

DOI:10.3876/j.issn.1000-1980.2013.04.014

单隐层感知机神经网络对权扰动的敏感性计算

曾晓勤,何嘉晟
(河海大学计算机与信息工程学院,江苏 南京 210098)

摘要:为使 MLP 神经网络对权扰动的敏感性计算算法更符合实际需求,提出一个不依赖隐层向量相互独立假设的敏感性计算算法。该算法通过自底向上依次计算单个神经元和单个层的敏感性来获得整个网络的敏感性,并采用数值计算方法给出敏感性的数学表达式。以 UCI 数据集为基础,验证该算法的计算精度,结果表明该算法得到的敏感性理论值能较好地逼近敏感性的模拟值。
关键词:敏感性;神经网络;单隐层感知机
中图分类号:TP183 **文献标志码:**A **文章编号:**1000-1980(2013)04-0360-05

**Computation of sensitivity of
single hidden-layer perceptron to weight perturbation**
ZENG Xiaoqin, HE Jiasheng
(College of Computer and Information Engineering, Hohai University, Nanjing 210098, China)

Abstract: In this study, an algorithm was developed to compute the sensitivity of a single hidden-layer perceptron to weight perturbation. This algorithm, which only requires that the first layer's input elements be independent of each other but which has no such a restriction on the hidden layer, is more adaptable to real applications. The sensitivity was computed bottom-up: the sensitivity of a neuron was considered first and the entire network was considered subsequently. The sensitivity expression was obtained with the numerical calculation method. Experimental results on UCI datasets verified the precision of the proposed algorithm. The computed value approached the theoretical value.
Key words: sensitivity; neural network; single hidden-layer perceptron

神经网络对于参数扰动的敏感性是衡量网络性能的重要尺度,如衡量容错性能与泛化性能等。目前有关学者已从不同角度、采用不同方法对不同神经网络模型做了许多研究^[1-6],研究结果被应用于解决神经网络研究领域里的一些热点问题,如提高网络容错性能与泛化性能^[7-8]、网络结构和属性的筛选^[9-10]、改进学习算法^[11-12]等。笔者拟在 Zeng 等^[5-6]的研究基础上,通过函数逼近来避免求解多重积分,推导 2 层前馈连续型单隐层感知机(MLP)神经网络对权扰动的敏感性计算公式。推导中只要求输入层输入的分量相互独立,不再假设隐层输入的分量必须相互独立,使得计算出的敏感性更符合实际情况,也在一定程度上提高了计算精度。

1 MLP 神经网络及符号定义

通常情况下,设一个 MLP 神经网络有 L 层,第 $l(1 \leq l \leq L)$ 层有 $n_l(n_l \geq 1)$ 个神经元。对第 $i(1 \leq i \leq n_l)$ 个神经元,输入为 $X_l = (x_{l,1}, x_{l,2}, \dots, x_{l,n_{l-1}})^T$,输入扰动为 $\Delta X_l = (\Delta x_{l,1}, \Delta x_{l,2}, \dots, \Delta x_{l,n_{l-1}})^T$,它的权为 $W_{l,i} = (w_{l,i,1}, w_{l,i,2}, \dots, w_{l,i,n_{l-1}})^T$,它的权扰动为 $\Delta W_{l,i} = (\Delta w_{l,i,1}, \Delta w_{l,i,2}, \dots, \Delta w_{l,i,n_{l-1}})^T$,偏置为 $b_{l,i}$,激活函数为 $f_l(x)$,输出为 $y_{l,i} = f_l(X_l^T \cdot W_{l,i} + b_{l,i})$ 。对于第 l 层,每一个神经元的输入都是相同的 X_l 。层权为 $W_l = \{W_{l,1}, W_{l,2}, \dots,$

$W_{l,n_l}\}$, 输出为 $Y_l=(y_{l,1},y_{l,2},\cdots,y_{l,n_l})^T$ 。神经网络的输入是第 1 层的输入 X_1 , 输出是最后一层的输出 Y_L 。
MLP 神经网络是由若干个神经元组合成层, 然后层与层之间前后连接构成的网络。相邻层的神经元全连接, 同一层与非相邻层的神经元无连接, 除了输入层与输出层之外的层都称为隐层。不计输入层, 其余每一层的输入都是前一层的输出。同一层每一个神经元都有相同的激活函数 $f_l(x)$, 常见的激活函数为 S 形函数:

$$f_l(x) = \frac{1}{1 + e^{-x}} \text{ 或 } f_l(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

