Mathematics of Deep Learning, Summer Term 2020

Week 2

Neural Networks

Philipp Harms Lars Niemann

University of Freiburg

Overview of Week 2

Acknowledgement of Sources

Sources for this lecture:

- Frank Hutter and Joschka Boedecker (Department of Computer Science, Freiburg): Course on Deep Learning.
- Philipp Christian Petersen (Faculty of Mathematics, University of Vienna): Course on Neural Network Theory.

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 1

Multilayer Perceptrons

Philipp Harms Lars Niemann

University of Freiburg

McCulloch and Pitts Neuron

The first neural network was devised by McCulloch and Pitts (1943) in an attempt to model a biological neuron.

Definition

A McCulloch and Pitts neuron is a function of the form

$$\mathbb{R}^d \ni x \mapsto \rho\left(\sum_{i=1}^{d} w_i x_i - \theta\right) \in \mathbb{R}$$

where $d \in \mathbb{N}$, $\rho = \mathbb{1}_{\mathbb{R}_+} : \mathbb{R} \to \mathbb{R}$, and $w_i, \theta \in \mathbb{R}$.

- ρ is called activation function,
- θ is called threshold,
- w_i are called weights, and
- the neuron fires (i.e., returns 1) if the weighted sum of inputs exceeds the threshold.

Multilayer Perceptron

A multilayer perceptron, as introduced by Rosenblatt (1958), links multiple neurons together in the sense that the output of one neuron forms an input to another.

Definition

Let d, $L \in \mathbb{N}$, $L \geq 2$ and $\rho : \mathbb{R} \to \mathbb{R}$. Then a multilayer perceptron (MLP) with d-dimensional input, L layers, and activation function $\rho\colon \mathbb{R} \to \mathbb{R}$ is a function

$$F\colon \mathbb{R}^d \to \mathbb{R}^{N_L}, \qquad F = T_L \circ \rho \circ T_{L-1} \circ \cdots \circ \rho \circ T_1,$$

where ρ is applied coordinate-wise and $T_l\colon \mathbb{R}^{l-1} \to \mathbb{R}^l$ is affine, for each $l \in \{1, \ldots, L\}$ and $N_l \in \mathbb{N}$ with $N_0 = d$.

Recall that an affine map is of the form $x \mapsto Ax + b$ for a matrix A and vector b.

Multilayer Perceptron (cont.)

- In contrast to the McCulloch and Pitts neuron, we now allow arbitrary activation functions ρ.
- Notice that the MLP does not allow arbitrary connections between neurons, but only between those, that are in adjacent layers, and only from lower layers to higher layers.

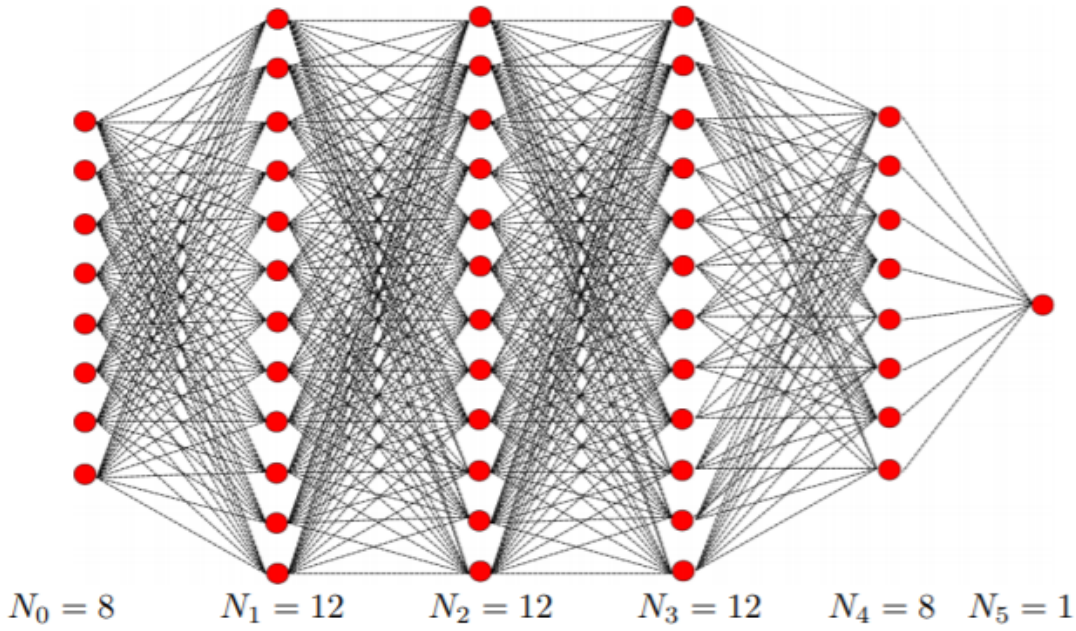


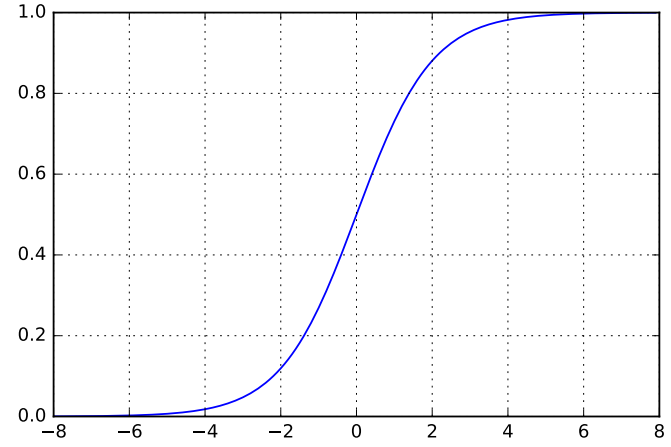
Illustration of a multi-layer perceptron with 5 layers. The red dots correspond to the neurons.

