Mathematics of AI

Deep Learning for Applied Mathematicians

Higham & Higham (2019)

This section

Deep learning and related topics

- Week 6: Deep Learning for Applied Mathematicians

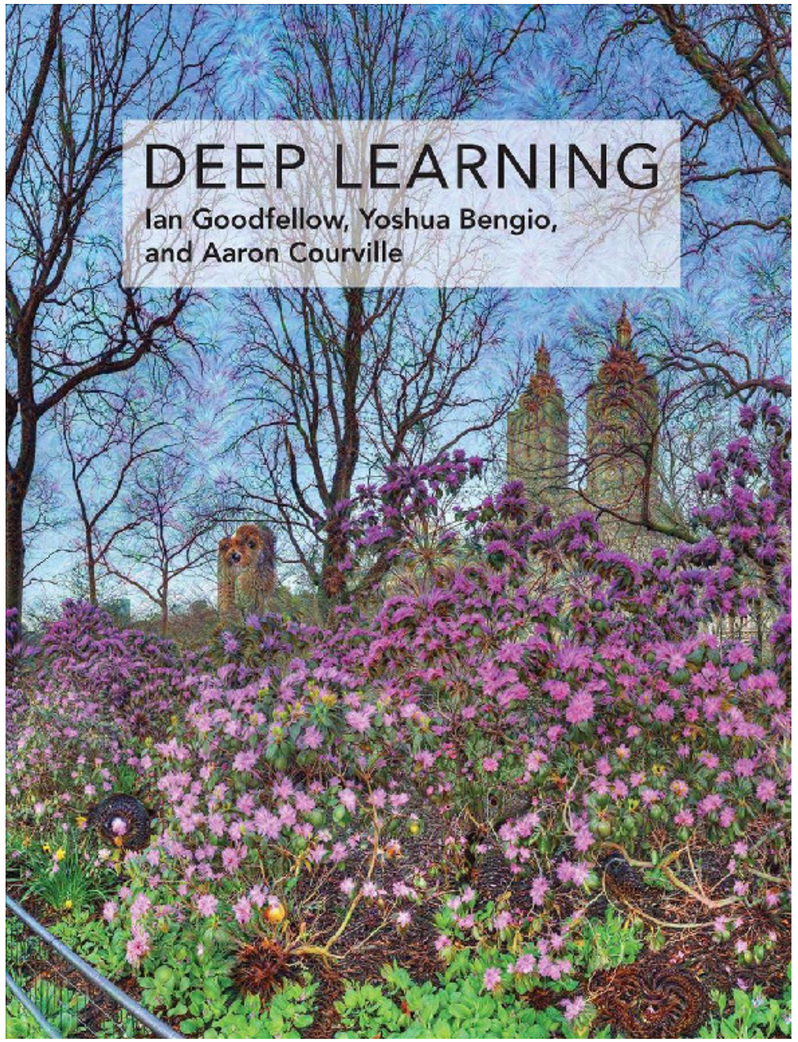
- Week 7: Further topics in optimisation
  - SGD and variants
  - Batch and minibatch algorithms
  - Practical issues (+ PyTorch implementation)

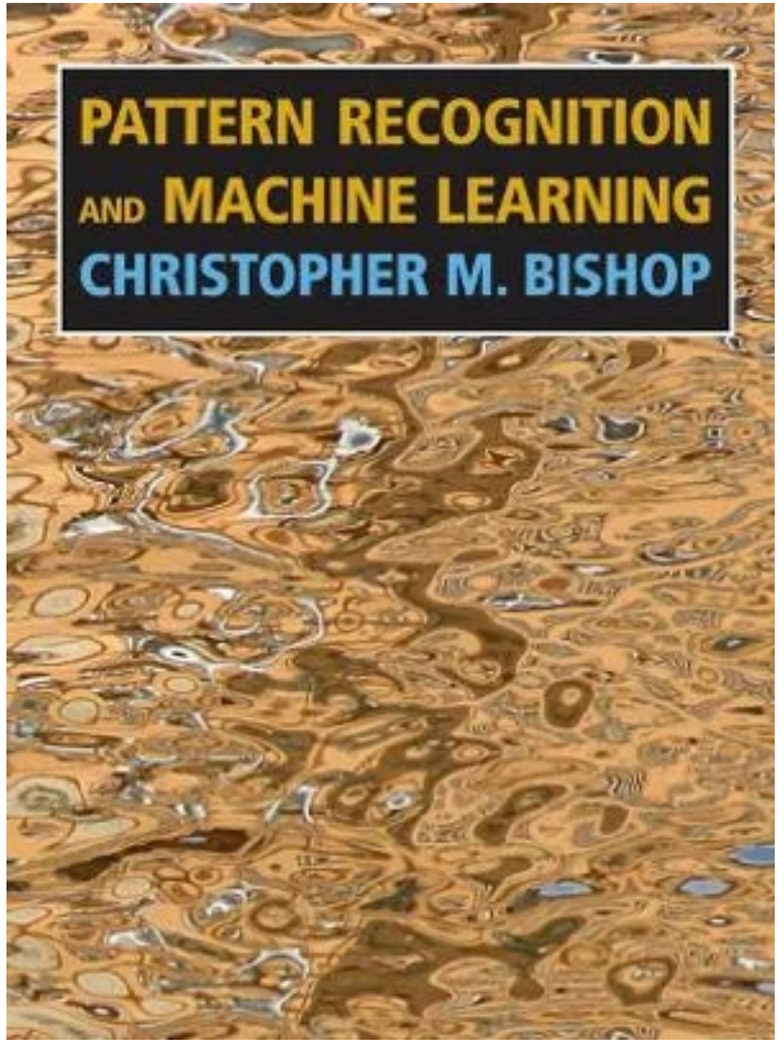
- Week 8: Regularisation
  - In “traditional” learning
  - In deep learning