在解决某些问题时, 根据网络输出范围的要求, 输出层的激活函数可能设为线性函数:
$$f_L(x) = x \quad (2)$$

笔者只讨论单隐层的 MLP 神经网络, 即 $L=2$, 因为已经证明, 只要隐层神经元数目足够多, 单隐层的 MLP 神经网络可以逼近任意的连续函数^[13]。本文用 $E(x)$, $D(x)$, $\phi(x)$ 分别代表随机变量 x 的数学期望、方差、密度函数。

2 MLP 神经网络权扰动敏感性定义

在 MLP 神经网络的参数发生扰动时, 网络输出会产生相应变化。MLP 神经网络参数敏感性就是反映参数改变对网络输出的影响程度。敏感性的定义通常不唯一。现实中, 网络输入或者权发生扰动最为常见。笔者主要研究权扰动产生的输出扰动:

$$\Delta y_{l,i} = f_l[X_l^T \cdot (W_{l,i} + \Delta W_{l,i}) + b_{l,i}] - f_l(X_l^T \cdot W_{l,i} + b_{l,i}) \quad (3)$$

在隐层中 X_l 也有扰动, 这是因为 MLP 神经网络前一层权的改变会影响后一层的输入。对于一个训练好的 MLP 神经网络, 它的 $W_{l,i}$, $b_{l,i}$ 都是确定的, 理论上只要有 X_l 和 $\Delta W_{l,i}$ 就可以计算得到 $\Delta y_{l,i}$ 。在定义敏感性时, 把网络的输入看作是随机变量, 其他参数都视为常数, 这样的敏感性是基于所有输入的敏感性。又由于输入扰动产生的影响可正可负, 为避免求期望时正负相消, 敏感性便定义为输出变化绝对值对所有输入的期望。

定义 1: 对第 $l(1 \leq l \leq L)$ 层的第 i 个神经元, 定义它的敏感性 $s_{l,i}$ 是 $\Delta W_{l,i}$ 的函数:
$$s_{l,i} = E_{X_l}(|f_l[(X_l + \Delta X_l)^T \cdot (W_{l,i} + \Delta W_{l,i}) + b_{l,i}] - f_l(X_l^T \cdot W_{l,i} + b_{l,i})|) \quad (4)$$

定义 2: 第 l 层的敏感性定义为该层所有神经元的敏感性构成的向量:
$$S_l = (s_{l,1}, s_{l,2}, \cdots, s_{l,n_l})^T \quad (5)$$

定义 3: 整个 MLP 神经网络的敏感性定义为输出层的敏感性:
$$S_{MLP} = S_L = (s_{L,1}, s_{L,2}, \cdots, s_{L,n_L})^T \quad (6)$$

3 MLP 神经网络权扰动敏感性计算公式推导

对于单隐层 MLP 神经网络, 需要计算的敏感性只有 2 个, 即隐层和输出层的敏感性。在本文的计算过程中, 始终假设输入层的输入分量 $x_{1,i}$ 是相互独立的。

3.1 隐层神经元的敏感性

第 1 层第 i 个神经元的敏感性为
$$s_{1,i} = E_{X_1}(|f_1[X_1^T \cdot (W_{1,i} + \Delta W_{1,i}) + b_{1,i}] - f_1(X_1^T \cdot W_{1,i} + b_{1,i})|) \quad (7)$$

为简化推导过程, 作如下符号替换:
$$z_{1,i} = X_1^T \cdot W_{1,i} \quad \Delta z_{1,i} = X_1^T \cdot \Delta W_{1,i} \quad (8)$$

因为激活函数(S 形函数)是单调递增函数:
$$s_{1,i} = \begin{cases} E(f_1(z_{1,i} + \Delta z_{1,i} + b_{1,i})) - E(f_1(z_{1,i} + b_{1,i})) & \Delta z_{1,i} \geq 0 \\ E(f_1(z_{1,i} + b_{1,i})) - E(f_1(z_{1,i} + \Delta z_{1,i} + b_{1,i})) & \Delta z_{1,i} < 0 \end{cases} \quad (9)$$

因此要得到 $s_{1,i}$, 只需要计算 $E(f_1(z_{1,i} + \Delta z_{1,i} + b_{1,i}))$ 和 $E(f_1(z_{1,i} + b_{1,i}))$ 。

