

第8讲 神经网络

主讲教师：欧新宇

February 21, 2020

AlphaGo、无人驾驶、人脸识别、智能翻译.....这一个个耳熟能详的名词无不预示着人工智能时代的到来。人工智能的本质，是对人的思维过程和行为方式的模拟。研究人的认知机理，引入人的神经元概念，势在必行。

**人工神经网络，可以说是这一切的起源。
或者说是这一切起源的核心。**

- 神经网络网络的前世今生
 - 神经网络的起源
 - 神经网络的两次寒冬
 - 神经网络的发展简史
- 神经网络的原理
- 激活函数及非线性矫正
- Baseline模型及参数调整
- MNIST手写字体识别

神经网络的起源

人工智能领域的一个重要任务是让计算机能够像人一样对输入的信息进行判定。

- 当计算机读入一幅图像后，能否判定里面有没有苹果，如果有，苹果在图中的哪个位置；
- 当计算机读入一段语音后，能否判定里面有没有提到"中国"二字，如果有，在什么时间点。

神经网络的起源

众所周知，人对世界的感知和理解主要通过数以亿计的神经元来完成，神经元之间彼此连接构成巨大的神经元网络，输入的信号（如视网膜上的神经元感受到的光线等）经过一层层的神经元往脑部传递，不断做出决策，再通过一层层的神经元输出到反馈端（如影响手脚部动作等）

神经网络的两次寒冬(1/3)

- 1943年，逻辑学家Walter Pitts和神经生理学家Warren McCulloch联合发表文章，首次将神经元概念引入计算领域，提出了第一个人工神经元模型，开启了神经网络的大门。
- 1957年，知名学者Frank Rosenblatt提出了感知器(Perceptron)的概念，该概念非常接近神经元的实际机理，通过将多层感知器前后连接，可以构成一个决策网络，从而为神经网络的研究奠定了基石。
- 1969年，被誉为人工智能之父的Marvin Minsky和Seymour Papert出版《Perceptron》一书探讨感知器的优劣，认为仅靠局部连接的神经网络无法有效地开展训练，而全连接的神经网络则过于复杂而不实用。更重要的是，限于当时的计算方法与能力，复杂神经网络核心权重参数的计算时长将无法忍受。这些情况影响了学界和业界对神经网络的信心，神经网络的研究陷入了第一次低谷期。

神经网络的两次寒冬(2/3)

近二十年后，当代神经网络三巨头（Geoffrey Hinton、Yoshua Bengio、Yann LeCun）相继发文，推动了神经网络研究的再次兴起。

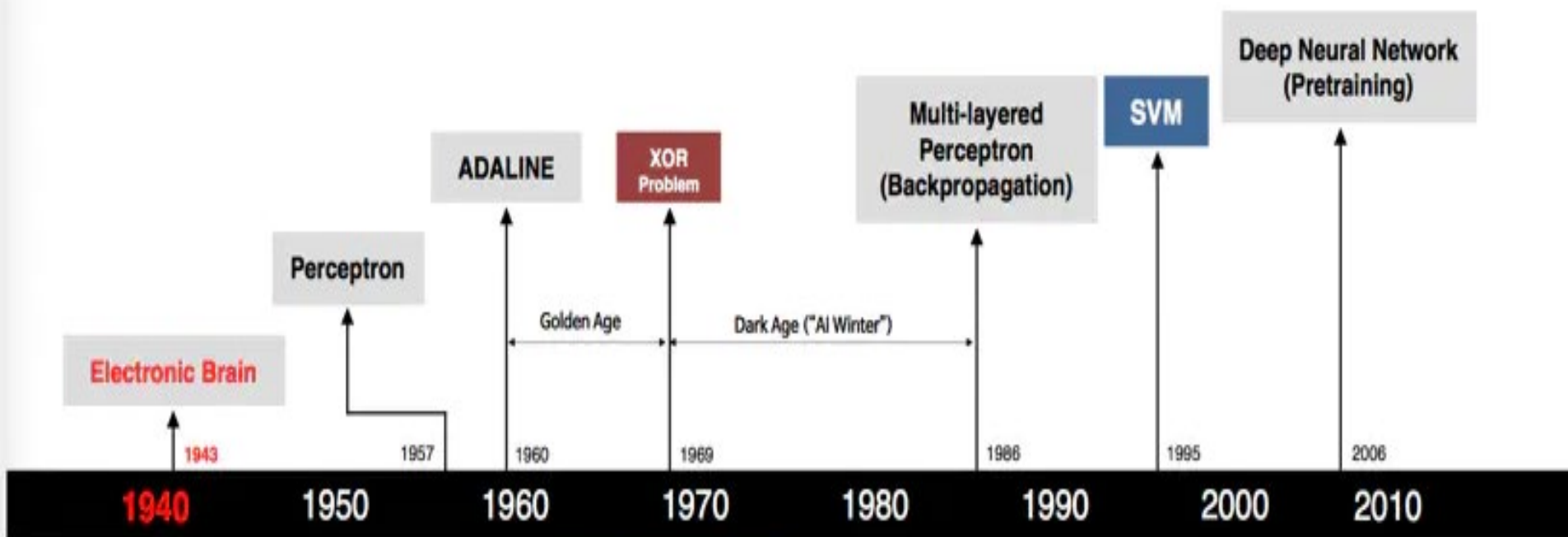
- ❁ 1986年，Geoffrey Hinton和David Rumelhart联合在Nature上发表论文，将BP算法用于神经网络模型，实现了对权重参数的快速计算。
- ❁ 1990年，Yann LeCun发表文章，采用BP神经网络实现对手写数字的识别，这可以被视作神经网络的"第一个"重大应用，直到上世纪九十年代末，超过10%的美国支票都采用该技术进行自动识别。

