

StatNet：统计理论指导下的神经网络

程凯越 无 93 2018011113

2023 年 3 月 5 日

摘要

近年来,深度学习发展迅速,在理论层面上有不断的创新,在应用方面在不同领域也表现出了不俗的成绩。特别是 chatGPT 发布后,通用大模型被大众所熟悉。巨大数量的数据和强大的算力是通用大模型的基础,但如何提高数据的利用效率也是一个重要的话题,能够加快训练速度,也可以增强模型的泛化能力。我们将 Boltzmann 统计理论引入深度学习神经网络中,将粒子能量、能级以及体系的熵、平均内能这些物理量,通过建模嵌入神经网络中。通过实验,我们证实了在统计理论指导下设计的神经网络相比于同架构但直接使用原始数据的神经网络,收敛速度更快,同样数据量训练效果更好。我们将代码开源在: <https://github.com/keyork/stat-net>。

1 简介

深度学习图像识别、目标跟踪、文本翻译、自然语言处理、强化学习等领域都取得了广泛的应用和很好的效果,特别是 ResNet 提出后,深层网络的梯度消失得以解决,RNN 和 LSTM 提出后,深度学习对于时间序列的数据也有了很强的建模能力。提出 Attention 和 Transformer 后,神经网络的性能再次得以提升。现在深度学习逐渐向可解释性、因果性和小样本学习方向发展。

统计力学中,Boltzmann 统计理论是很重要的一部分,半经典分布和可区分粒子分布统称为 Boltzmann 分布,以此为基础的理论统称为 Boltzmann 理论。根据 Boltzmann 分布引入配分函数,可以计算得到系统的宏观量,例如内能 E 、压强 P 、熵 S 。

我们在构建了一个三层的全连接网络来完成手写数字识别任务,在全连接网络的基础上,将图像的像素值视为单个粒子的能量测量值,通过构建一个“将粒子逐个加

入系统”的过程，得到整个过程中系统的熵 S 、平均内能 \bar{E} 的变化过程。将变化过程作为数据的特征，作为全连接网络隐藏层的结构。

我们在手写数字识别任务上训练出两个模型，分别是原始的三层全连接网络（记为 MLPNet）和我们通过构建统计力学过程得到的神经网络（记为 StatNet）。在训练过程中，我们发现二者的差异很显著，StatNet 比 MLPNet 的收敛速度更快，我们逐步减少训练数据量，发现 StatNet 在小样本学习的任务上也比 MLPNet 表现更好。

2 相关工作

2.1 全连接神经网络

全连接网络一般由输入层、隐藏层和输出层构成，如图1所示，每一个圆圈代表一个神经元，通常使用 X 表示输入向量，用 W 表示神经元的权值， b 表示神经元的偏置， o 表示神经元的输出， y 表示神经元经过激活函数后的输出。则每一个神经元上的运算都可以表示为：

$$o = WX^T + b, \quad y = f(o),$$

其中 f 代表激活函数，常用的激活函数有 Sigmoid 和 ReLU。

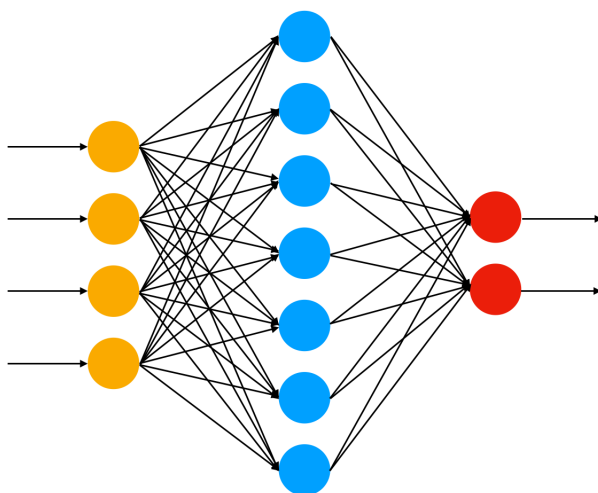


图 1: 全连接网络示意图

神经网络在训练时，分为两个阶段：前向传递和反向传递。在前向传递阶段，神经

网络先根据目前每个神经元的参数计算出一组输出，记为 y_{pred} ，然后计算输出与标签值 y_{label} 的差距：

$$L = f_{loss}(y_{pred}, y_{label}),$$

其中， f_{loss} 表示计算差距的方式，常用的有 L1 距离、L2 距离、MSE 等，总之，前向传递阶段的最终任务就是计算出一组输出，并且根据这组输出求出与标签的差距。在反向传递阶段，根据前向传递得到的 loss 值，计算出 loss 值对每个参数的偏导数，从而作为更新参数的依据，例如某个神经元最初的参数是 (W_i, b_i) ，则在反向传递后，更新后的参数为：