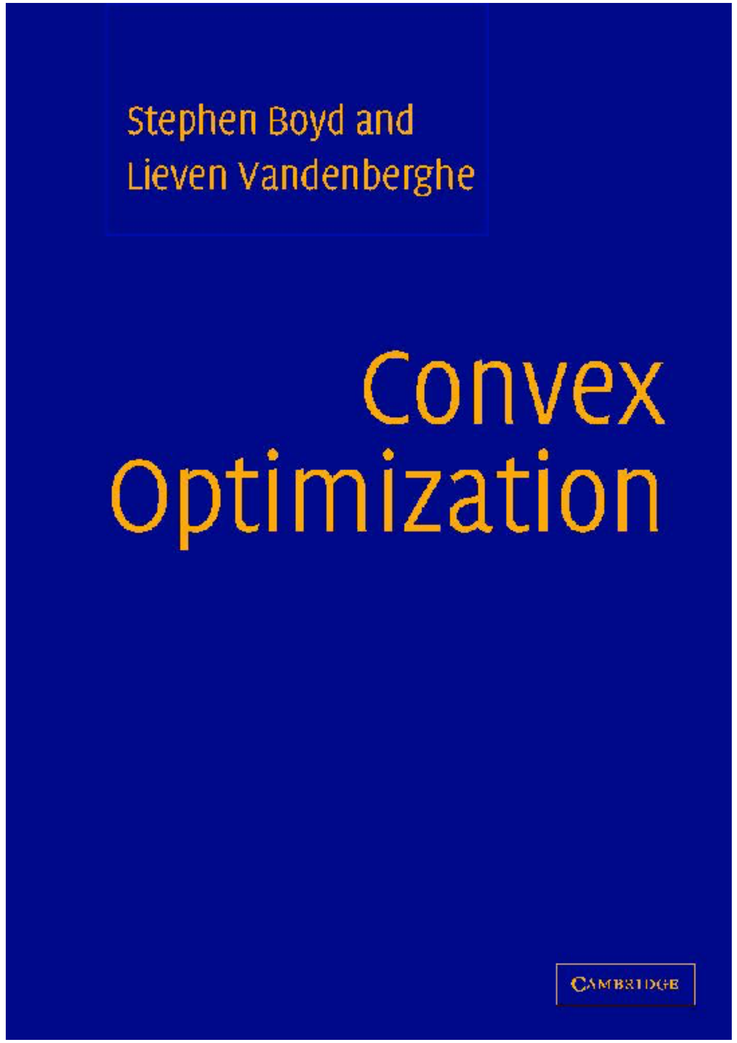
- Week 9: Compressive sensing redux
  - Regularisation + compression!

Resources

My favourite books for this section (all in Course Readings)







Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.

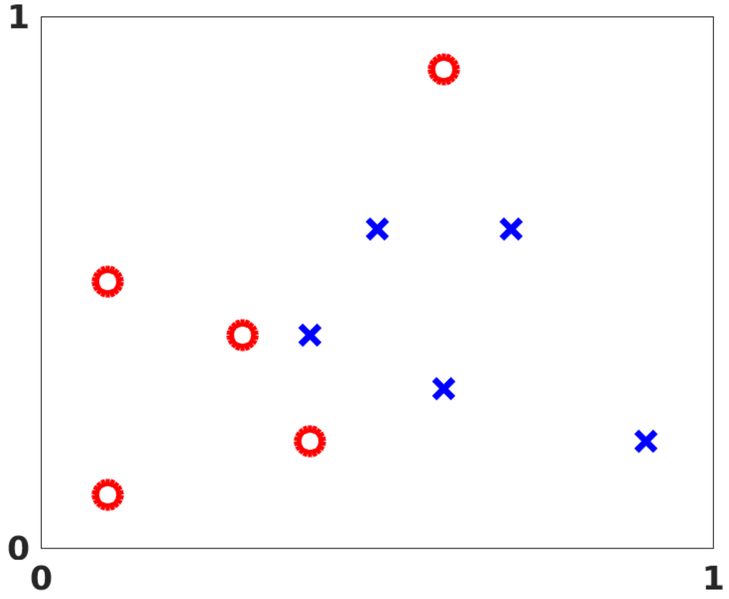
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.



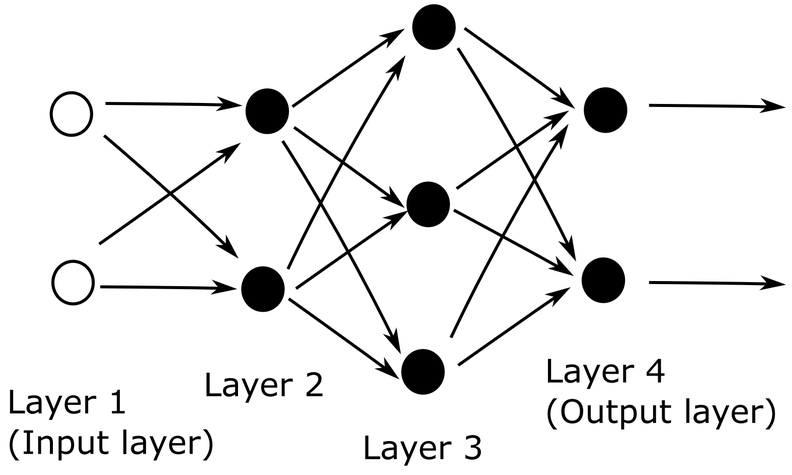
Mathematics > History and Overview

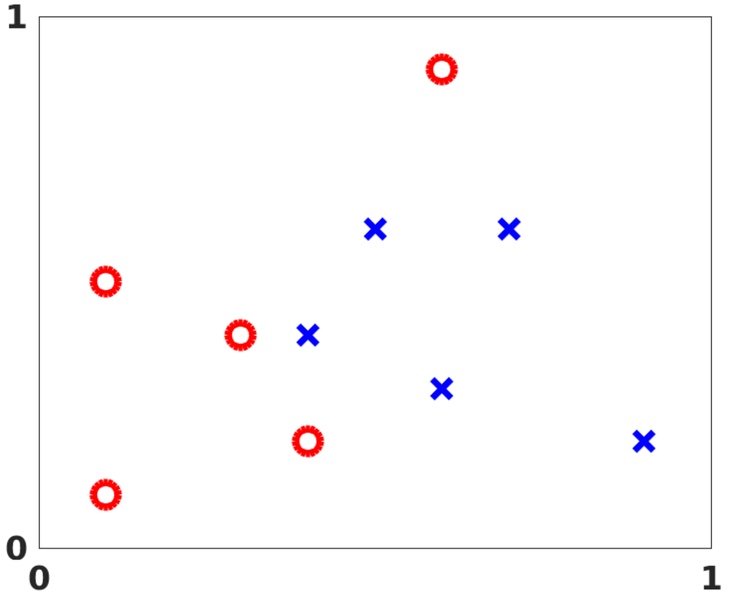
[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.