根据期望的定义, 有
$$E(f_1(z_{1,i} + \Delta z_{1,i} + b_{1,i})) = \int \cdots \int_{n_0} f_1\left(\sum_{j=1}^{n_0} x_{1,j}(w_{1,j} + \Delta w_{1,j}) + b_{1,i}\right) \phi(x_{1,1}) \cdots \phi(x_{1,n_0}) d(x_{1,1}) \cdots d(x_{1,n_0}) \quad (10)$$

式(10)为一个 n_0 重积分,直接计算很难。笔者使用 Remez 算法^[14]将激活函数 $f_1(x)$ 逼近为一个多项式之和,同时式中的每一项可以继续展开。Remez 算法是采用逐次逼近思想的求连续函数最佳一致逼近多项式的近似算法。对于一个给定的连续函数,可以找到在给定区间上的 p 次最佳一致逼近多项式 $P(x)$:

$$f_1(x) \approx P(x) = \sum_{t=0}^p c_t x^t \tag{11}$$

式中: p ——多项式的最高次,且当它取值越大时, $f_1(x)$ 和 $P(x)$ 的差距越小; c_t ——多项式系数,对于给定的 p 和 $f_1(x)$,在指定区间内,可以通过 Remez 计算得到。

将式(11)代入式(10),可得 $E(f_1(z_{1,i}+\Delta z_{1,i}+b_{1,i}))$ 的逼近多项式为

$$E(f_1(z_{1,i} + \Delta z_{1,i} + b_{1,i})) \approx \sum_{t=0}^p c_t E(z_{1,i} + \Delta z_{1,i} + b_{1,i})^t \tag{12}$$

将式(12)中的 $(z_{1,i}+\Delta z_{1,i}+b_{1,i})^t$ 进行多项式展开,并求期望得

$$\begin{aligned} E((z_{1,i} + \Delta z_{1,i} + b_{1,i})^t) &= \sum_{a_1+\cdots+a_{n_0+1}=t} \frac{t!}{\prod_{k=1}^{n_0+1} (a_k)!} \left[\prod_{k=1}^{n_0} E(x_{1,k})^{a_k} \right] \left\{ \left[\prod_{k=1}^{n_0} (w_{1,ik} + \Delta w_{1,ik})^{a_k} \right] b_{1,i}^{a_{n_0+1}} \right\} = \\ &= \sum_{a_1+\cdots+a_{n_0+1}=t} \frac{t!}{\prod_{k=1}^{n_0+1} (a_k)!} b_{1,i}^{a_{n_0+1}} \prod_{k=1}^{n_0} (w_{1,ik} + \Delta w_{1,ik})^{a_k} \int x_{1,k}^{a_k} \phi(x_{1,k}) dx_{1,k} \end{aligned} \tag{13}$$

式中: a_k ——非负整数, $0 \leq a_k \leq t$; $\sum_{a_1+\cdots+a_{n_0+1}=t}$ ——取遍所有使得 $a_1+a_2+\cdots+a_{n_0}+a_{n_0+1}=t$ 成立的 a_k 序列;

$\phi(x_{1,1}), \phi(x_{1,2}), \cdots, \phi(x_{1,n_0})$ ——每一维输入的密度函数,多数情况下它们是不同的。

经过类似推导可得

$$\begin{cases} E(f_1(z_{1,i} + b_{1,i})) \approx \sum_{t=0}^p c_t E((z_{1,i} + b_{1,i})^t) \\ E((z_{1,i} + b_{1,i})^t) = \sum_{a_1+\cdots+a_{n_0+1}=t} \frac{t!}{\prod_{k=1}^{n_0+1} (a_k)!} \left(\prod_{k=1}^{n_0} w_{1,ik}^{a_k} \right) b_{1,i}^{a_{n_0+1}} \prod_{k=1}^{n_0} \int x_{1,k}^{a_k} \phi(x_{1,k}) dx_{1,k} \end{cases} \tag{14}$$

3.2 输出层神经元的敏感性

对于单隐层 MLP 神经网络的输出层,第 k 个神经元的敏感性为

$$s_{2,k} = E(|f_2((X_2 + \Delta X_2)^T \cdot (W_{2,k} + \Delta W_{2,k}) + b_{2,k}) - f_2(X_2^T \cdot W_{2,k} + b_{2,k})|) \tag{15}$$

为求 $s_{2,k}$,只需要求得 $E(f_2((X_2+\Delta X_2)^T \cdot (W_{2,k}+\Delta W_{2,k})+b_{2,k}))$ 和 $E(f_2(X_2^T \cdot W_{2,k}+b_{2,k}))$ 。

