CEng 783 – Deep Learning

Fall 2017 - Week 1

Emre Akbas
Deep Learning (DL)

• DL is a branch of **machine learning**.

• **Machine learning:**

  "A computer program is said to learn from experience \( E \) with respect to some class of tasks \( T \) and performance measure \( P \) if its performance at tasks in \( T \), as measured by \( P \), improves with experience \( E \)."

  Prof Tom Mitchel
Deep Learning (DL)

DL: the set of algorithms and models that work on multilayer graphs called artificial neural networks which were inspired by the structure and function found in the biological brain.

In a DL model, multiple layers of linear and non-linear operations are carried out.

“Deep” implies more than 1 hidden layer.
A little bit of history

- Artificial neural networks are not new (Rosenblat’s “Perceptron” dates back to 1958).
- Their expressive power has been known.
- In 1957, Kolmogorov proved that a three layer neural network, given sufficient number of hidden units, proper nonlinearities and weights, can represent arbitrary continuous functions from input to output.
A little bit of history

- Kolmogorov’s theorem has been refined by others:
  - Cybenko 1989 - Universal approximation theorem

Approximation by Superpositions of a Sigmoidal Function*

G. Cybenko†

Abstract. In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functionals can uniformly approximate any continuous function of n real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

Key words. Neural networks, Approximation, Completeness.

1. Introduction

A number of diverse application areas are concerned with the representation of general functions of an n-dimensional real variable, x ∈ R^n, by finite linear combinations of the form

\[ \sum_{j=1}^{N} a_j \sigma(y_j^T x + \theta_j), \]

where y_j ∈ R^n and a_j, \theta_j ∈ R are fixed. (y^T is the transpose of y so that y^T x is the inner product of y and x.) Here the univariate function \sigma depends heavily on the context of the application. Our major concern is with so-called sigmoidal \sigma’s:

\[ \sigma(t) = \begin{cases} 1 & \text{as } t \to +\infty, \\ 0 & \text{as } t \to -\infty. \end{cases} \]

These theorems are about the representation power of NNs not about their learnability.
ANNs are not new

- In fact, with the advent of “backpropagation” they were popular again in late 80s, early 90s.
- Until the support vector machines and the kernel trick took over.
- With “deep learning,” ANNs are back on stage.
- Then, what was wrong in the 90s? And, what changed?
Backpropagation

- Error signal
- Desired output
- Output layer
- Hidden layers
- Input layer

Backpropagate the error:
- Compute gradients and update hidden layer weights to reduce the error

Nothing but an application of the “chain rule”
Why did people abandon backpropagation?

- “People in machine learning had largely given up on it because:
  - It did not seem to be able to make good use of multiple hidden layers
  - It did not work well in recurrent networks”

What was **actually** wrong with backpropagation?

“We all drew the wrong conclusions about why it failed. The real reasons were:

1. Our datasets were thousands of times too small.
2. Our computers were millions of times too slow.
3. We initialized the weights in a stupid way.
4. We used the wrong type of non-linearity.”

DL: the comeback of neural networks

In 2009, G. Hinton and his students developed a method to train a deep network for speech recognition.

First, they trained each layer (one layer at a time) in an unsupervised manner.

Then, they added a final layer of supervision and then ran backpropagation.
DL: the comeback of neural networks

This deep network outperformed a long standing state-of-the-art on a medium sized speech recognition dataset.

DL: the comeback of neural networks

This speech recognition deep network was in use in the Android operating system as early as 2012.
DL: the comeback of neural networks

New regularization technique: Dropout

Prevented overfitting

Was very important for the success of the DL model
The second success story

In computer vision.

ILSVRC 2012: A competition.

1.2 million images, 1000 categories

The task: given an image, make 5 predictions for the dominant object in the image. If one of them is correct, then it is counted as a success.
The second success story

G. Hinton’s student Alex Krizhevsky trained a 7-layer convolutional neural network (which is now known as AlexNet). [Krizhevsky, Sutskever & Hinton 2012]

Used the same trick (Mohamed, Dahl, Hinton 2009) to train the network.
The second success story

AlexNet achieved 16% error rate.

While the second best algorithm, which was a combination of best computer vision algorithms back in 2012 (namely, weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively) achieved 26%.
The second success story

The second success story

Since then, deep learning based participants dominated the ILSVRC competitions.

In 2015, the error rate went down to ~5% (which is on par with the human error rate).

And, many other success stories so far.
Object detection

(Figure from Ren, He, Girschik, Sun, NIPS 2015)
Object detection

(Figure from “Fast RCNN slides” by R. Girshik)
Machine Translation

State-of-the-art machine translation using Recurrent Neural Networks (RNNs)

Using a combination of Encoder and Decoder RNNs

[Sutskever, Vinyals and Le 2014]

(Figure from Hinton’s Royal Society talk)
Image captioning

A group of people shopping at an outdoor market.
(GT: People are crouched around in an open market.)

A close up of a child holding a stuffed animal
(GT: A young girl asleep on the sofa cuddling a stuffed bear.)

(Figure from Hinton’s Royal Society talk)
Image captioning

\[ f = (a, \text{ man}, \text{ is, jumping, into, a, lake, .}) \]

[Donahue et al. 2015]
Generative deep networks

Image generation:

*Figure 1. Image completions sampled from a PixelRNN.*

(From Oord, Kalchbrenner, Kavukcuoglu 2016)
Generative deep networks: image generation

(Figure from Gatys, Ecker, Bethge 2015)
Generative deep networks: image generation

METU’s “Bilim Agaci Heykeli” painted in the style of Van Gogh’s Starry Night. Generated by Deep Dream (http://deepdreamgenerator.com/ )
Neural Turing Machines

Train the network with input & outputs to learn an algorithmic solution to a problem. E.g., sorting numbers, convex hull, TSP, etc.

[Kurach et al 2015]
Other applications

• NVIDIA’s driverless car
  – end-to-end deep reinforcement learning
  – https://youtu.be/-96BEOXJMs0

• Visual Query Answering:
  http://cloudcv.org/vqa/

• Music generation:
  https://www.youtube.com/watch?v=0VTI1BBLydE
Why deep learning

Performance vs. Amount of data

How do data science techniques scale with amount of data?

[Slide by Andrew Ng]


References