[figure from Petersen, Ch. 1]

Activation Functions - Examples

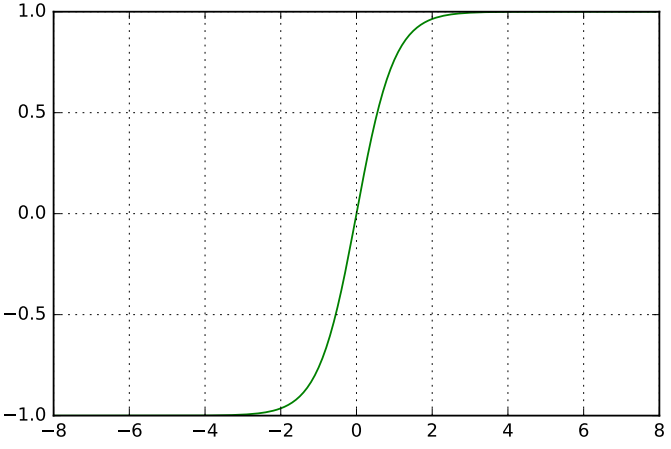
Logistic sigmoid activation function:

$$g_{logistic}(z) = \frac{1}{1 + \exp(-z)}$$



Logistic hyperbolic tangent activation function:

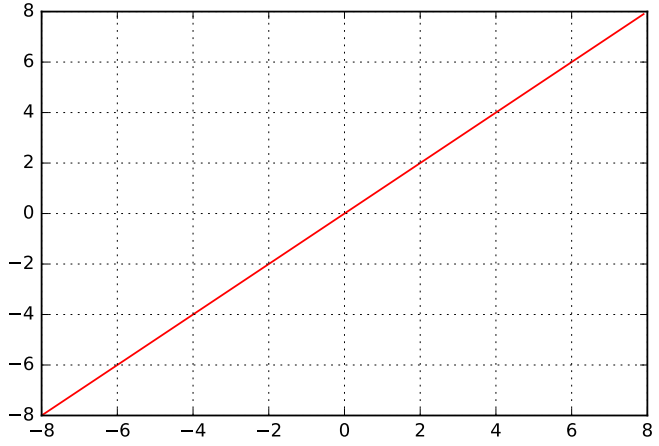
$$g_{tanh}(z) = \tanh(z)$$
$$= \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$



Activation Functions - Examples (cont.)

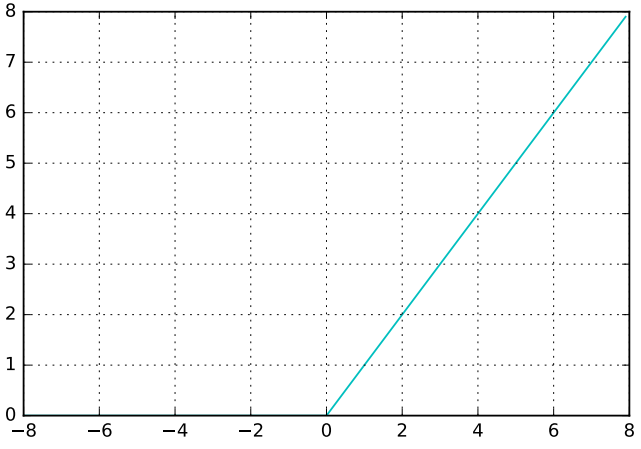
Linear activation function:

$$g_{linear}(z) = z$$



Rectified Linear (ReLU) activation function:

$$g_{relu}(z) = \max(0, z)$$



Deep learning

Definition

Deep learning is the use of multilayer perceptrons in learning tasks.

For example, supervised learning, i.e., empirical risk minimization:

- Given observations $(x_1, y_1), \ldots, (x_n, y_n)$,
- Find a multilayer perceptron f such that $f(x_i) \approx y_i$.

Questions to Answer for Yourself / Discuss with Friends

- Repetition: What is a multi-layer perceptron?
- Application of what you just learned:
  What class of functions is represented by multi-layer perceptrons with linear, polynomial, or ReLu activation functions?
- Transfer: How do multi-layer perceptrons differ from spline or finite element discretizations?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 2

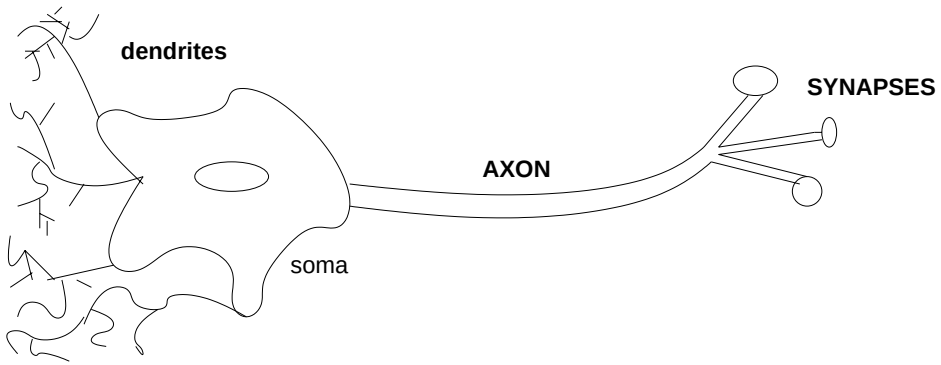
A Brief History of Deep Learning

Philipp Harms Lars Niemann

University of Freiburg

Biological Inspiration of Artificial Neural Networks

- Dendrites input information to the cell
- Neuron fires (has action potential) if a certain threshold for the voltage is exceeded
- Output of information by axon
- The axon is connected to dentrites of other cells via synapses
- Learning: adaptation of the synapse's efficiency, its synaptical weight



History of Deep Learning

Deep Learning has developed in several waves

The early days, under the name of artificial neural networks/cybernetics

- 1942 Artificial neurons as a model of brain function [McCulloch/Pitts]
- 1949 Hebbian learning [Hebb]
- 1958 Rosenblatt perceptron [Rosenblatt]
- 1960 Adaline $\rightarrow$ stochastic gradient descent [Widrow/Hoff]

The first time the popularity of NNs declined

- Negative result: linear models cannot represent the XOR function
- Backlash against biologically inspired learning [Minsky/Papert, 1969]

History of Deep Learning

1980 - early 2000s (under the name of connectionism)

- 1980 Neocognitron [Fukushima]
- 1986 Multilayer Perceptrons and backpropagation [Rumelhart et al.]
- 1989 Autoencoders [Baldi and Hornik],
  Convolutional neural networks [LeCun]
- 1997 LSTMs [Hochreiter and Schmidhuber]

The second time the popularity of NNs declined

- Ventures based on NNs made unrealistically ambitious claims
  - AI research could not fulfill these unreasonable expectations
- Other fields of machine learning made advances
  - E.g., SVMs and graphical models
  - SVMs were the state of the art on many datasets (data was small), specialized ConvNets held state of the art on MNIST but didn't scale

History of Deep Learning and ANNs (cont.)