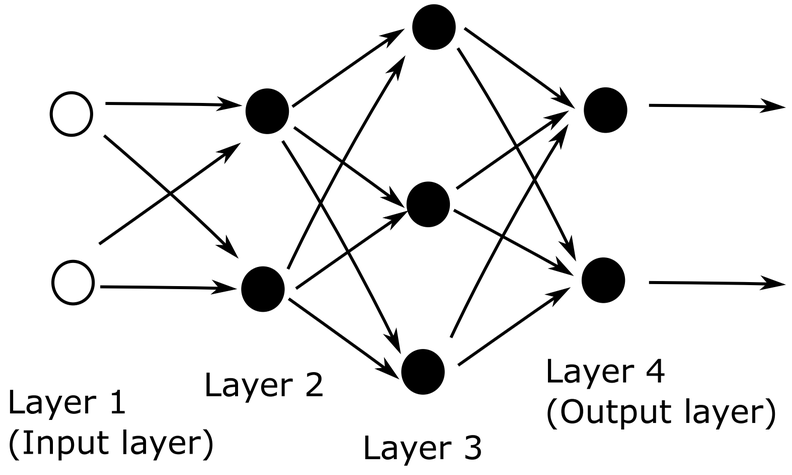
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

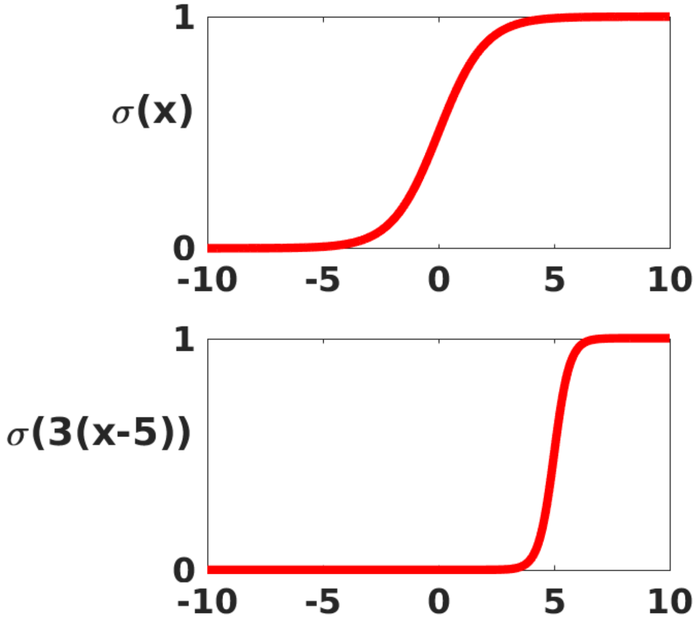
Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



https://arxiv.org/abs/1801.05894

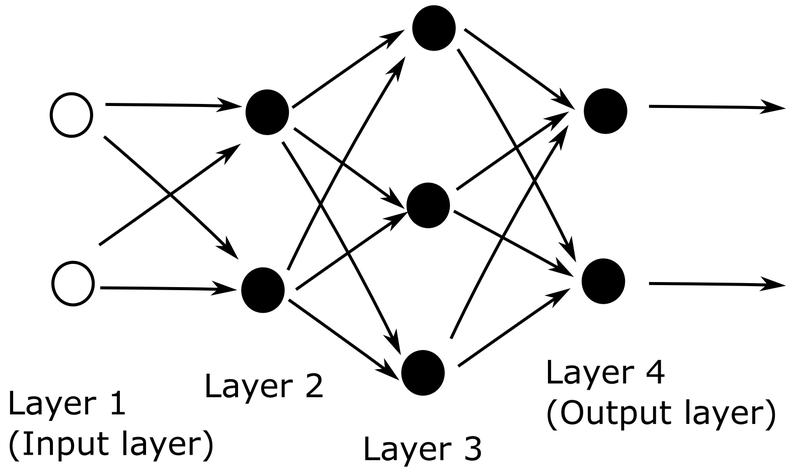
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

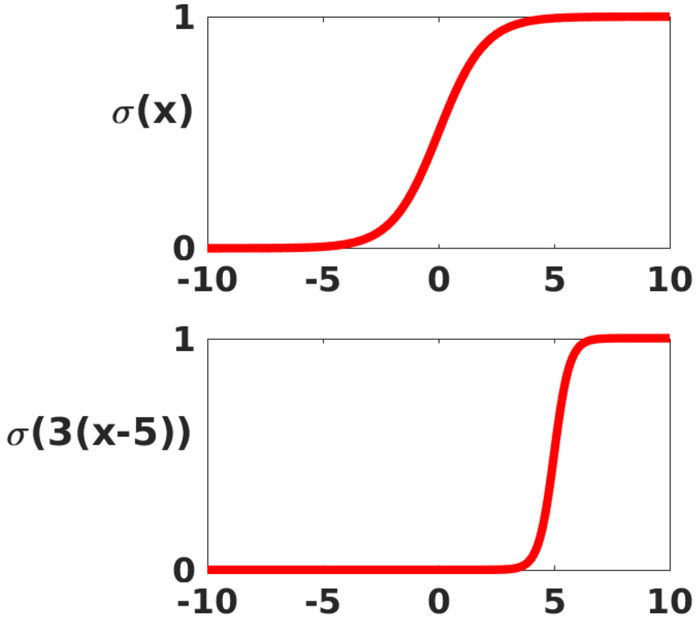
Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma\left(\sum_j w_{ij}a_j + b_i\right)$$