由于 $s_{1,i}$ 已知,令 $b_{2,k'} = |s_{1,q}|(w_{2,kq}+\Delta w_{2,kq})+b_{2,k}$ (常数替换不影响后面计算),并将 $E(f_2((X_2+\Delta X_2)^T \cdot (W_{2,k}+\Delta W_{2,k})+b_{2,k}))$ 进行多项式逼近,得

$$\begin{aligned} E(f_2((X_2 + \Delta X_2)^T \cdot (W_{2,k} + \Delta W_{2,k}) + b_{2,k})) &\approx \sum_{t_2=0}^p c_{t_2} E([(X_2 + \Delta X_2)^T \cdot (W_{2,k} + \Delta W_{2,k}) + b_{2,k}]^{t_2}) = \\ &= \sum_{t_2=0}^p c_{t_2} E\left(\left[\sum_{q=1}^{n_1} x_{2,q}(w_{2,kq} + \Delta w_{2,kq}) + b_{2,k'}\right]^{t_2}\right) \end{aligned} \tag{16}$$

将 $\left[\sum_{q=1}^{n_1} x_{2,q}(w_{2,kq} + \Delta w_{2,kq}) + b_{2,k'}\right]^{t_2}$ 进行多项式展开,代入式(14)并求期望得

$$\begin{aligned} E\left(\left[\sum_{q=1}^{n_1} x_{2,q}(w_{2,kq} + \Delta w_{2,kq}) + b_{2,k'}\right]^{t_2}\right) &= \\ &= \sum_{a_1+\cdots+a_{n_1+1}=t_2} \frac{t_2!}{\prod_{r=1}^{n_1+1} (a_r)!} E\left(\left\{\prod_{r=1}^{n_1} \left[\sum_{t=0}^p c_t (z_{1,r} + b_{1,r})^{a_r}\right]\right\} \left[\prod_{r=1}^{n_1} (w_{2,kr} + \Delta w_{2,kr})^{a_r}\right] b_{2,k'}^{a_{n_1+1}}\right) \end{aligned} \tag{17}$$

其中 $E\left(\prod_{r=1}^{n_1} \left[\sum_{t=0}^p c_t (z_{1,r} + b_{1,r})^{a_r}\right]\right) = E\left(\prod_{r=1}^{n_1} \left[\sum_{t=0}^p c_t \sum_{d_1+\cdots+d_{n_0+1}=t} \left(\frac{t!}{\prod_{s=1}^{n_0+1} d_s!} b_{1,r}^{d_{n_0+1}} \prod_{s=1}^{n_0} x_{1,s}^{d_s} \prod_{s=1}^{n_0} w_{1,rs}^{d_s}\right)\right]\right)^{a_r}$

同理可得

$$E(f_2(X_2^T \cdot W_{2,k} + b_{2,k})) = \sum_{t_2=0}^p c_{t_2} E((\sum_{i=1}^{n_1} x_{2,i} w_{2,ki} + b_{2,k})^{t_2}) =$$
$$\sum_{a_1+\cdots+a_{n_1+1}=t_2} \frac{t_2!}{\prod_{r=1}^{n_1+1} a_r!} E\left(b_{2,k}^{a_{n_1+1}} \prod_{r=1}^{n_1} [\sum_{t=0}^p c_t (z_{1,r} + b_{1,r}^{t_1})^{a_r}] (\prod_{r=1}^{n_1} w_{2,kr}^{a_r})\right) \quad (18)$$

由于输入层的输入分量 $x_{1,s}$ 是相互独立的,于是计算期望时的多重积分可以变为一重积分的积,因此式 (17) 是可计算的。由于已经证明单隐层 MLP 神经网络可以逼近任意连续函数,因此本文算法解决了 MLP 神经网络权扰动敏感性的计算问题。

4 算 法 验 证

以 UCI 数据集^[15]为基础,通过对比 MLP 神经网络参数敏感性的模拟值和算法得到的理论值验证所提算法的准确性。笔者采用 MATLAB 7. 11. 0 (R2010b) 版本的神经网络工具箱获取模拟值, MATLAB 获取理论值。

从 UCI 数据集中取出 iris, wine, vehicle 共 3 组数据集,针对每组数据集训练几个不同的 MLP 神经网络。训练完毕后,随机生成 1 万个正态分布的权扰动值。然后将输入再次输入网络,得到 MLP 神经网络权扰动敏感性的模拟值。将权扰动值、网络输入、权等代入式 (16) (17) 可得到理论值。