神经网络的两次寒冬(3/3)

1998年, Yann LeCun又发文提出了LeNet-5的框架, 即现在热火朝天的卷积神经网络 (Convolutional Neural Network) 的基本框架。然而卷积要消耗大量计算资源, BP方法又会带来梯度弥散的问题, 从而限制了神经网络的深度和效果。

相反, 俄罗斯学者Vladimir Vapnik在1963年提出的支撑向量机 (Support Vector Machine, SVM) 概念则不断深入发展。到2002年, 已将手写数字识别的错误率降至0.56%, 远高于同期神经网络的效果。神经网络的研究迎来了第二次寒冬。

神经网络的发展历史



S. McCulloch - W. Pitts



F. Rosenblatt



B. Widrow - M. Hoff



M. Minsky - S. Papert



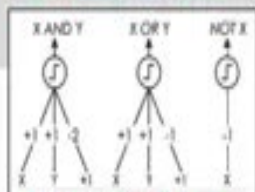
D. Rumelhart - G. Hinton - R. Williams



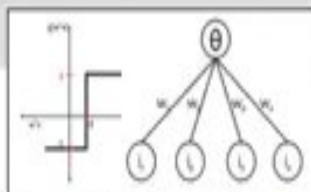
V. Vapnik - C. Cortes



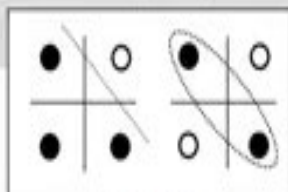
G. Hinton - S. Ruslan



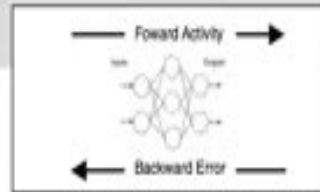
- Adjustable Weights
- Weights are not Learned



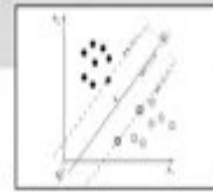
- Learnable Weights and Threshold



- XOR Problem



- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting



- Limitations of learning prior knowledge
- Kernel function; Human Intervention



- Hierarchical feature Learning

神经网络的发展历史

神经网络发展最核心的步骤：

- 感知器模型 Perceptron
- 前馈神经网络 =》 多层感知机 Multi-layered Perceptron, MLP
- BP神经网络 Back Propagation
- 深度学习
 - 卷积神经网络 Convolutional Neural Network, CNN
 - 限制玻尔兹曼机 Restricted Boltzmann machine, RBM
 - 深度自编码器 Deep Auto-Encoder, DAE