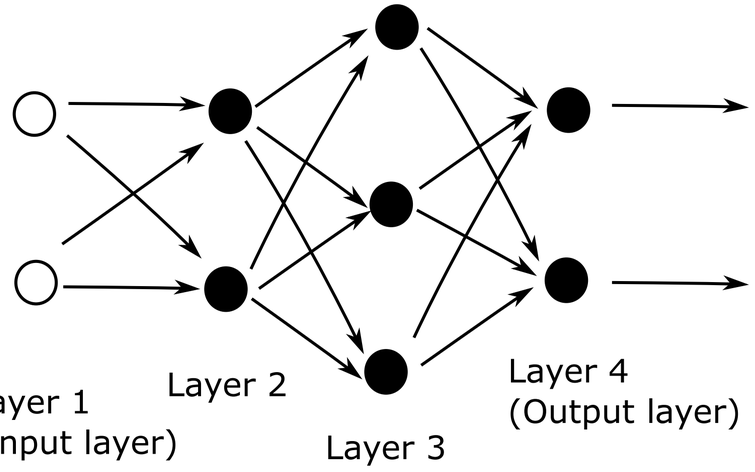
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.



$$F(x) = \sigma\left(W^{[4]}\sigma\left(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}\right) + b^{[4]}\right) \in \mathbb{R}^2.$$

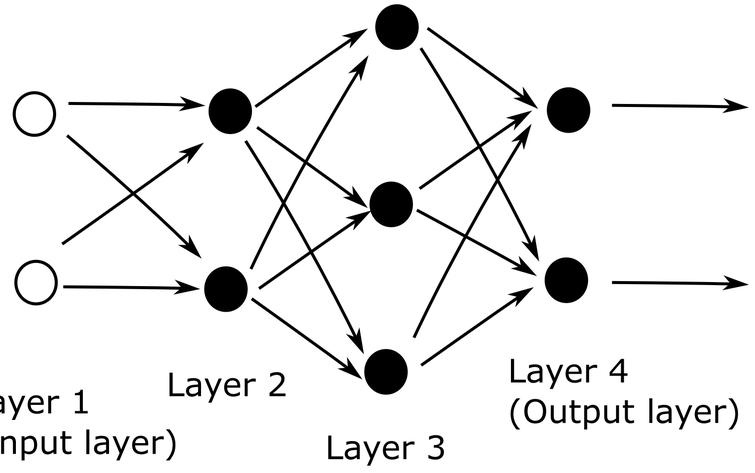
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host of application fields. At the heart of this deep learning revolution are familiar concepts from applied and computational mathematics; notably, in calculus, approximation theory, optimization and linear algebra. This article provides a very brief introduction to the basic ideas that underlie deep learning from an applied mathematics perspective. Our target audience includes postgraduate and final year undergraduate students in mathematics who are keen to learn about the area. The article may also be useful for instructors in mathematics who wish to enliven their classes with references to the application of deep learning techniques. We focus on three fundamental questions: what is a deep neural network? how is a network trained? what is the stochastic gradient method? We illustrate the ideas with a short MATLAB code that sets up and trains a network. We also show the use of state-of-the art software on a large scale image classification problem. We finish with references to the current literature.



$$F(x) = \sigma\left(W^{[4]}\sigma\left(W^{[3]}\sigma(W^{[2]}x + b^{[2]}) + b^{[3]}\right) + b^{[4]}\right) \in \mathbb{R}^2.$$

$$\text{Cost}\left(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]}\right) = \frac{1}{10}\sum_{i=1}^{10} \frac{1}{2}\|y(x^{\{i\}}) - F(x^{\{i\}})\|_2^2.$$