Mid 2000s, the field got re-invigorated:

- Greedy layer-wise pretraining [Hinton, 2006]
  - It was now possible to train much deeper networks
- Several groups "resurrected" the idea of training large neural networks supervisedly using large amounts of data.
  - Most prominently [Krizhevsky et al., 2012] improved results on Imagenet benchmark by large margin
- Since then: exponential growth
  - NeurIPS attendance has grown exponentially
    - In 2018, it sold out in 12 minutes; lottery system since then
  - Some people are raising unrealistic expectations
  - Let's see how long this current wave persists

Questions to Answer for Yourself / Discuss with Friends

- Discussion: How long will the current deep learning wave persist?
  - What are reasons that it will continue?
  - What are reasons that it will end?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 3

Deep Learning as Representation Learning

Philipp Harms Lars Niemann

University of Freiburg

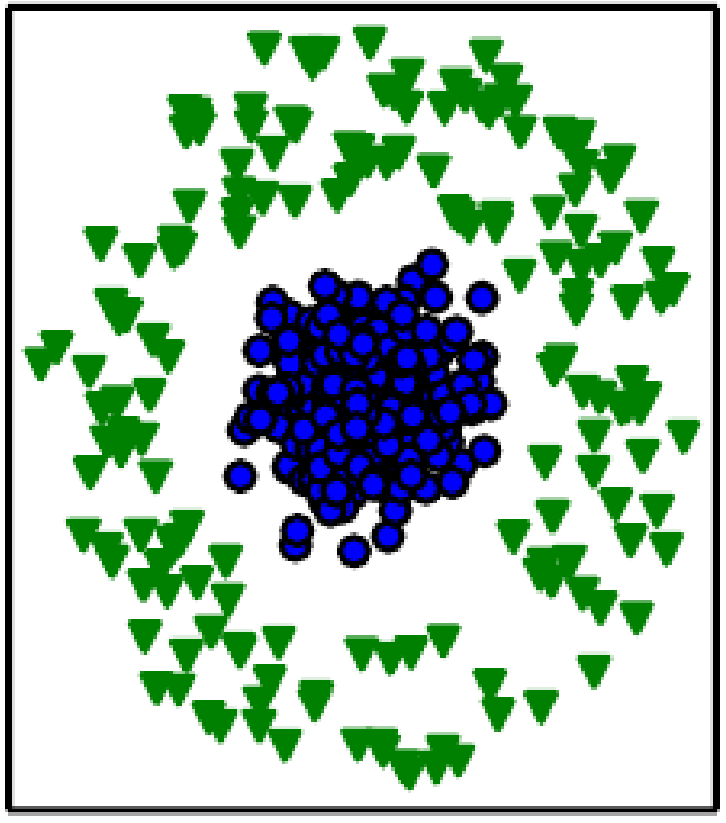
Some terminology

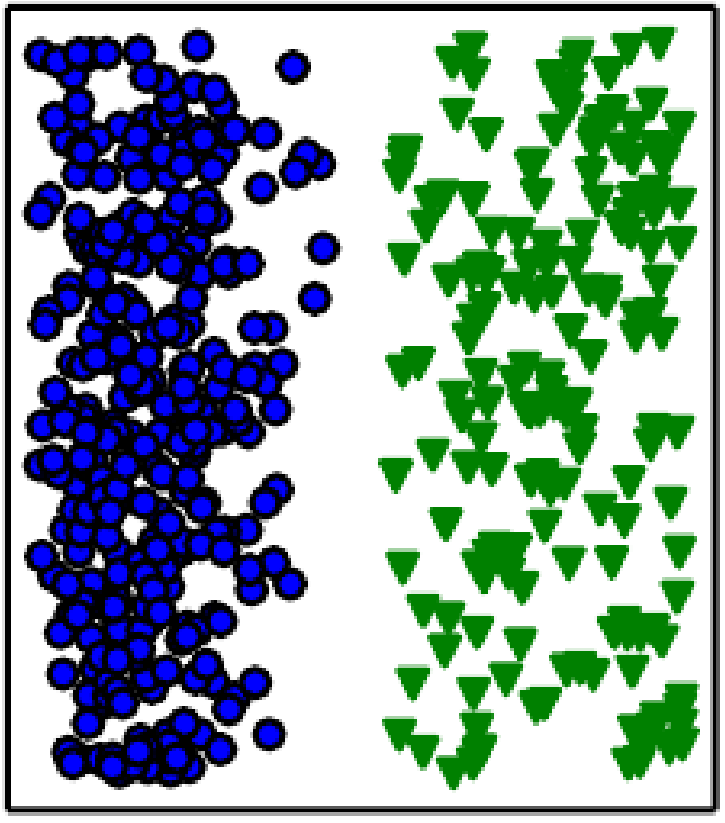
- Supervised learning: given data (x_i, y_i), find a function f such that $f(x_i) \approx y_i$
- Classification: special case where f is an indicator function (aka. classifier) and y_i belong to $\{0, 1\}$
- Data representation: a coordinate system for x
- Feature: a coordinate
- Linearly separable: y_i equals the sign of a linear functional of x_i

Definition: Representation learning

Representation learning

"a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification" - LeCun et al., 2015





Example for a Poor Representation: Roman Numbers

In particular, poor for the task of addition.
E.g., perform CCCLXIX + DCCCXLV (369 + 845)

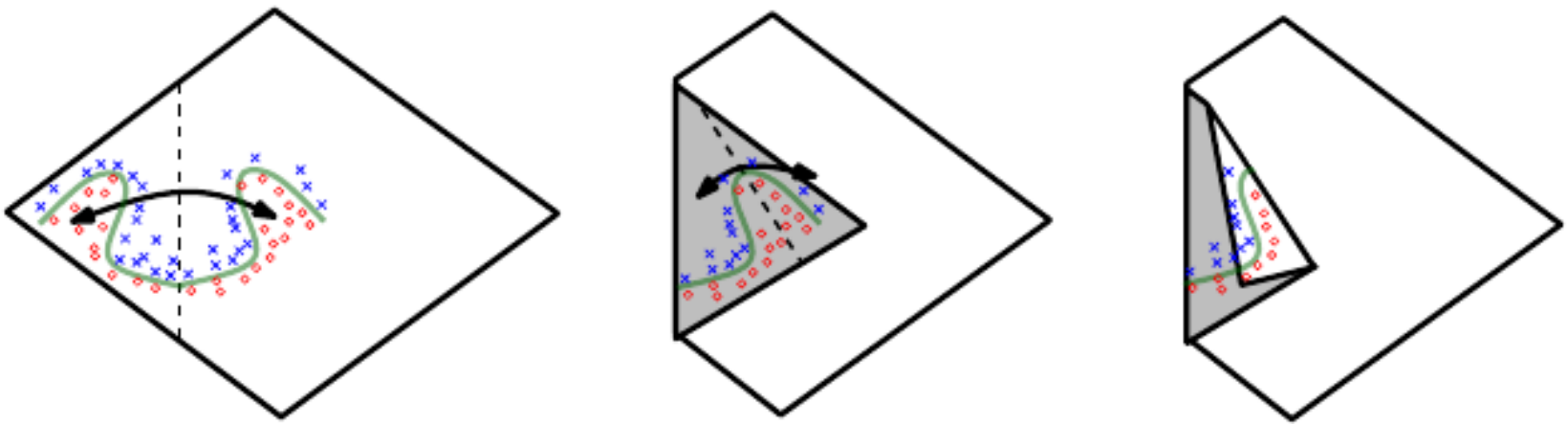
1. Substitute for any subtractives : CCCLXVIIII + DCCCXXXXV
2. Concatenate: CCCLXVIIIIDCCCXXXXV
3. Sort : DCCCCCCLXXXXXVVIIII
4. Combine groups to obtain:
   DCCCCCCLXXXXXXIIII
   DCCCCCCLLXIIII
   DCCCCCCCXIIII
   DDCCXIIII
   MCCXIIII
5. Re-Substitute any subtractives:
   MCCXIV

In contrast, converting to our current number system: $369 + 845 = 1214$.