神经网络的发展历史



Andrew Ng
斯坦福大学, 百度



Yoshua Bengio
蒙特利尔大学, 微软



Michael E. Jordan
加州大学伯克利分校
美国三院院士



李飞飞
斯坦福大学
Google AI



Yann LeCun
纽约大学, Facebook



颜水成
新加坡国立, 奇虎
360



Geoffrey Hinton
多伦多大学,
Google



汤晓鸥
香港中文大学
商汤科技

神经网络的原理 —— 多层感知机MLP

● 线性模型

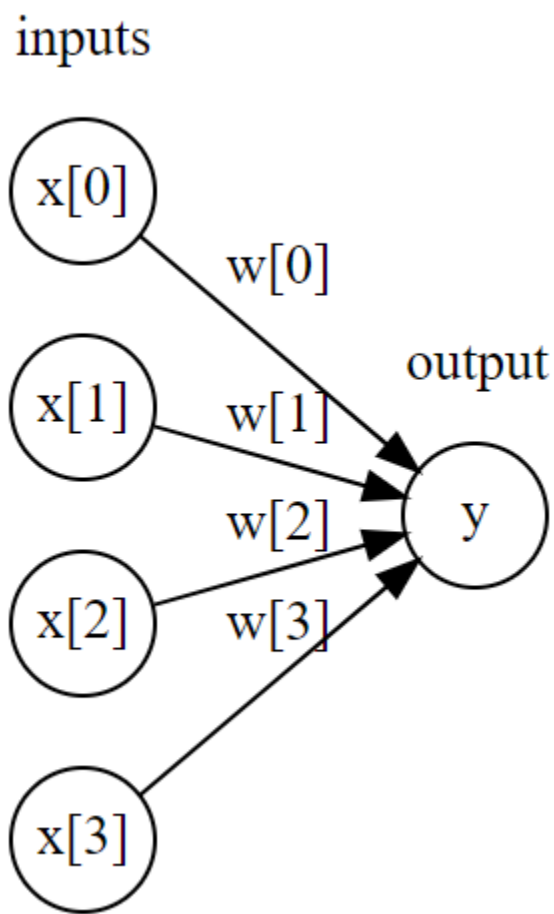
$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \dots + w[p] * x[p] + b$$

在该公式中 \hat{y} 表示对真实 y 值的估计值. $x[0], x[1], \dots, x[p]$ 是数据集中样本的**特征值**; w 和 b 是模型的参数, 其中 w 表示每个特征的**权重**, 也就是指每个神经元对于模型**贡献的重要程度**。

因此, 从数学表示表达式来看, 我们可以认为 \hat{y} 就是每个特征的**加权和**。

神经网络的原理 —— 多层感知机MLP

● 线性模型



左图给出了线性模型求解过程图示。其中 $x[i]$ 称为输入层(inputs), y 称为输出层(output)。

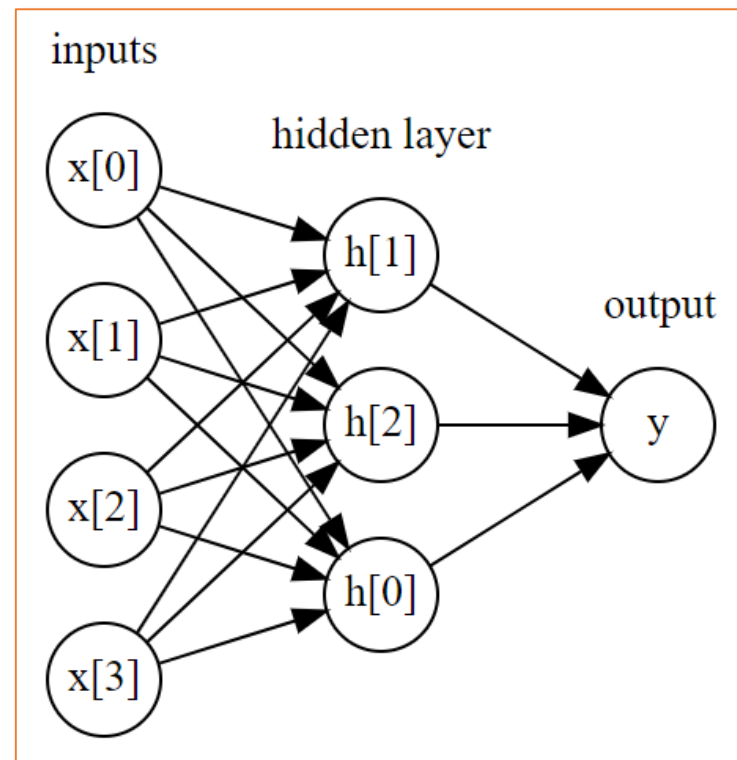
当 $x[i](i = 0, 1, 2, 3, \dots, n)$, 表示样本 x 有 $n+1$ 个特征输入, 左图表示有4个特征输入。输入 x 和输出 y 之间的连线用系数 w 表示, 可以将这个过程理解为线性关系: $x[i]*w[i] = y[i]$ 。