https://arxiv.org/abs/1801.05894

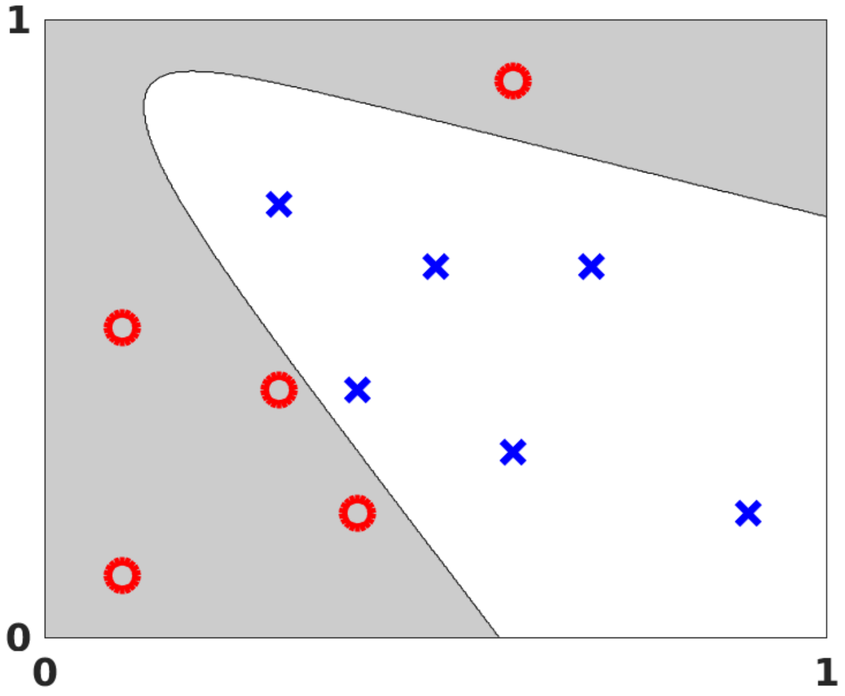
Mathematics > History and Overview

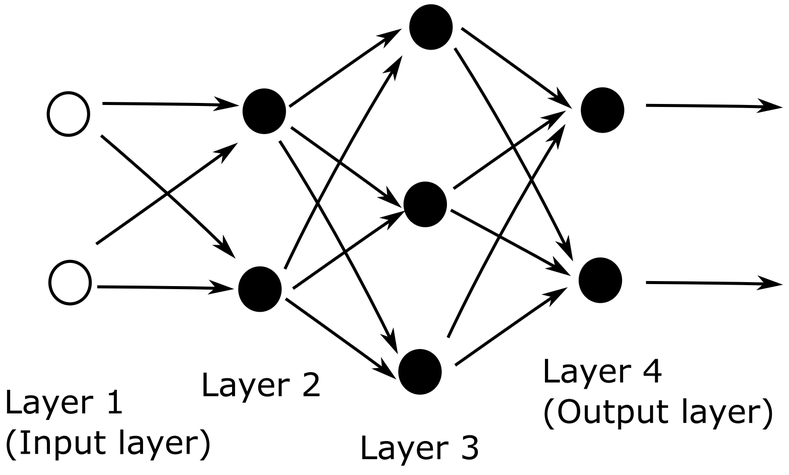
[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Multilayered artificial neural networks are becoming a pervasive tool in a host
revolution are familiar concepts from applied and computational mathematics;
linear algebra. This article provides a very brief introduction to the basic ideas
perspective. Our target audience includes postgraduate and final year undergr
the area. The article may also be useful for instructors in mathematics who wis
deep learning techniques. We focus on three fundamental questions: what is a
stochastic gradient method? We illustrate the ideas with a short MATLAB code
state-of-the art software on a large scale image classification problem. We fin





$$\text{Cost}\left(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]}\right) = \frac{1}{10}\sum_{i=1}^{10} \frac{1}{2}\|y(x^{\{i\}}) - F(x^{\{i\}})\|_2^2.$$

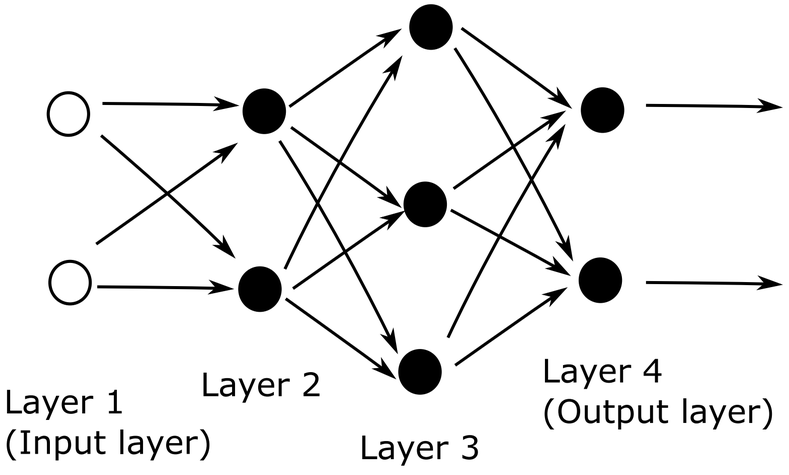
Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

How many parameters?



$$\text{Cost}\left(W^{[2]}, W^{[3]}, W^{[4]}, b^{[2]}, b^{[3]}, b^{[4]}\right) = \frac{1}{10}\sum_{i=1}^{10} \frac{1}{2}\|y(x^{\{i\}}) - F(x^{\{i\}})\|_2^2.$$

https://arxiv.org/abs/1801.05894

Deep Feedforward Networks

Lecture slides for Chapter 6 of *Deep Learning*
www.deeplearningbook.org
Ian Goodfellow
Last updated 2016-10-04

Roadmap

- Example: Learning XOR
- Gradient-Based Learning
- Hidden Units
- Architecture Design
- Back-Propagation

