

武汉纺织大学计算机与人工智能学院

# 深度学习基础

## 2. 神经网络的数学基础

吴晓堃

xkun.wu at gmail dot com

2021/03/15

## Outline

## 本章内容

张量。张量运算。基于梯度的优化。反向传播算法与链式求导。实践：二维仿射变换，基本激活函数，ReLU 合成法构造一般函数。

**重点：**张量、张量运算、三种基本激活函数、深度学习层间运算的一般形式；

**难点：**张量运算的几何解释、基于梯度的优化、反向传播算法、ReLU 合成法、一致逼近理论。

3

## 学习目标

4 / 14

- 理解张量的概念；
- 掌握张量运算，并理解其几何解释；
- 理解基于梯度的优化及随机梯度下降算法；
- 理解反向传播算法的理论基础：链式求导法；
- 掌握三种基本激活函数：ReLU，Sigmoid，Tanh；
- 了解 ReLU 合成法构造连续函数的方法。

4

## 概念

### 张量

数据的容器。0D、1D、2D 张量又分别成为标量、向量、矩阵。

### 张量运算

数据不同表示之间的变换函数。

### 深度学习层间运算的一般形式

$$\underline{output} = \text{activate}(\underline{GT})$$

$$\underline{GT} = \mathbf{W} * \underline{input} + \mathbf{b}$$

5

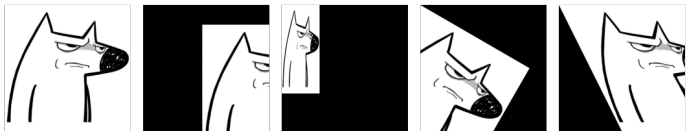
## 二维仿射变换

### 二维仿射变换的一般形式

$$\underline{GT} = \mathbf{W} * \underline{input} + \mathbf{b}$$

### 基本二维仿射变换矩阵

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & e_x & 0 \\ e_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



6

## 深度学习的几何解释

### 如何想象高维空间？

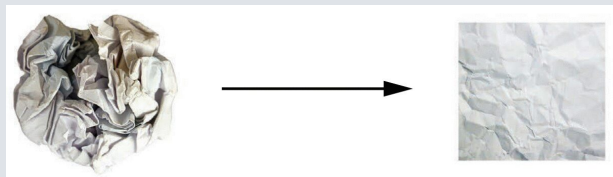
首先研究低维空间，归纳出规律，然后将规律泛化到高维。

7

## 深度学习的几何解释

### 深度学习可以解释为高维空间中非常复杂的几何变换

- 一切数据都是张量，即几何空间中的点。
- 模型的每一层对数据点做一个几何变换；而模型本身是这些变换的合成。
- 注意：几何变换必须可微，意味着从输入到输出的几何变形必须平滑且连续。



### 模型的可探索空间（假设空间）需要足够大

只要模型的参数足够多，就能捕捉到原始数据中所有的映射关系。想象“ $\Omega$  路径”。

8

# 深度学习与神经网络

## 深度学习的核心在于连续的几何空间操作

- 事物之间的映射关系在几何空间中有自然的度量函数：距离。
- 并且，从计算的角度来看，处理向量空间很高效。
- 注意：大脑是否通过几何空间来实现认知，则是另一个问题。

因此也可称为：分层表示学习、层级表示学习、深度可微模型、链式几何变换。

## 神经网络名称纯粹是出于历史原因

- 神经网络最初来自于使用图对知识进行编码。
- 但它与神经或网络都没有关系，尤其是和大脑几乎没有任何关系。

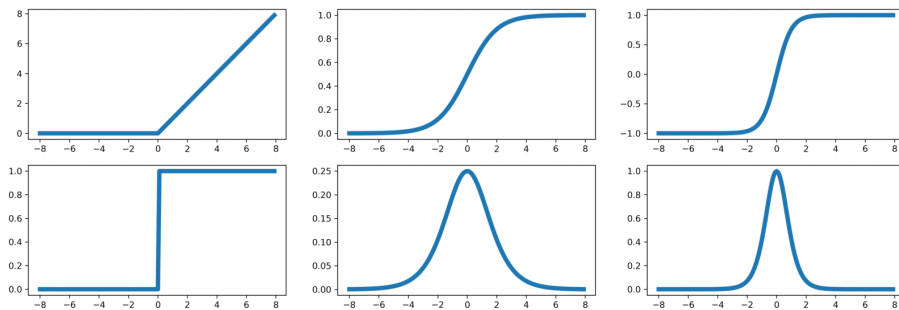
9

10 / 14

# 激活函数

## 三种基本激活函数

ReLU, Sigmoid, Tanh。



## 激活函数的作用

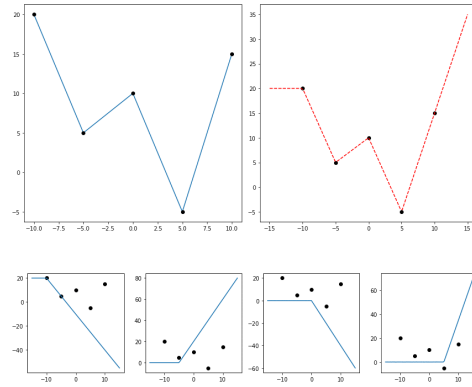
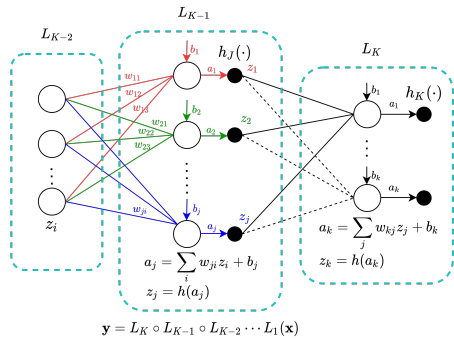
提供非线性。

10

# ReLU 合成法

## ReLU 合成的一般形式

$$\sum \text{relu}(\mathbf{W} * \text{input} + \mathbf{b})$$



11

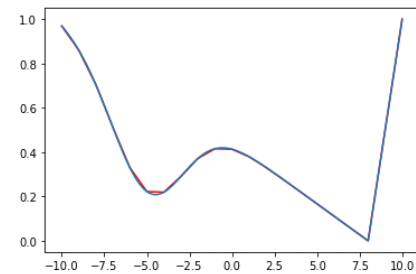
## 一致逼近原理

## Universal approximation theorem (非常有用的废话)

In approximation theory, both **shallow** and **deep** networks are known to **approximate any continuous functions** at an **exponential cost**.

构造法:

- 构造线性函数;
- 构造分段线性函数;
- 构造离散化的任意函数。



12

Objective:  $\arg \min_{W,b} J$ ,  
with:  $J = \|\bar{y} - y\|, \bar{y} = \sum \text{relu}(\mathbf{W} * \mathbf{x} + \mathbf{b})$

$$W_1 = W_0 - \nabla J * s$$

- 
- A graph illustrating the cost function  $J(W)$  versus weights  $W$ . The curve is U-shaped, representing the cost landscape. The minimum point is labeled "Global minimum". A series of points connected by arrows shows the path of an optimization algorithm starting from "Initial weights" and moving towards the global minimum.

13

## 反向传播算法

将链式法则应用于神经网络梯度值的计算，得到的算法叫作反向传播算法。