神经网络的原理 —— 多层感知机MLP

● 多层感知机MLP模型

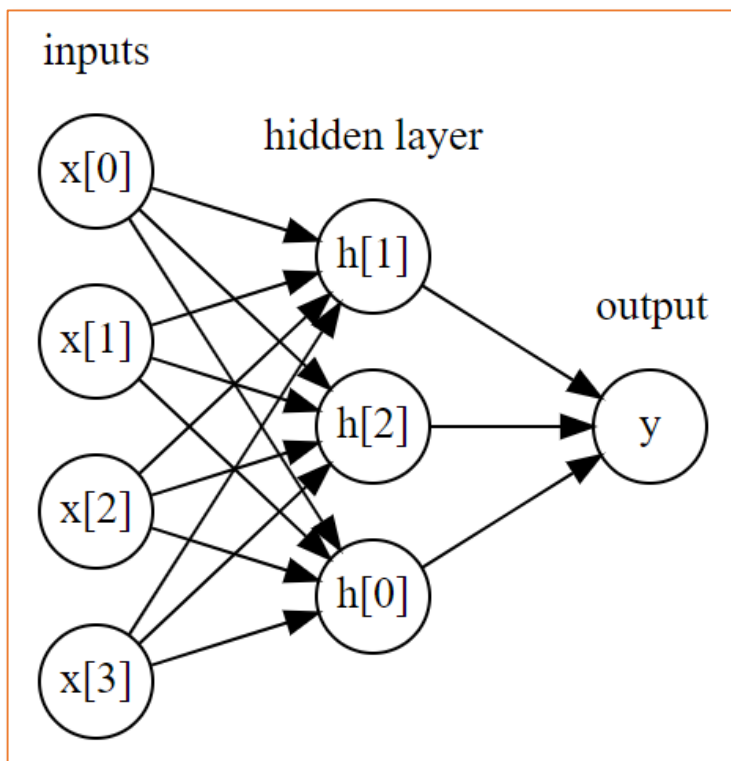
我们在输入层(inputs)和输出层(output)增加一个隐藏层(Hidden Layer), 然后将MLP理解为一个两层的线性模型, 并模拟线性模型中加权求和的过程。

- 第一层模型的输入层是原来的输入层(inputs), 输出层是新增加的隐层(hidden layer), 此时输出变成了隐层单元 $h[j]$.
- 第二层模型的输入层变成了新增加的隐层(hidden layer), 而输出层依然是原来的输出层(output), 此时输入变成了隐层单元 $h[j]$.



神经网络的原理 —— 多层感知机MLP

● 多层感知机MLP模型



与**线性模型**相似，在**第一层模型**中的每一个 $x[i]$ 和隐层单元 $h[j]$ 之间、**第二层模型**中的每一个隐层单元 $h[j]$ 和 y 之间都有一个系数 w 。相似的，在每个模型中也都有一个**偏移量 b** 。

此时，最终的**输出 y** 等于 j 个**加权和的加权和**。

激活函数及非线性矫正

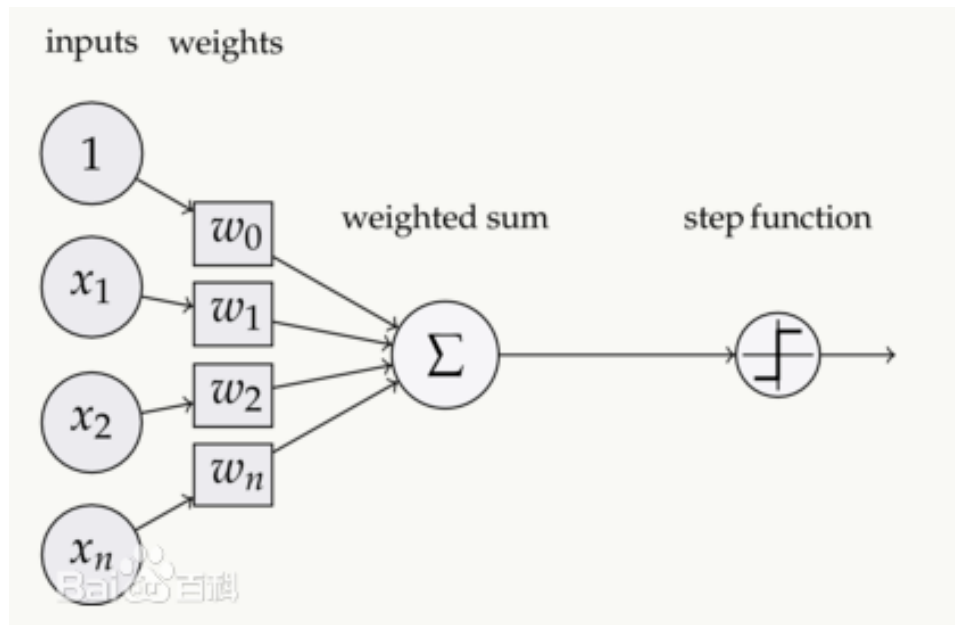
值得注意的是，上面所介绍的MLP模型，并非完整的MLP模型，更非完整的神经网络模型。从数学的角度来分析，不难想象，输出 y 变成了 j 个加权和的加权和，但其结果依然不会有什么区别，为了让模型更强大，我们还需要增加一些处理。例如：