$$W_{i+1} = W_i + \gamma \frac{\partial L}{\partial W_i}, \quad b_{i+1} = b_i + \gamma \frac{\partial L}{\partial b_i},$$

其中， γ 为学习率。使用数据集中的数据依次对神经网络中的参数进行更新，就是神经网络的训练过程。

在训练过程中， L 值是逐渐降低的，这也代表着模型的输出值与真实标签值越来越接近。

2.2 Boltzmann 统计理论

半经典分布和可区分粒子分布统称为 Boltzmann 分布，其数学表达式为

$$n_i = g_i e^{-\alpha - \beta \varepsilon_i},$$

以此为基础的理论通常为 Boltzmann 统计理论。配分函数定义为：

$$Z = \sum_i g_i e^{-\beta \varepsilon_i},$$

这是建立微观和宏观之间的桥梁。通常在计算宏观量时，首先使用量子力学计算出能级 ε_i 和简并度 g_i ，然后求出配分函数 Z ，通过配分函数就可以很简单地得到宏观量。

对于平均内能，我们可以通过推导得到：

$$\bar{E} = \sum_i n_i \varepsilon_i = -n \frac{\partial (\ln Z)}{\partial \beta},$$

其中， $\beta = \frac{1}{kT}$ ， k 为玻尔兹曼常数， T 为温度。

同样，对于熵，我们也可以得到：

$$S = Nk \left(\ln Z - \beta \frac{\partial \ln Z}{\partial \beta} + S' \right),$$

其中, S' 为熵常数。

3 模型

3.1 MLPNet

我们先构建基本的全连接网络如图2所示, 输入是一个 784×1 的向量, 隐藏层有 1024 个神经元, 输出是一个 10×1 的向量。

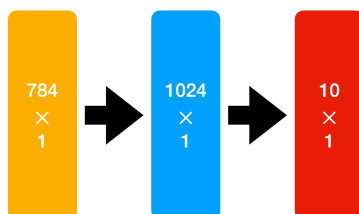


图 2: 基本全连接网络

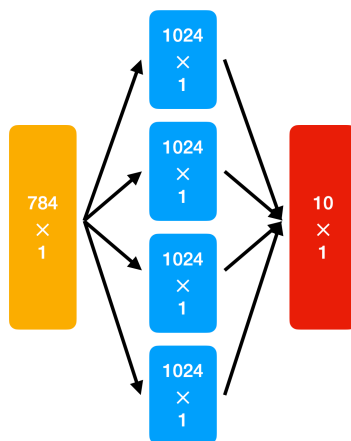


图 3: MLPNet

为了便于后续引入统计力学参量, 我们对于模型进行一些修改如图3所示, 将隐藏层修改为 4 个并行的 1024×1 的神经元, 即从输入层同时连接着 4 组隐藏层, 这 4 组隐藏层同时连接着输出层, 相当于将隐藏层的神经元个数修改为 4096, 但为了便于后续表示, 我们将其拆分成 4 个部分。这就是 MLPNet 的网络结构。

3.2 统计理论

输入为一个 784×1 的向量，我们构建了一个统计力学的模型，来表征这个向量。

将 784×1 的向量的数值归一化到 $[0, 255]$ ，向量中每一个元素代表一个全同粒子，每个元素的数值代表该全同粒子的能量测量值。现有一个其中没有任何粒子的系统，例如三维无限深势阱，将粒子按照顺序逐个归入该系统中，每放入一个粒子，系统的熵和平均内能都会发生改变。我们再作如下假定：