Definition: Deep learning

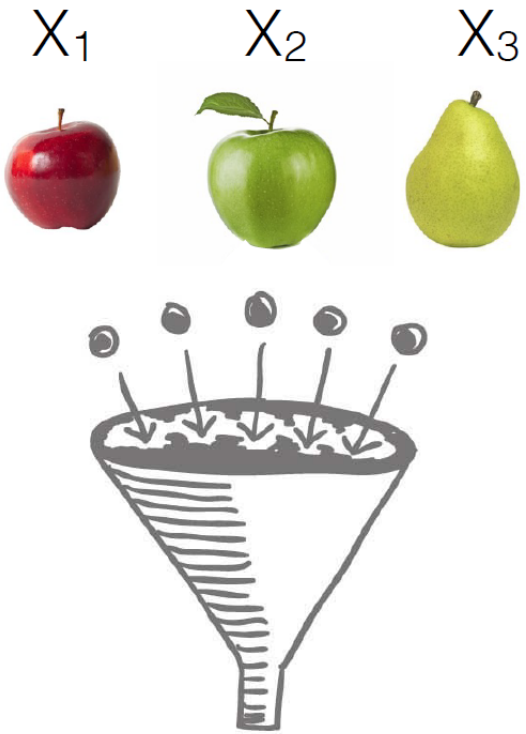
Deep learning

"representation learning methods with multiple levels of representation, obtained by composing simple but nonlinear modules that each transform the representation at one level into a [...] higher, slightly more abstract (one)" - LeCun et al., 2015

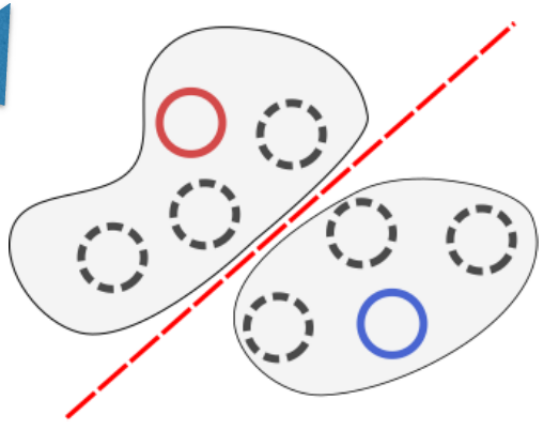


Standard Machine Learning Pipeline

- Standard machine learning algorithms are based on high-level attributes or features of the data
- They require (often substantial) feature engineering, i.e., extraction and selection of features.

