

考核作业三实验报告——文献阅读报告

文献选择

受到考核问题二的启发，它用退火算法解决的是一种方案的最优解，拓展了简单的找函数最值这一功能，因此我思考这种优化方法是否能够应用在机器学习的模型训练上，因为模型训练的本质是拟合函数，找到模型权重相对于真实数据分布的最优解，这与旅行商问题不无相似。因此在网上搜寻了许多相关文献，找到了一篇简洁明了，思路清晰的文章，是研究模拟退火算法在CNN中应用的：

L.M. Rasdi Rere, Mohamad Ivan Fanany, Aniati Murni Arymurthy,
Simulated Annealing Algorithm for Deep Learning,
Procedia Computer Science,
Volume 72,
2015,
Pages 137-144,
ISSN 1877-0509,
<https://doi.org/10.1016/j.procs.2015.12.114>

原理解析

SA的应用

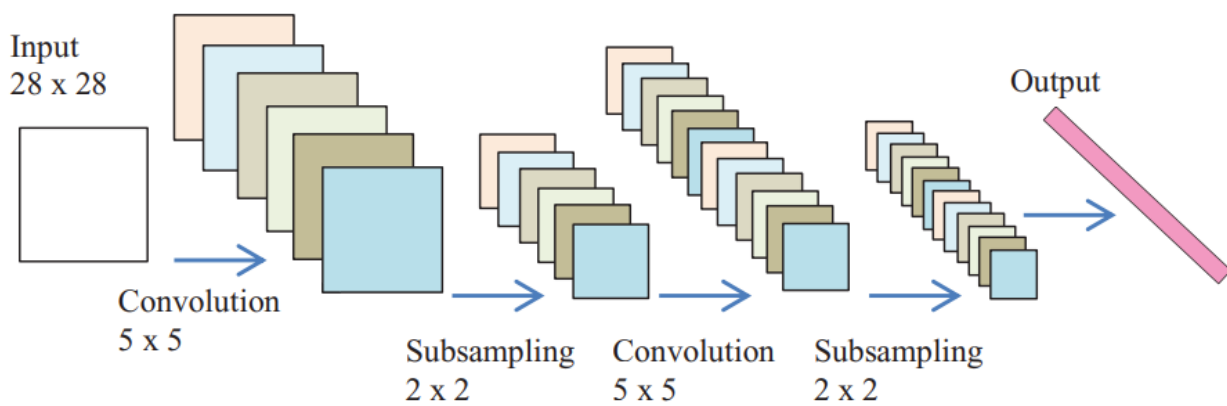
SA算法用在模型训练的一步，将模型的所有权重看成一个向量，这个多维向量即为模型的解定义域，能量函数则为用此权重拟合得到的loss结果，越小越好。如此便转化为了可用SA来求解的问题。其余的步骤和普通的SA差不多，伪代码如下：

```

Simulated annealing algorithm
1  Select the best solution vector  $x_0$  to be optimized
2  Initialize the parameters: temperature  $T$ , Boltzmann's constant  $k$ , reduction factor  $c$ 
3  while termination criterion is not satisfied do
4      for number of new solution
5          for number of iterations
6              Select a new solution:  $x_0 + \Delta x$ 
7              if  $f(x_0 + \Delta x) < f(x_0)$  then
8                   $f_{\text{new}} = f(x_0 + \Delta x)$ ;  $x_0 = x_0 + \Delta x$ 
9              else
10                  $\Delta f = f(x_0 + \Delta x) - f(x_0)$ 
11                 random  $r(0, 1)$ 
12                 if  $r > \exp(-\Delta f/kT)$  then
13                      $f_{\text{new}} = f(x_0 + \Delta x)$ ,  $x_0 = x_0 + \Delta x$ 
14                 else
15                      $f_{\text{new}} = f(x_0)$ ,
16                 end if
17             end if
18              $f = f_{\text{new}}$ 
19             Decrease the temperature periodically:  $T = c \times T$ 
20         end for
21     end while
```