XOR is not linearly separable

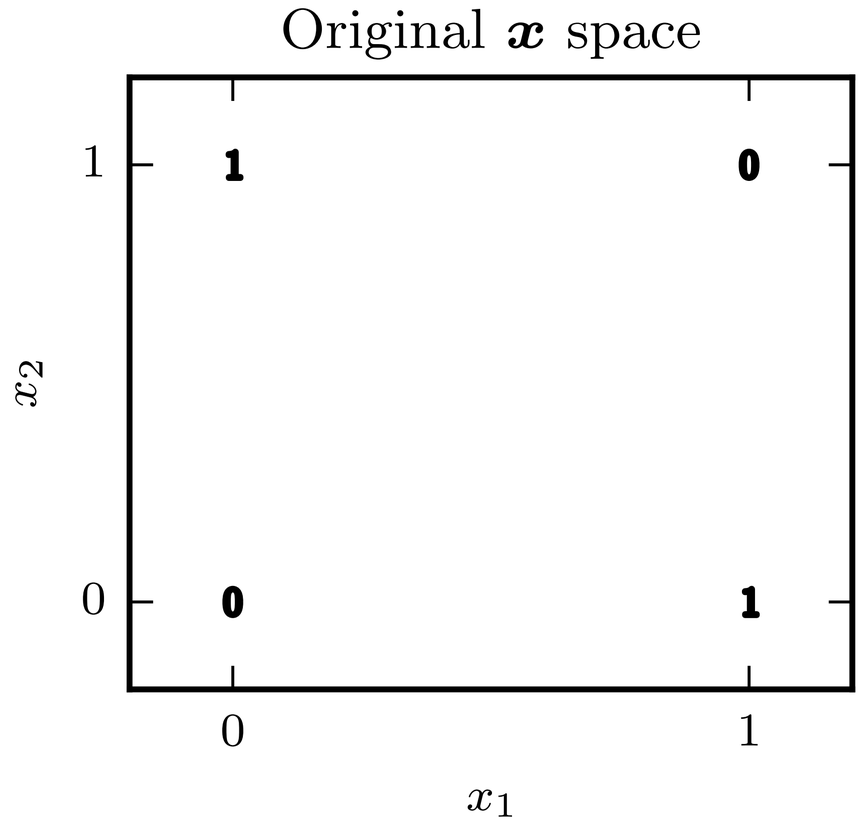


Figure 6.1, left

Rectified Linear Activation

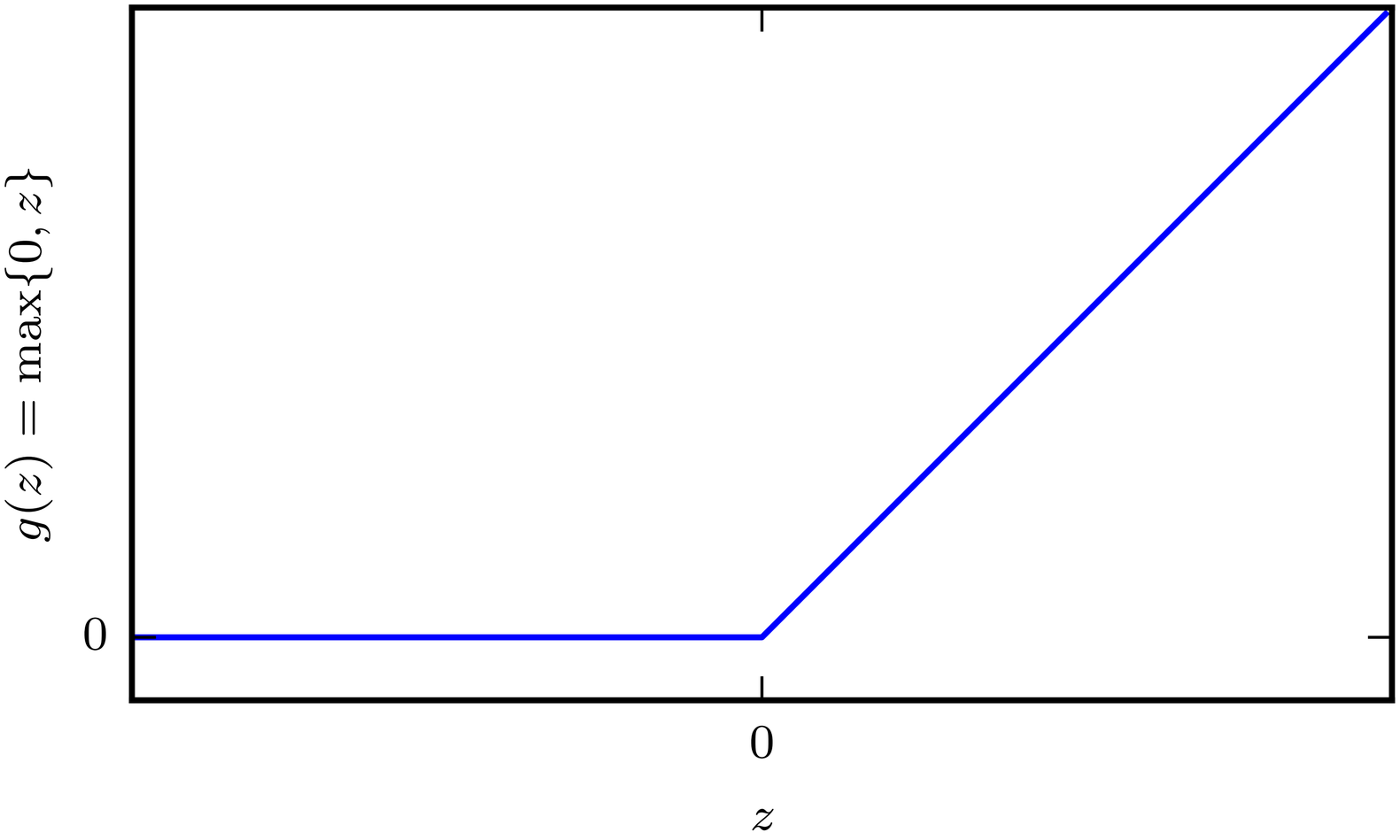


Figure 6.3

Network Diagrams

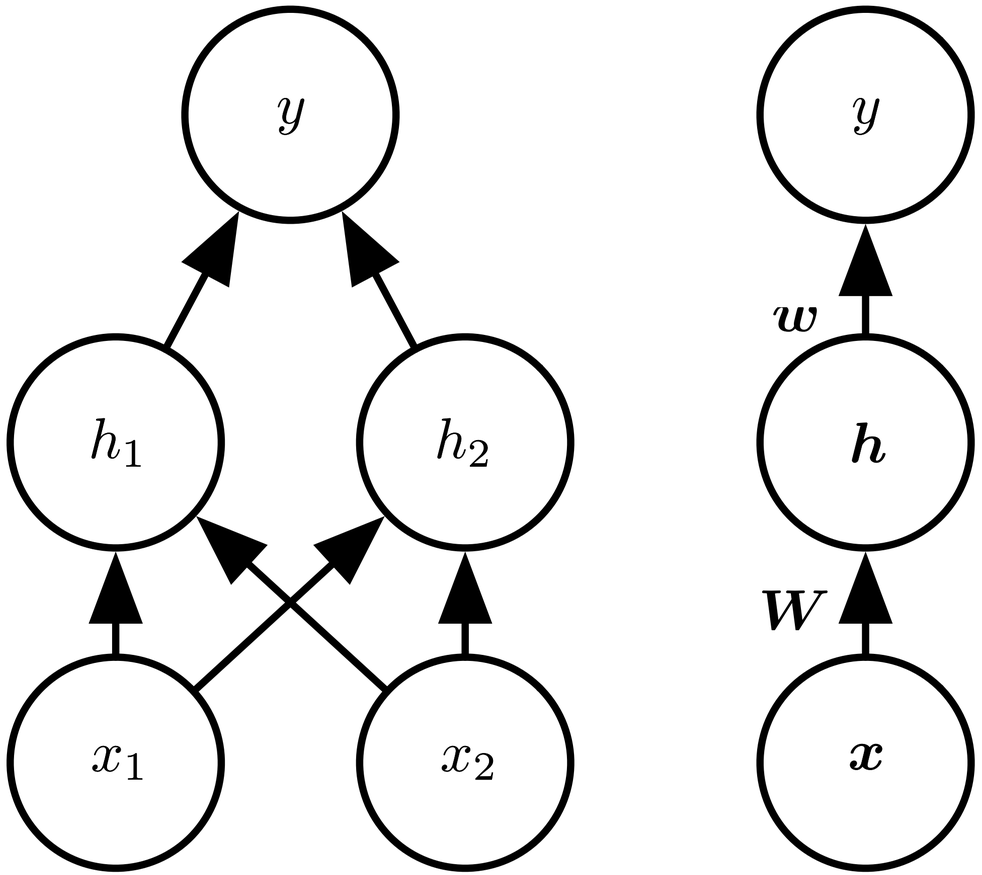


Figure 6.2

Solving XOR

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b. \tag{6.3}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \tag{6.4}$$

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \tag{6.5}$$

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \tag{6.6}$$

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Exercise: transform **X** by hand.