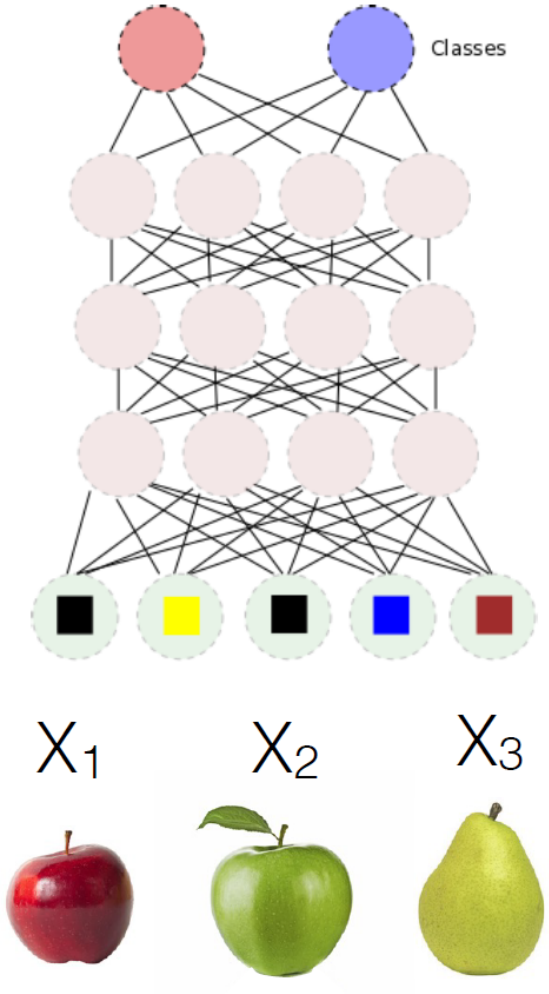


	Merkmale
X_1	rot, 3.5 cm
X_2	grün, 4 cm
X_3	grün, 10 cm

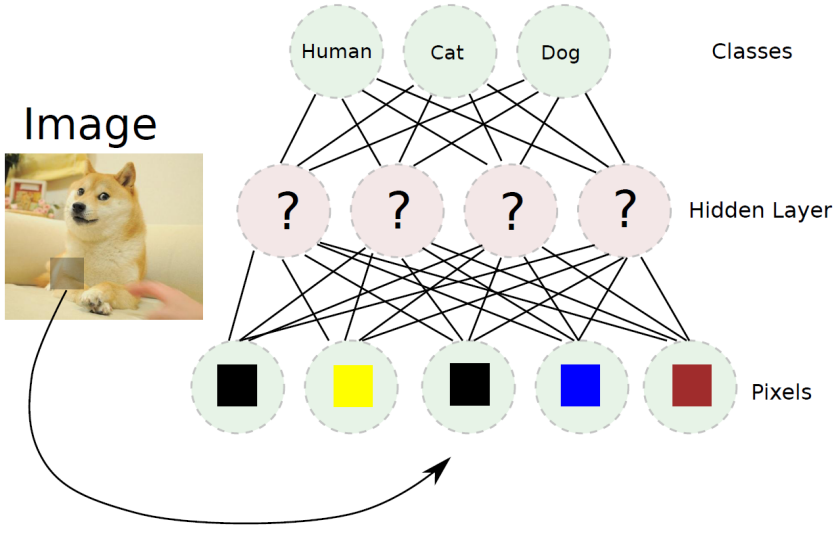


Representation Learning Pipeline

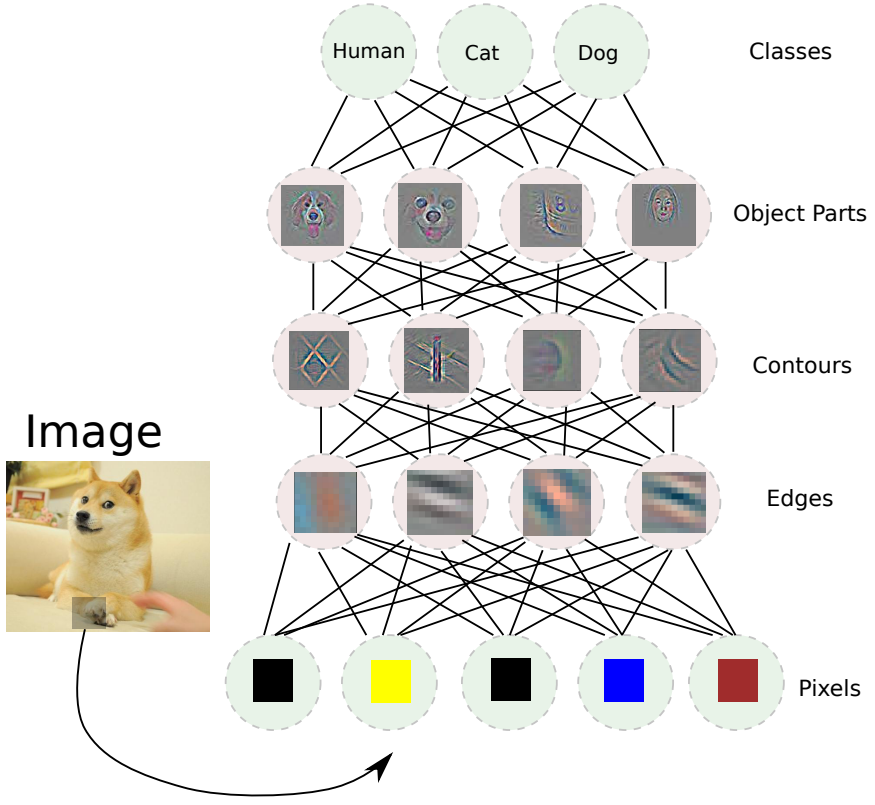
- Jointly learn features and classifier, directly from raw data
- This is also referrred to as end-to-end learning



Shallow vs. Deep Learning



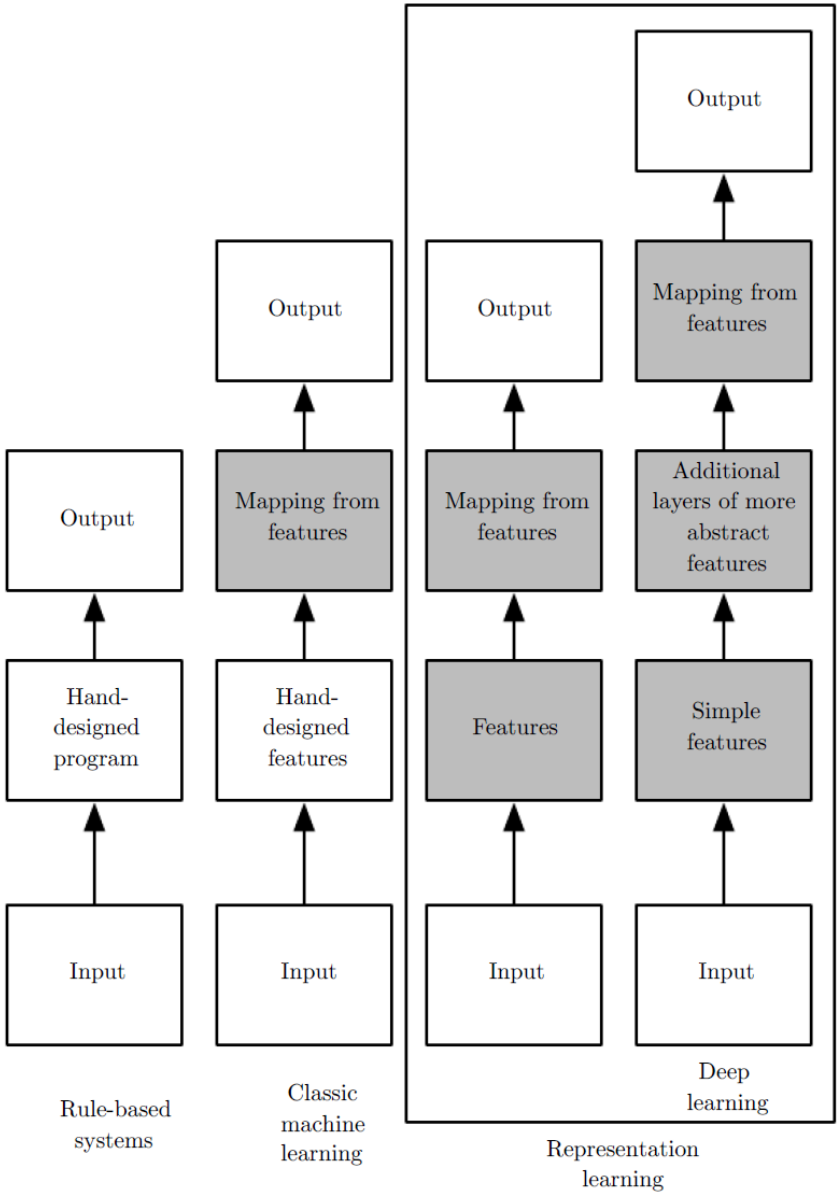
Shallow vs. Deep Learning



[Visualizations of network activations taken from Zeiler [2014]]

- Deep Learning: learning a hierarchy of representations that build on each other, from simple to complex
- Features are learned in an end-to-end fashion, from raw data

Relation to More Traditional Learning Approaches



Questions to Answer for Yourself / Discuss with Friends

- Relation to your interests:
  What would be a good and a bad representation for a problem you find interesting?

- Discussion: Are deep networks always better than shallow ones?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 4

Definition of Neural Networks

Philipp Harms Lars Niemann

University of Freiburg

Neural Networks: Definition

Definition

Let $d, L \in \mathbb{N}$. A neural network with input dimension d and L layers is a sequence of matrix-vector tuples

$$\Phi = \big((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)\big),$$

where $N_0 := d$, $N_1, \ldots, N_L \in \mathbb{N}$, $A_l \in \mathbb{R}^{N_{l-1} \times N_l}$, and $b_l \in \mathbb{R}^{N_l}$ for $l \in \{1, \ldots, L\}$.