- 非线性（限制线性）激活、
- Dropout、
- Pooling
- shortcut(ResNet)、
- batchNorm
- ...

激活函数及非线性矫正

对于MLP，此处我们仅引入激活函数Activation Function. 基本思路是，在每个隐层后面都增加一个激活函数。假设我们用 $h(x)$ 表示线性变化， $ReLU(x)$ 表示激活函数，则原来的线性变换 $y=f(x)$ 可以表示为： $y=ReLU(f(x))$ 。

而两层MLP就可以表示为 $y = ReLU(f_2(ReLU(f_1(x))))$

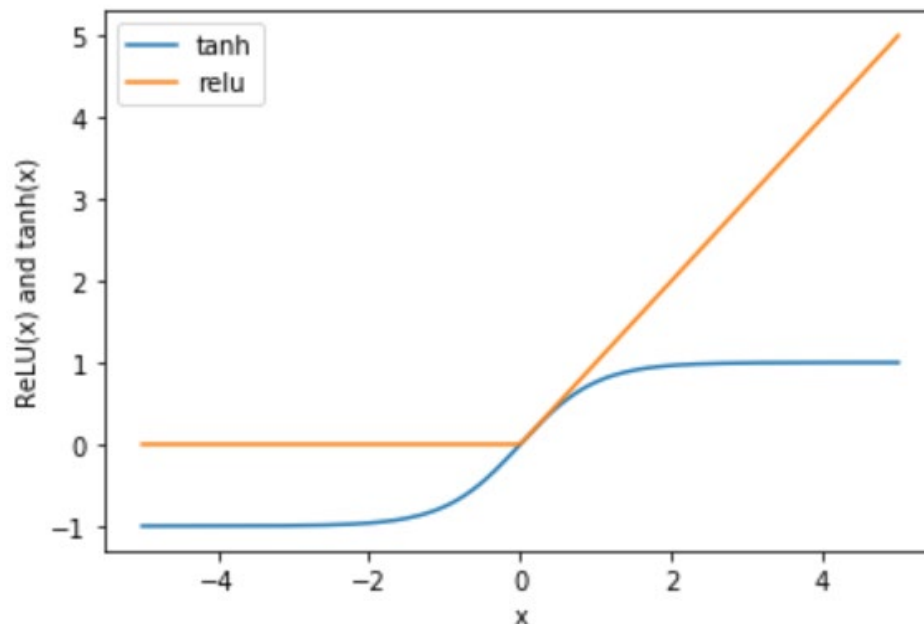


[URL: 激活函数](#)