Solving XOR

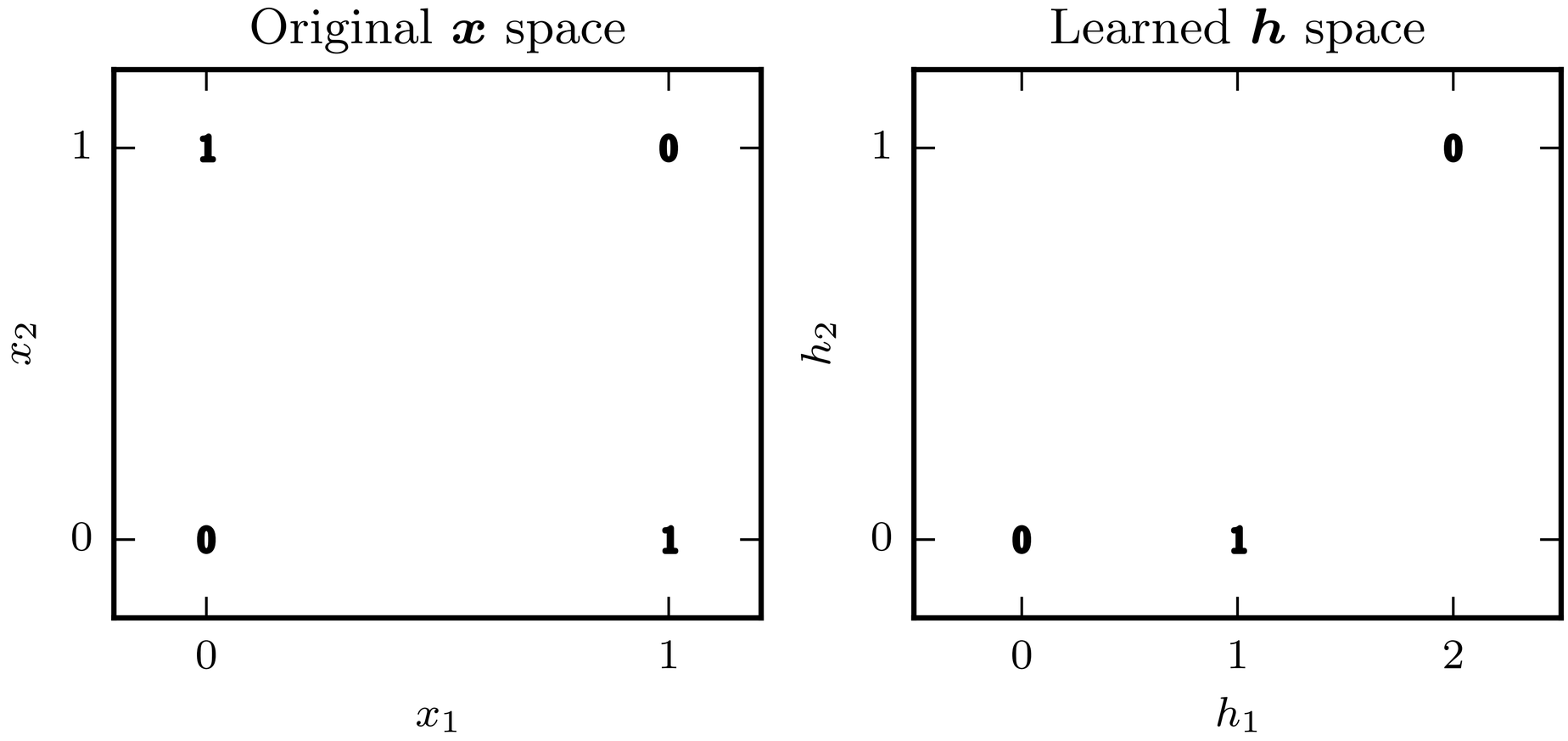


Figure 6.1

Comments/questions

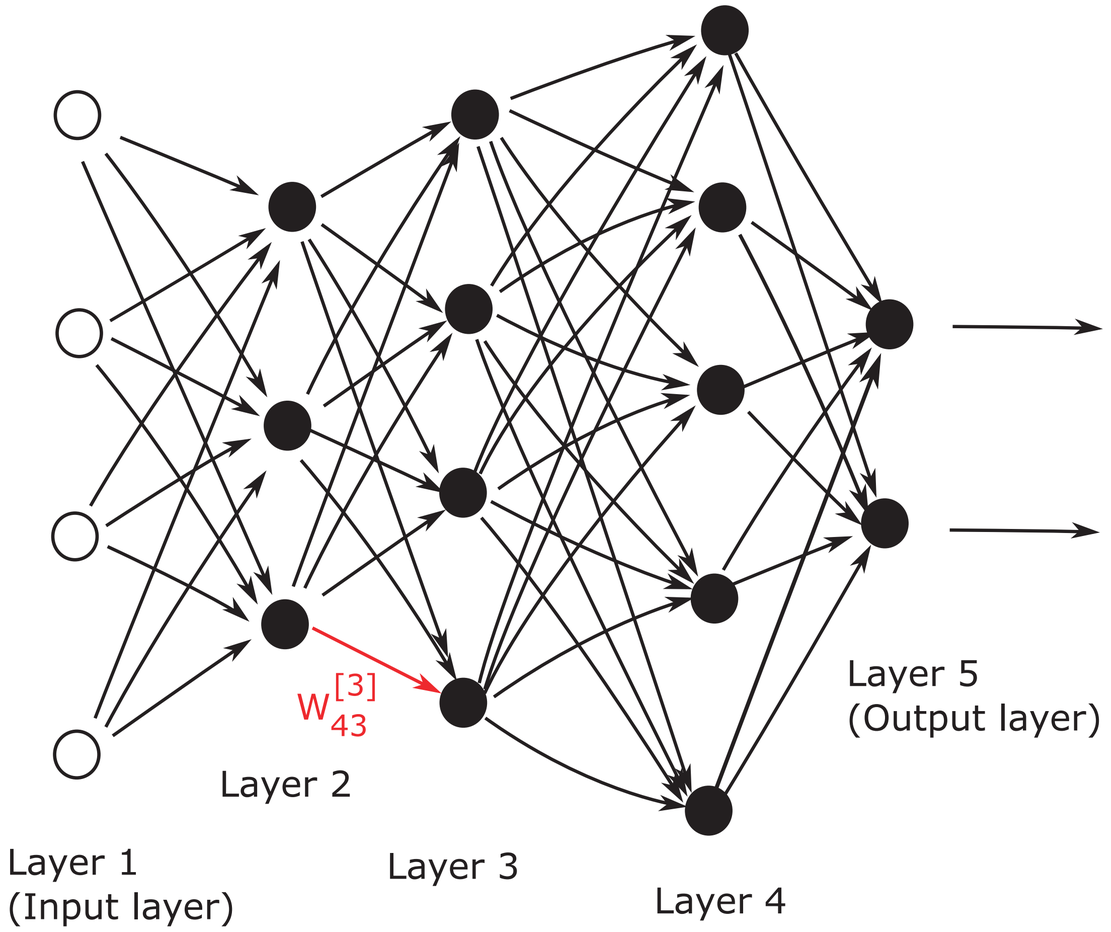
- Note the transformation: stretched into a “larger” space, but low-dimensional. (This is a theme!)
- This is a global minimum: why?
- Are other transformations possible? How many? What does the cost function surface look like?
- Train a neural network to do this: MLWM workshop 1 (in PyTorch!)