- According to this definition, neural networks are the coefficients of multi-layer perceptrons.
- This distinction is useful but not always made in the literature.

Neural Networks: Definition (cont.)

Definition

The realization of a neural network Φ with activation function $\rho\colon \mathbb{R} \to \mathbb{R}$ is the function

$$\mathrm{R}(\Phi)\colon \mathbb{R}^d \to \mathbb{R}^{N_L}, \quad \mathrm{R}(\Phi)(x) \coloneqq x_L,$$

where the output x_L results from

$$\begin{aligned} x_0 &\coloneqq x, \\ x_l &= \rho(A_l x_{l-1} + b_l) \text{ for } l \in \{1, \dots, L-1\}, \\ x_L &\coloneqq A_L x_{L-1} + b_L. \end{aligned}$$

Here ρ is understood to act component-wise.

- Thus, a multilayer perceptron is the realisation of a neural network.

Neural Networks: Definition (cont.)

Definition

We call $\mathrm{N}(\Phi) := d + \sum_{l=1}^{L} N_l$ the number of neurons, $\mathrm{L}(\Phi) := L$ the number of layers or depth, and

$$\mathrm{M}(\Phi) := \sum_{l=1}^{L} \mathrm{M}_l := \sum_{l=1}^{L} \|A_l\|_0 + \|b_l\|_0$$

the number of weights. Here $\|\cdot\|_0$ denotes the number of non-zero entries of a matrix or vector.

Neural Networks: Definition (cont.)

Definition

Let $L \in \mathbb{N}$. A vector $S = (N_0, \ldots, N_L) \in \mathbb{N}^{L+1}$ is called architecture of a neural network

$$\Phi = ((A_1, b_1), \ldots, (A_L, b_L))$$

if $A_l \in \mathbb{R}^{N_{l-1} \times N_l}$ for $l = 1, \ldots, L$. Given such a vector S, we denote by $\mathcal{NN}(S)$ the set of all neural networks with architecture S.

Note: $\mathcal{NN}(S)$ is a finite-dimensional linear space.

Questions to Answer for Yourself / Discuss with Friends

- Check: Is $\|\cdot\|_0$ a norm?
- Repetition: What are neural networks, and how do they differ from multi-layer perceptrons?
- Discussion: Is the realization map continuous in some sense?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 5

Operations on Neural Networks

Philipp Harms Lars Niemann

University of Freiburg

Operations on neural networks

Lemma (Operations)

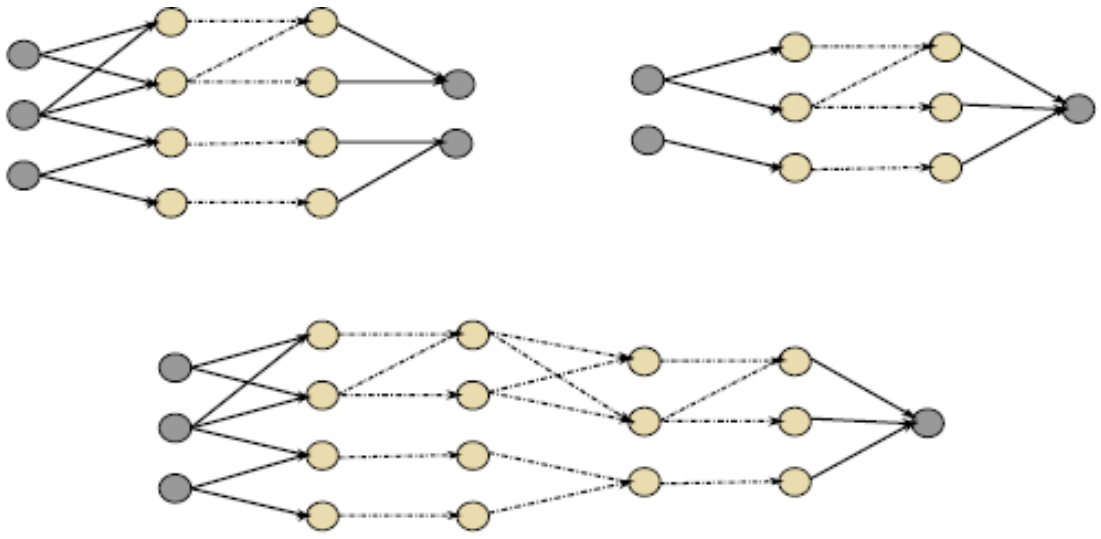
Let Φ^1 and Φ^2 be two neural networks, and let Δ denote the diagonal map $x \mapsto (x, x)$.

- *If the composition $\mathrm{R}(\Phi^1) \circ \mathrm{R}(\Phi^2)$ is well-defined, it can be represented as the realization of a neural network $\Phi^1 \bullet \Phi^2$.*
- *The full parallelization $\big(\mathrm{R}(\Phi^1), \mathrm{R}(\Phi^2)\big)$ can be represented as the realization of a neural network $\mathrm{FP}(\Phi^1, \Phi^2)$.*
- *If the parallelization $\big(\mathrm{R}(\Phi^1), \mathrm{R}(\Phi^2)\big) \circ \Delta$ is well-defined, it can be represented as the realization of a neural network $\mathrm{P}(\Phi^1, \Phi^2)$.*
- *The number of nodes satisfy $\mathrm{M}(\mathrm{P}(\Phi^1, \Phi^2)) = \mathrm{M}(\mathrm{FP}(\Phi^1, \Phi^2)) = \mathrm{M}(\Phi^1) + \mathrm{M}(\Phi^2)$.*

Proof. The networks defined next have the desired properties. □

Concatenation: Intuition

Composition of functions corresponds to concatenation of neural networks:



Concatenation [Figure from Petersen]

Concatenation: Definition

Definition (Concatenation)