激活函数及非线性矫正

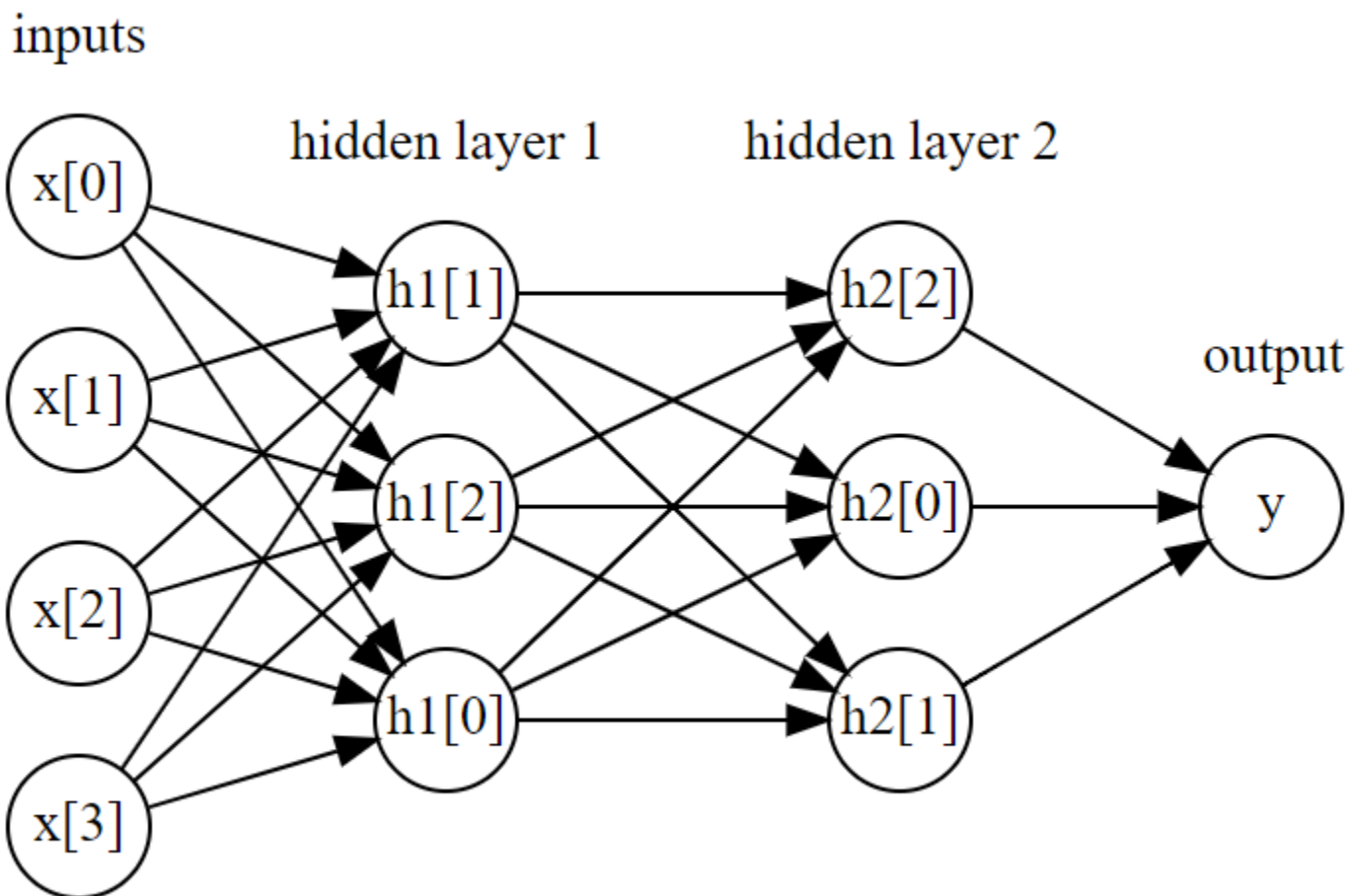
下面给出ReLU和tanh对特征进行限制激活的效果：

- ReLU, 限制线性矫正单元 Rectified Linear Unit
- tanh, 双曲正切 Tangens Hyperbolicus



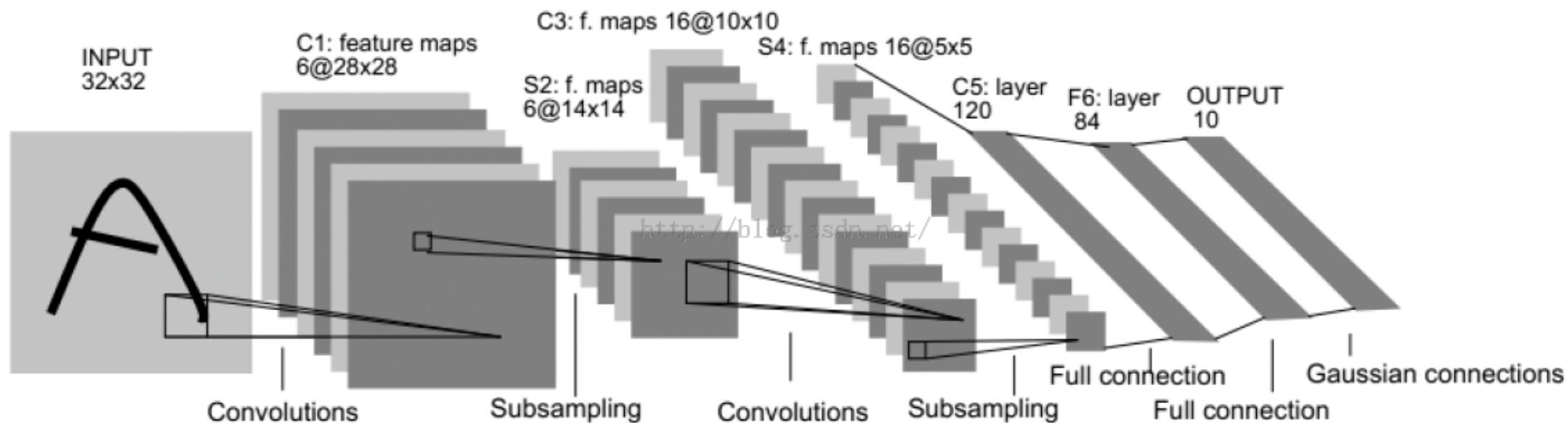
从图中可以看出，**双曲正切函数tanh**将特征**限制**到了 $[-1,1]$ 之间，而**限制线性单元ReLU**则将**小于0的部分直接归0**。