模型结构

模型结构是简单的CNN：

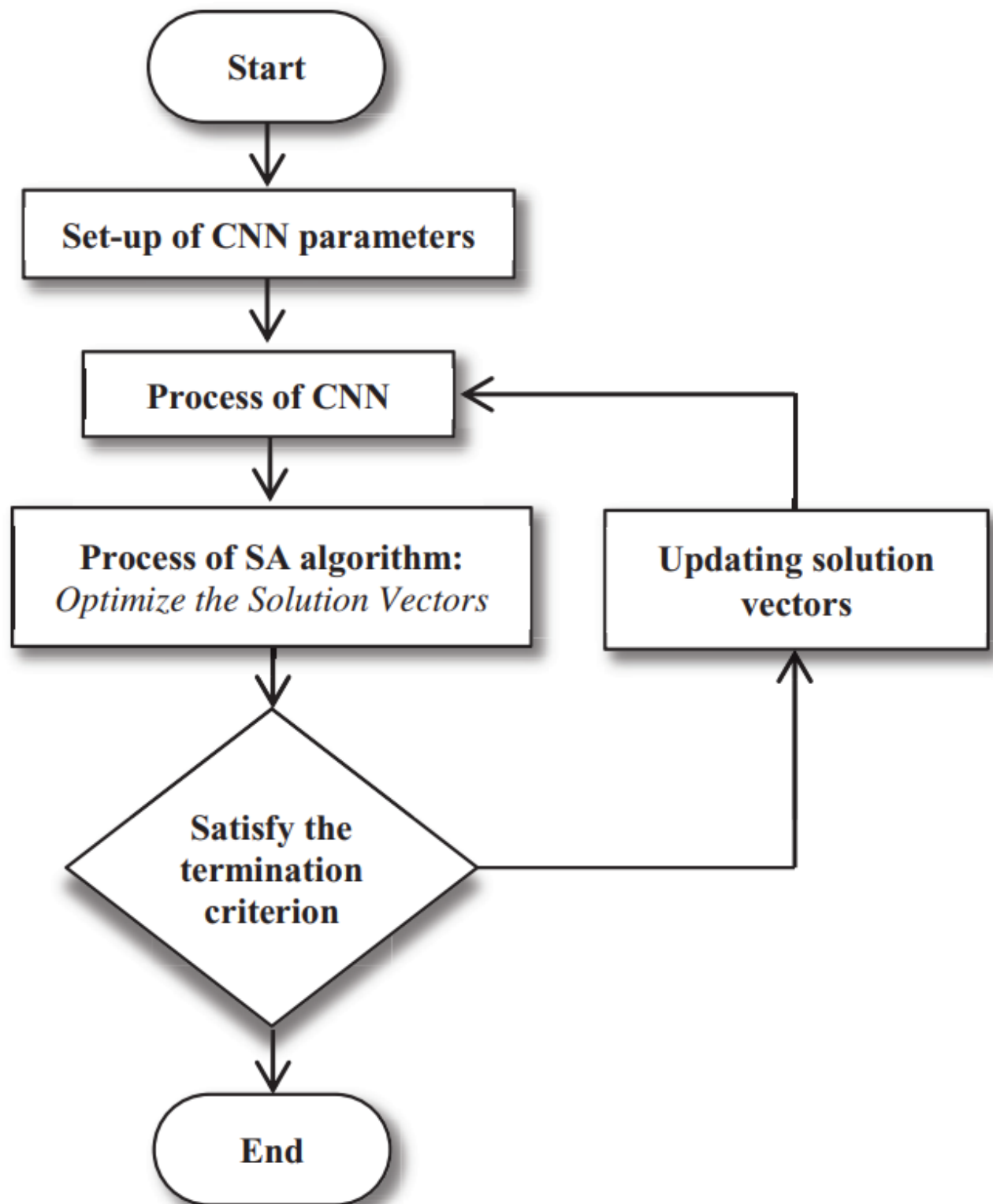


损失函数

损失函数同时也是能量函数，是简单的MSE，如下：

$$f = \frac{1}{2} \sqrt{\frac{\sum_{i=1}^R (o - y)^2}{R}}$$

整体流程：



实验结论

本文章展示的实验是简单的手写体数字识别，给出了原始CNN，CNN+SA（10，20，50内循环次数）对比的准确率，错误率以及优化时间，如下：

Table 1. Performance comparison of cnn, cnnSA10, cnnSA20 and cnnSA50 in terms of accuracy

Methods	Number of epochs									
	1	2	3	4	5	6	7	8	9	10
cnn	88.87	92.25	93.90	94.81	95.44	96.18	96.59	96.92	97.22	97.27
cnnSA10	89.18	92.38	94.20	95.19	95.81	96.50	96.77	97.04	97.27	97.41
cnnSA20	90.50	92.77	94.68	95.45	96.66	96.87	97.08	97.26	97.30	97.61
cnnSA50	91.10	94.16	95.49	96.20	96.91	96.99	97.33	97.42	97.40	97.71