Let $L_1, L_2 \in \mathbb{N}$ and let

$$\Phi^1 = \big((A_1^1, b_1^1), \dots, (A_{L_1}^1, b_{L_1}^1)\big)$$

$$\Phi^2 = \big((A_1^2, b_1^2), \dots, (A_{L_2}^2, b_{L_2}^2)\big)$$

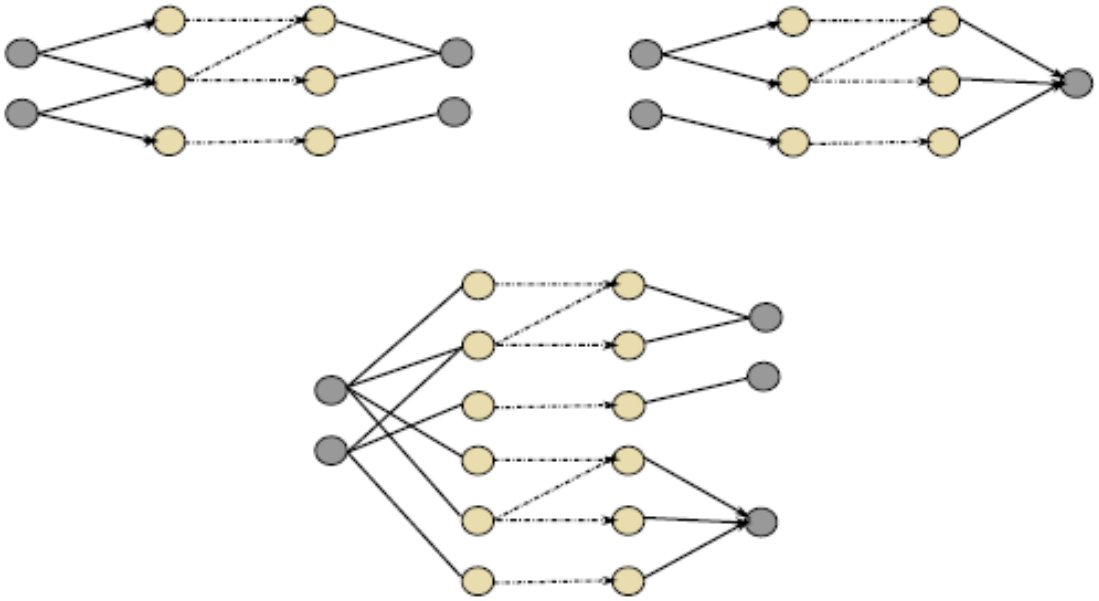
be two neural networks such that the input layer of Φ^1 has the same dimension as the output layer of Φ^2.

Then the concatenation of Φ^1 and Φ^2 is the neural network $\Phi^1 \bullet \Phi^2$ with $L_1 + L_2 - 1$ layers given by

$$\Phi^1 \bullet \Phi^2 := \big((A_1^2, b_1^2), \dots, (A_{L_2-1}^2, b_{L_2-1}^2), (A_1^1 A_{L_2}^2, A_1^1 b_{L_2}^2 + b_1^1), (A_2^1, b_2^1), \dots, (A_{L_1}^1, b_{L_1}^1)\big).$$

Parallelisation: Intuition

- The parallelization $\mathrm{P}(\Phi^1, \Phi^2)$ is a neural network with input dimension $d_1 = d_2$, where the inputs are shared.
- The full parallelization $\mathrm{FP}(\Phi^1, \Phi^2)$ is a neural network with input dimension $d_1 + d_2$, where the inputs are not shared.



Parallelisation with shared inputs [Figure from Petersen]

Parallelisation: Definition

Definition

Let Φ^1 and Φ^2 be two neural networks with the same number L of layers and input dimensions d_1 and d_2, respectively:

$$\Phi^1 = \left((A_l^1, b_l^1)\right)_{l\in\{1,\dots,L\}}, \quad \Phi^2 = \left((A_l^2, b_l^2)\right)_{l\in\{1,\dots,L\}}.$$

Then the parallelization and full parallelization of Φ^1 and Φ^2 are the neural networks

$$\mathrm{P}(\Phi^1, \Phi^2) := \Big((\hat{A}_1, \hat{b}_1), (\tilde{A}_2, \tilde{b}_2), \dots, (\tilde{A}_L, \tilde{b}_L)\Big) \quad \text{if } d_1 = d_2,$$

$$\mathrm{FP}(\Phi^1, \Phi^2) := \Big((\tilde{A}_1, \tilde{b}_1), (\tilde{A}_2, \tilde{b}_2), \dots, (\tilde{A}_L, \tilde{b}_L)\Big) \quad \text{for arbitrary } d_1, d_2,$$

where for each $l \in \{1, \dots, L\}$,

$$\hat{A}_l := \begin{pmatrix} A_l^1 \\ A_l^2 \end{pmatrix}, \quad \hat{b}_l := \begin{pmatrix} b_l^1 \\ b_l^2 \end{pmatrix}, \quad \tilde{A}_l := \begin{pmatrix} A_l^1 & 0 \\ 0 & A_l^2 \end{pmatrix}, \quad \tilde{b}_l := \begin{pmatrix} b_1^1 \\ b_1^2 \end{pmatrix}.$$

Questions to Answer for Yourself / Discuss with Friends

- Repetition: Take a pen and paper and verify that the network concatenations and parallelizations satisfy the properties claimed in the lemma.
- Check: Can multiplication of functions be represented as an operation on neural networks?
- Discussion: Can you think of any further operations on neural networks?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 6

Universality of Neural Networks

Philipp Harms Lars Niemann

University of Freiburg

Universality

Definition

Let $d, L \in \mathbb{N}$, and let $\rho\colon \mathbb{R} \to \mathbb{R}$ be a continuous activation function. For $K \subseteq \mathbb{R}^d$ compact, denote by $\mathrm{MLP}(\rho, d, L; K)$ the set of multilayer perceptrons with input dimension d, L layers and output dimension 1, restricted to K.

We say that $\mathrm{MLP}(\rho, d, L; K)$ is universal if it is dense in $C(K)$, the space of real-valued continuous functions on K with the supremum norm.

Universal approximation theorem

Definition (Discriminatory activation functions)

Let $d \in \mathbb{N}, K \subseteq \mathbb{R}^d$ compact. A continuous function $\rho : \mathbb{R} \to \mathbb{R}$ is called discriminatory (on K) if the only signed Radon measure μ on K with

$$\int_K \rho(ax - b) d\mu(x) = 0 \quad (a \in \mathbb{R}^d, b \in \mathbb{R})$$

is the zero measure $\mu = 0$.

Theorem (Universal approximation theorem of Cybenko)

Let $d \in \mathbb{N}, K \subseteq \mathbb{R}^d$ compact, and $\rho : \mathbb{R} \to \mathbb{R}$ discriminatory. Then $\mathrm{MLP}(\rho, d, 2; K)$ is universal.

Tool: Riesz–Markov–Kakutani representation

Notation

- Let K be a compact Hausdorff topological space.
- Denote by $C(K)$ the Banach space of real-valued continuous functions on K with the supremum norm.
- Denote by $\mathcal{M}(K)$ the Banach space of finite signed Radon measures on K with the total variation norm.
- Recall that a Borel measure is called Radon if it is regular and locally finite.