1. 粒子能级的能量值分别为： $(0^2, 1^2, 2^2, \dots, 16^2)$ ；
2. 粒子的能量测量值存在误差，在计算能级时，先将能量值开方，再按照四舍五入的规则计算粒子所处的能级；
3. 每个能级的简并度都为向量长度（即 784）。

记录下每个粒子加入后 {该粒子的能量测量值，该粒子所处的能级，系统此时的熵，系统此时的平均内能} 四组数据，作为我们构建新的神经网络的理论依据。

3.3 StatNet

有了统计理论，我们在 MLPNet 的基础上就可以构建出基于统计理论的神经网络——StatNet，如图4所示。

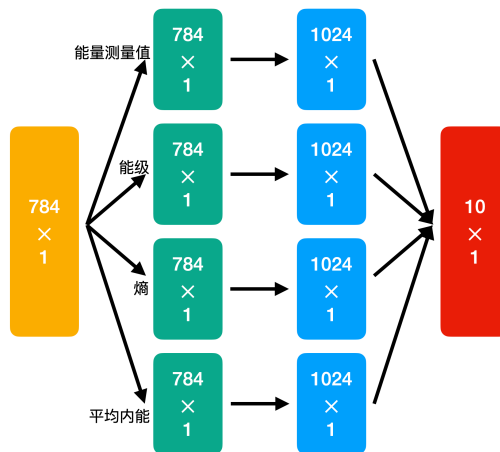


图 4: StatNet

输入 784×1 的向量，按照 3.2 所描述的过程，分别计算整个过程中能量测量值、能级、系统熵、系统平均内能，这些都是 784×1 的向量。再通过全连接层到隐藏层、隐藏层到输出层，得到最终的 10×1 的输出向量。

4 实验

4.1 数据

我们采用手写数字识别数据集 (MNIST) 进行对模型进行评估, 输入一张图像, 输出这个图像上的数字 (0-9)。MNIST 数据集已经划分好了训练集和测试集, 其中, 训练集共有 60000 张图像, 测试集共有 10000 张图像, 图像均为灰度图, 尺寸均为 28×28 。



图 5: 手写数字识别 (MNIST) 数据集

4.2 实验环境

我们使用的工作站的硬件及系统信息如下:

- **CPU:** Intel® Xeon® CPU E5-2650 v4 @ 2.20GHz $\times 2$
- **GPU:** NVIDIA TITAN X (Pascal™)
- **System:** Ubuntu 16.04.7 LTS
- **CUDA:** 10.1

我们使用的编程环境信息如下:

- python 3.7.15
- torch 1.8.0+cu101
- torchvision 0.9.0+cu101
- numpy 1.21.6

4.3 实验设置

我们共进行了两组实验，区别在于是否调整训练数据的数量。

在第一组实验中，我们将训练集划分成 55000 张训练数据和 5000 张验证数据，用于观测训练过程中模型的变化。分别对 MLPNet 和 StatNet 进行训练，记录并对比二者在训练过程中的 loss 值与 acc 值的变化曲线，并对比同一时刻的 loss 值与 acc 值。

在第二组实验中，我们降低训练数据的数量，分别采用 30000 张、10000 张、5000 张、1000 张、500 张、100 张进行训练，同样观测训练过程中模型的变化以及 MLPNet 与 StatNet 的对比。

4.4 实验结果

实验一

我们进行了 20 个 epoch 的训练，记录下训练过程中，训练集 loss、acc 和验证集 loss、acc 如图6所示，最后在测试集上对模型进行测试，得到测试集 loss、acc 见表1。