更深的网络



更更深的网络

LeNet-5



更更深的网络

AlexNet

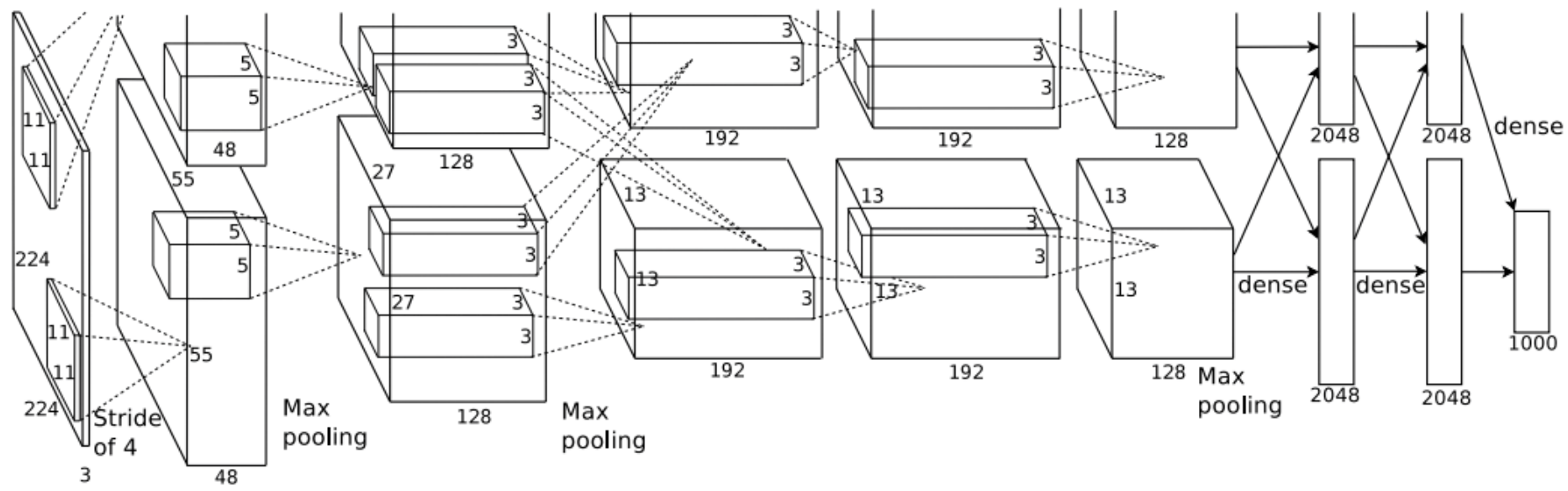
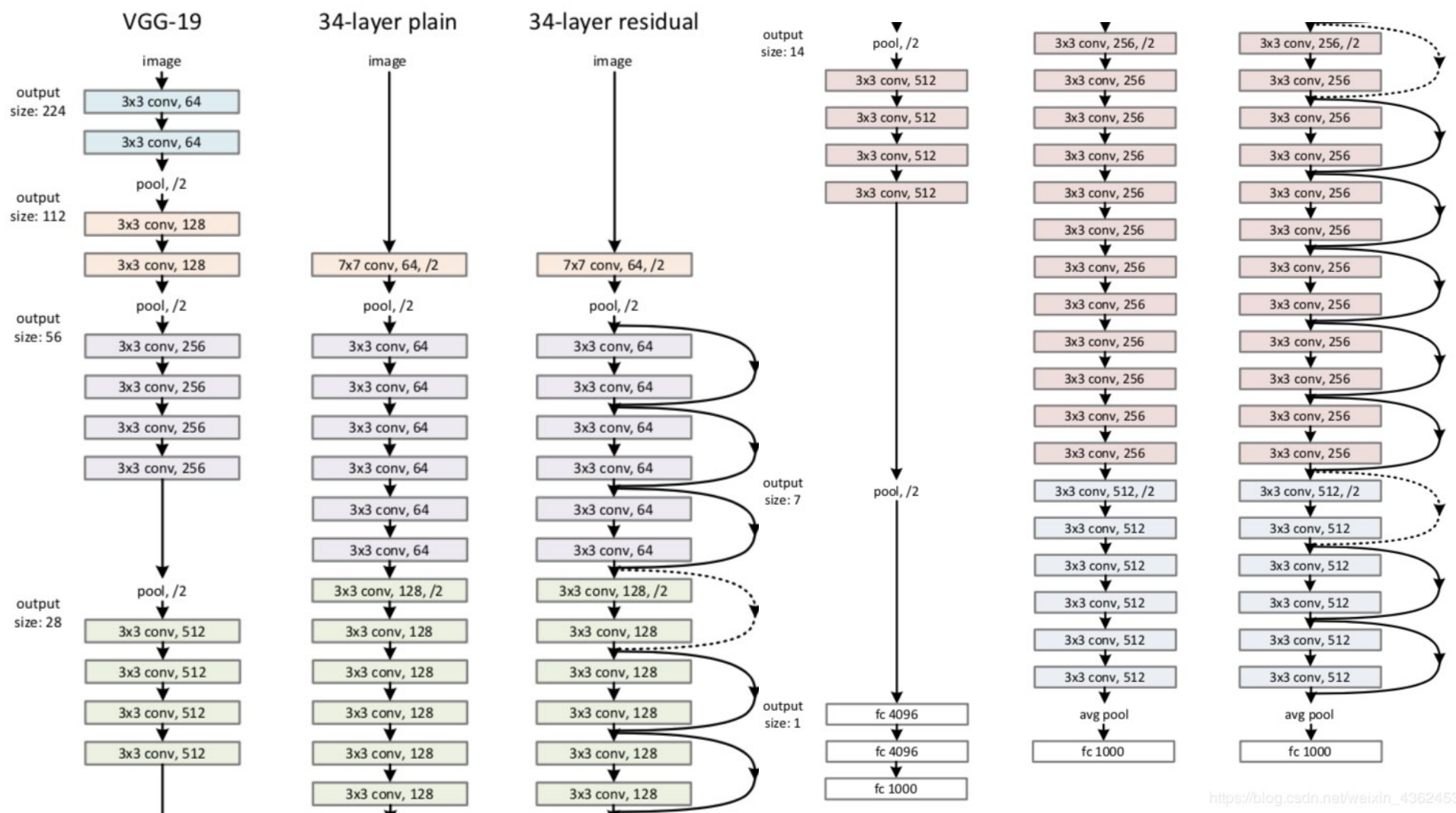


