing a spoken natural language utterance into the corresponding sequence of words intended by the speaker

Before 2009, state of the art speech recognition systems were a combination of HMM and GMM. HMM modeled the association between the acoustic features and phonemes whereas HMM modeled the sequence of phonemes

With much larger models / datatsets, deep learning has essentially replace GMM based systems for ASR.

Current systems use a type of unsupervised deep learning model called Restricted Boltzmann Machines

12.3 Speech Recognition

Over time deep networks for speech recognition have shifted from being based on pretraining and RBMs to ReLUs and dropout

Today, most industrial speech recognition systems have incorporated deep neural networks in their products.

Other recent innovations include using CNNs and LSTM RNNs to train to