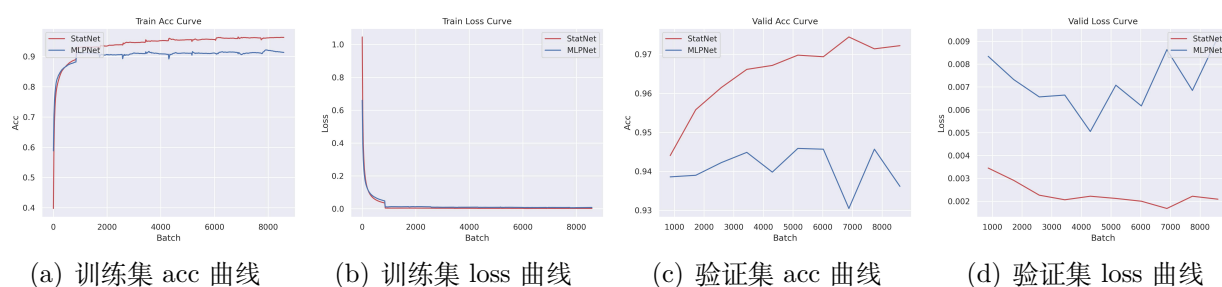


图 6: 实验一实验结果

表 1: 测试集测试结果

模型	测试集 Acc	测试集 Loss
StatNet	97.52%	0.001943
MLPNet	94.55%	0.007593

可见，StatNet 无论是在收敛速度，还是最终的测试集表现，均优于普通的 MLPNet。

实验二

我们调整训练数据的数量，得到模型表现随着训练数据量变化的曲线如图7所示。

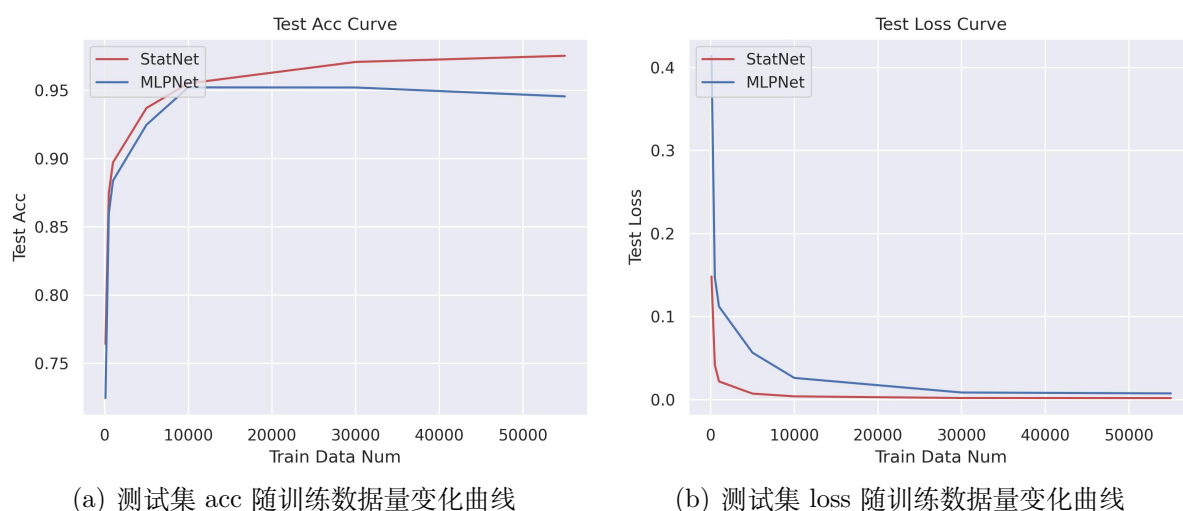


图 7: 实验二实验结果

可见, StatNet 在各种训练数据量下, 表现均优于 MLPNet, 特别是训练数据量较小时, 其 loss 值与 MLPNet 的 loss 值低更多, 说明模型的稳定性和性能更好。

另外, 在实验二进行过程中, 我们记录了每组实验的训练过程, 具体数据及图像可以在 GitHub 仓库中查看: <https://github.com/keyork/stat-net/tree/main/img/exp2>。

5 总结

我们通过实验证明了, 在 Boltzmann 统计理论指导下构建出的神经网络, 在 MNIST 手写数字识别任务上的性能优于常规神经网络, 具体表现为: 同样的训练数据和训练参数下, 收敛速度更快、测试集准确率更高、Loss 值更低, 小样本低数据量的情况下性能损失更小。此实验中是将图像数据 (矩阵) 转换成了向量, 向量中每个值之间的逻辑关联性不强, 对于自然形成的序列数据 (例如: 文本、信号等), 其表现或许与图像场景下的表现存在差异。

我们认为, 此项工作后续可以进行扩展, 例如, 对于卷积神经网络 (CNN) 进行统计理论构建、对于循环神经网络 (RNN) 以及 LSTM 进行统计理论构建并将其应用于序列数据、基于统计理论构造新的模型对神经网络进行建模等。