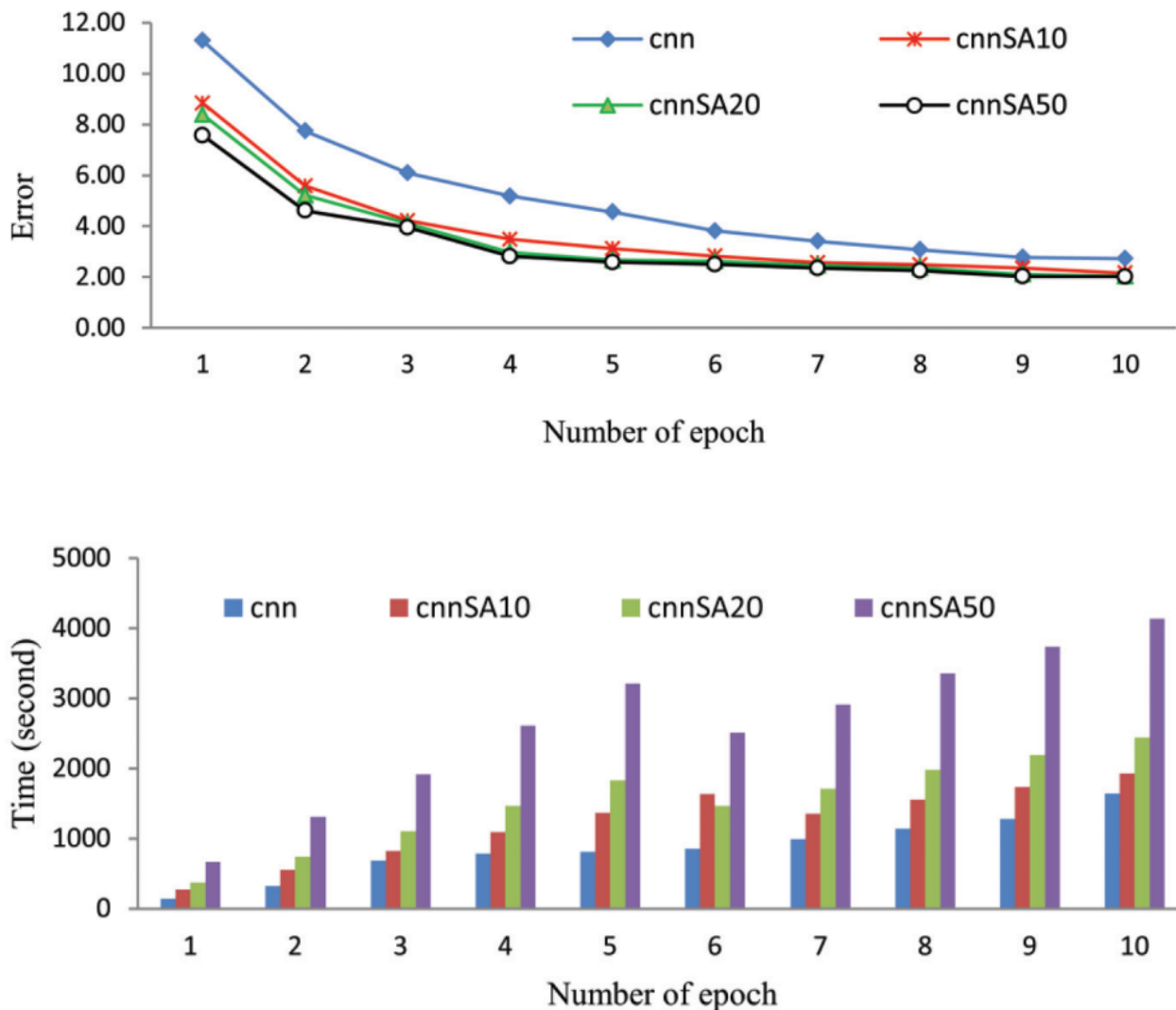


Fig. 7 Computation time for cnn, cnnSA10, cnnSA20 and cnnSA50

可以看出，用SA优化的CNN效果在epoch层面是比普通方法好的，主要体现在早期的epoch。但是这只是epoch层面，再从时间消耗来看SA优化明显耗时更久，但是作者只训练到10epoch，没有后面的数据，若后面准确率差不多，那么SA优化优势就不明显了，即耗时训练效果又没有本质提升。

个人思考

首先，将SA应用在模型训练上是很有效的，但是在阅读的过程中其实能够发现，SA可能只适用与这种小数据+小模型的模式，因为这样的随机是有意义的，如果模型超级大，权重参数及其多，靠SA这么随机跳显然不是很合理。同时，SA算能量又要用数据过模型，这在大规模数据下显然耗时会暴增，弱势更加体现。所以我认为用SA的优化方法做小实验是合理的，真正用于大模型训练还有待考证。