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

更更深的网络

VGG 和 ResNet



https://blog.csdn.net/weixin_4362453

神经网络的使用

Baseline

Ch0801Baseline.py



调整单层的隐藏神经元数量

Ch08021L10.py



调整神经网络的深度

Ch08032L.py



调整激活函数 (ReLU、Tanh)

Ch0804Tanh.py



调节超参数 (Alpha)

Ch0805Alpha.py

MNIST手写字体识别

MNIST数据集被誉为人工智能领域的“Hello World”，不仅因其“简单”，更因为其**实用**。同时也因为它并不是真的那么“简单”，它既可以用来验证普通的**机器学习**算法，也可以被应用到**深度学习**领域。当然，直到卷积神经网络的出现，MNIST的性能才真正提高到了**99.8%**以上。

MNIST手写字体（数字）数据集目前由三驾马车之一的纽约大学教授Yann LeCun实验室维护，包含多种数据格式。整个数据集包含**0-9十个数字**的灰度图片共计**70000**张，都是分辨率为 28×28 的图片，其中训练集样本**60000**张，测试集**10000**张。

MNIST手写字体识别

1. 数据集载入

- 使用内置集成工具载入数据集
- 使用文件读取方式载入数据集

2. 训练MLP神经网络

3. 模型性能分析

- 权值图分析
- 超参数分析
- 测试集评估、单样本评估（测试集样本、自定义样本）
- 保存和载入模型

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ: 14777591 (宇宙骑士)

Email: ouxinyu@alumni.hust.edu.cn

Tel: 18687840023