Theorem (Riesz–Markov–Kakutani representation)

On any compact Hausdorff topological space K, the topological dual of $C(K)$ is $\mathcal{M}(K)$.

Tool: Hahn–Banach extension

Theorem (Hahn–Banach extension)

If $\mathcal{X}$ is a normed space, M a linear subspace, and λ a continuous linear functional on M, then λ can be extended to a functional $\Lambda\colon \mathcal{X} \to \mathbb{R}$ such that $\|\lambda\| = \|\Lambda\|$.

Consequently, M is dense if and only if every continuous linear functional on $\mathcal{X}$ that vanishes on M is trivial.

Proof of the universal approximation theorem

- Note that $\mathrm{MLP}(\rho, d, 2; K) \subseteq C(K)$ is a linear subspace
- Assume for contradiction that $\mathrm{MLP}(\rho, d, 2; K)$ is not dense
- By Hahn-Banach, there is a non-zero measure μ with

$$\int_K f d\mu = 0 \quad (f \in \mathrm{MLP}(\rho, d, 2; K))$$

- However, the functions $f_{a,b}(x) := \rho(ax - b)$ belong to $\mathrm{MLP}(\rho, d, 2; K)$ for all $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$.
- As ρ is discriminatory, this gives the desired contradiction $\square$

Questions to Answer for Yourself / Discuss with Friends

- Repetition: Recount the universal approximation theorem and its proof.
- Check: Verify that one has indeed $K \ni x \mapsto \rho(ax - b) \in \mathrm{MLP}(\rho, d, 2; K)$ for $a \in \mathbb{R}^d, b \in \mathbb{R}$
- Transfer: How does Cybenko's universality theorem differ from the Stone–Weierstrass approximation theorem?

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 7

Discriminatory Activation Functions

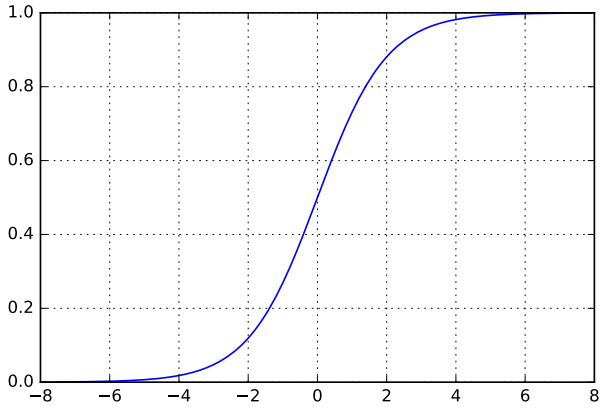
Philipp Harms Lars Niemann

University of Freiburg

Sigmoidal functions

Definition

A continuous function $\rho : \mathbb{R} \to \mathbb{R}$ is called sigmoidal, if $\rho(x) \to 1$ for $x \to \infty$ and $\rho(x) \to 0$ for $x \to -\infty$.



Example: The logistic (aka. sigmoidal) function $x \mapsto (1 + e^{-x})^{-1}$ is sigmoidal

Theorem (Cybenko)

Let $d \in \mathbb{N}, K \subseteq \mathbb{R}^d$ compact. Then every sigmoidal function $\rho\colon \mathbb{R} \to \mathbb{R}$ is discriminatory on K.

Proof that sigmoidal functions are discriminatory

- Let $\mu \in \mathcal{M}(K)$ such that $\int_K \rho(ax-b)d\mu(x) = 0$ for $a \in \mathbb{R}^d, b \in \mathbb{R}$
- For any $\theta \in \mathbb{R}$,

$$\lim_{\lambda \to \infty} \rho(\lambda(ax-b)+\theta) = \begin{cases} 1 & ax-b > 0 \\ \rho(\theta) & ax-b = 0 \\ 0 & ax-b < 0 \end{cases}$$

- Thus, by dominated convergence,

$$\mu(\{ax > b\}) + \rho(\theta)\mu(\{ax = b\}) = \lim_{\lambda \to \infty} \int_K \rho(\lambda(ax-b)+\theta)d\mu(x) = 0$$

- Taking the limit $\theta \to -\infty$, we conclude that

$$\mu(\{ax > b\}) = 0 \quad (a \in \mathbb{R}^d, b \in \mathbb{R})$$

Proof that sigmoidal functions are discriminatory (cont.)

- In particular, for any $b_1 < b_2$,

$$\mu(\{ax > b_1\}) - \mu(\{ax > b_2\}) = \int_K \mathbb{1}_{(b_1,b_2]}(ax)d\mu(x) = 0$$

- This extends first by linearity to step functions and then by density to continuous bounded functions:

$$\int_K g(ax)d\mu(x) = 0 \quad (g \in C_b(\mathbb{R}))$$

- By choosing $g = \sin$ and $g = \cos$, we arrive at

$$0 = \int_K \exp(iax)d\mu(x) \quad (a \in \mathbb{R}^d)$$

- This means the Fourier transform of μ vanishes; whence $\mu = 0$. □

Extensions and variations

- The above proof also works for other dual pairings such as e.g. $L^1(\mathbb{R}^d)$ and $L^\infty(\mathbb{R}^d)$.
- Alternatively, for activation functions $\rho \in \{\sin, \cos, \exp\}$, density of $\{\rho(a \cdot + b); a \in \mathbb{R}^d, b \in \mathbb{R}\}$ in $C(K)$ follows directly from Stone–Weierstrass.
- Alternatively, for activation functions ρ with $\int \rho(x)dx \neq 0$, density in $L^1(K)$ can be shown using the Tauberian theorem of Wiener: any translation-invariant subspace of $L^1(\mathbb{R})$, which contains for any $\xi \in \mathbb{R}$ a function f with $\hat{f}(\xi) \neq 0$, is dense. [Cybenko]

Questions to Answer for Yourself / Discuss with Friends

- Check: Are sigmoidal functions bounded?
- Background: Do you recall the proof of the injectivity of the Fourier transform on measures? (Hint: Stone–Weierstrass for trigonometric polynomials.)

Mathematics of Deep Learning, Summer Term 2020

Week 2, Video 8

Wrapup

Philipp Harms Lars Niemann

University of Freiburg

Outlook on this week's discussion and reading session

- Reading:
  - Hornik (1989): Multilayer Feedforward Networks are Universal Approximators
  - Cybenko (1989): Approximation by superpositions of a sigmoidal function
  - Hornik (1991): Approximation capabilities of multilayer feedforward networks

Summary by learning goals

Having heard this lecture, you can now . . .

- Describe the structure of multi-layer perceptrons and neural networks
- Sketch a brief history of deep learning and put it into the perspective of representation learning.
- State the universal approximation theorem and understand its elegant proof