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham



How many parameters?

https://arxiv.org/abs/1801.05894

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Stochastic Gradient Descent: 2 approaches

$$p \to p - \eta \nabla \mathrm{Cost}(p). \qquad \text{where} \qquad \nabla \mathrm{Cost}(p) = \frac{1}{N} \sum_{i=1}^{N} \nabla C_{x\{i\}}(p). \qquad \text{Too expensive!}$$

https://arxiv.org/abs/1801.05894

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Stochastic Gradient Descent: 2 approaches

$$p \to p - \eta \nabla \mathrm{Cost}(p). \qquad \text{where} \qquad \nabla \mathrm{Cost}(p) = \frac{1}{N} \sum_{i=1}^{N} \nabla C_{x\{i\}}(p). \qquad \text{Too expensive!}$$

Approach 1: (epochs)

1. Shuffle the integers $\{1, 2, 3, \ldots, N\}$ into a new order, $\{k_1, k_2, k_3, \ldots, k_N\}$.
2. For $i = 1$ upto N, update

$$(4.7) \qquad p \to p - \eta \nabla C_{x\{k_i\}}(p).$$

https://arxiv.org/abs/1801.05894

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Stochastic Gradient Descent: 2 approaches

$$p \to p - \eta \nabla \mathrm{Cost}(p). \qquad \text{where} \qquad \nabla \mathrm{Cost}(p) = \frac{1}{N} \sum_{i=1}^{N} \nabla C_{x\{i\}}(p). \qquad \text{Too expensive!}$$

Approach 1: (epochs)

1. Shuffle the integers $\{1, 2, 3, \ldots, N\}$ into a new order, $\{k_1, k_2, k_3, \ldots, k_N\}$.
2. For $i = 1$ upto N, update

$$p \to p - \eta \nabla C_{x\{k_i\}}(p). \tag{4.7}$$

Approach 2: (minibatch)

1. Choose m integers, $k_1, k_2, \ldots, k_m$, uniformly at random from $\{1, 2, 3, \ldots, N\}$.
2. Update

$$p \to p - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla C_{x\{k_i\}}(p). \tag{4.8}$$

https://arxiv.org/abs/1801.05894

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Calculating the gradient: Back propagation

The important definitions:

$$C = \tfrac{1}{2}\|y - a^{[L]}\|_2^2. \tag{5.1}$$

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \in \mathbb{R}^{n_l} \qquad \text{for } l = 2, 3, \ldots, L. \tag{5.2}$$

$$a^{[l]} = \sigma\left(z^{[l]}\right) \qquad \text{for } l = 2, 3, \ldots, L. \tag{5.3}$$

$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}} \qquad \text{for } 1 \le j \le n_l \quad \text{and} \quad 2 \le l \le L. \tag{5.4}$$

https://arxiv.org/abs/1801.05894

Mathematics > History and Overview

[Submitted on 17 Jan 2018]

Deep Learning: An Introduction for Applied Mathematicians

Catherine F. Higham, Desmond J. Higham

Calculating the gradient: Back propagation

The important definitions:

$$C = \tfrac{1}{2}\|y - a^{[L]}\|_2^2. \tag{5.1}$$

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \in \mathbb{R}^{n_l} \qquad \text{for } l = 2, 3, \ldots, L. \tag{5.2}$$

$$a^{[l]} = \sigma\left(z^{[l]}\right) \qquad \text{for } l = 2, 3, \ldots, L. \tag{5.3}$$

$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}} \qquad \text{for } 1 \le j \le n_l \quad \text{and} \quad 2 \le l \le L. \tag{5.4}$$

LEMMA 5.1. *We have*

$$\delta^{[L]} = \sigma'(z^{[L]}) \circ (a^L - y), \tag{5.5}$$

$$\delta^{[l]} = \sigma'(z^{[l]}) \circ (W^{[l+1]})^T \delta^{[l+1]} \tag{5.6}$$

$$\frac{\partial C}{\partial b_j^{[l]}} = \delta_j^{[l]} \tag{5.7}$$

$$\frac{\partial C}{\partial w_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]} \tag{5.8}$$

https://arxiv.org/abs/1801.05894

Deep Mysteries of Deep Learning

- Why does it converge at all?
- Why does it work well?
  - non-convex optimisation
  - many local minima
  - generalises to unseen data
- How does regularisation happen?
- Why do high dimensions seem to help?
  - segue into project…