实验结果如下:实验 1, iris 与 wine 数据集,输入维数为 4,输出维数为 1,输入样本数为 150,网络均方误差均小于 0. 005,权扰动期望为 0,方差为 0. 1;实验 2, iris 数据集,输入维数为 4,输出维数为 1,输入样本数为 150,网络均方误差均小于 0. 005,隐层节点数为 5,权扰动期望为 0,方差分别为 0. 05, 0. 3, 0. 5;实验 3, vehicle 数据集,输入维数为 18,输入样本数为 878,网络均方误差均小于 0. 01,隐藏节点数为 7,权扰动期望为 0,方差为 0. 1。

从模拟结果(表 1~3)可以看出,模拟值与理论值有一定的差异,但是就趋势而言,基本保持一致。也就是使用本算法计算出的理论值可以作为一个相对尺度来应用。

表 1 iris 和 wine 数据集不同隐层节点数的模拟结果

Table 1 Simulated results of sensitivity for nodes of different hidden layers (iris and wine datasets)

隐层节点数	敏感性模拟值		敏感性理论值	
	iris	wine	iris	wine
4	0. 174 350 0	0. 281 16	0. 402 840 0	0. 628 730
5	0. 002 143 5	0. 515 72	0. 007 699 2	0. 058 728
7	0. 058 601 0	0. 183 37	0. 116 704 0	0. 224 710

表 2 iris 数据集不同权扰动方差的模拟结果

Table 2 Simulated results of sensitivity for different variances of weight perturbation (iris dataset)

权扰动方差	敏感性模拟值	敏感性理论值
0. 05	0. 002 143 5	0. 007 699 2
0. 30	0. 076 011 0	0. 121 994 0
0. 50	0. 160 410 0	0. 310 800 0

5 结 语

在将 MLP 神经网络敏感性定义为输出变化绝对值对所有输入的期望的基础上,通过 Remez 算法推导了敏感性的近似计算公式。实验数据检验结果表明,敏感性的理论值能较好地逼近模拟值,两者间的误差由算法中使用函数逼近所致。在逼近过程中选用的多项式次数越高,则计算精度越高;无论多项式次数的多少,理论值和模拟值的变化趋势基本上保持一致。

参考文献:

[1] FU L, CHEN T. Sensitivity analysis for input vector in multilayer feedforward neural networks[C]//Proc of IEEE International Conference on Neural Networks. San Francisco, CA:IEEE Press, 1993:215-218.

表 3 vehicles 数据集激活函数为 S 形函数与线性函数的模拟结果

Table 3 Simulated results of sensitivity for different activation functions (vehicle dataset)

激活函数	敏感性模拟值	敏感性理论值
S 形函数	0. 349 55	0. 560 51
线性函数	0. 302 08	0. 513 07

- [2] YEUNG D S, SUN Xuequan. Using function approximation to analyze the sensitivity of the MLP with antisymmetric squashing activation function[J]. IEEE Transactions on Neural Networks, 2002,13(1): 34-44.
- [3] ZENG Xiaoqin, YEUNG D S. Sensitivity analysis of multilayer perceptron to input and weight perturbations[J]. IEEE Transactions on Neural Networks, 2001, 12(6): 1358-1366.
- [4] ZENG Xiaoqin, YEUNG D S. A quantified sensitivity measure for multilayer perceptron to input perturbation[J]. Neural Computation, 2003, 15(1): 183-212.
- [5] YANG Jing, ZENG Xiaoqin, ZHONG Shuiming. Computation of multilayer perceptron sensitivity to input perturbation[J]. Neurocomputing, 2013, 99(1): 390-398.
- [6] YANG S, HO C, SIU S. Sensitivity analysis of the split-complex valued multilayer perceptron due to the errors of the i. i. d. Inputs and Weights[J]. IEEE Trans on Neural Networks, 2007, 18(5): 1280-1293.
- [7] BERNIER J L, ORTEGA J, ROJAS I, et al. Improving the tolerance of Multilayer Perceptrons by minimizing the statistical sensitivity to weight deviations[J]. Neurocomputing, 2000, 31(1): 87-103.
- [8] BERNIER J L, ORTEGA J, ROS E, et al. A quantitative study of fault tolerance, noise immunity and generalization ability of MLPs[J]. Neural Computation, 2000, 12(12): 2941-2964.
- [9] ENGELBRECHT A P. A new pruning heuristic based on variance analysis of sensitivity information[J]. IEEE Transactions on Neural Networks, 2001, 12(6): 1386-1399.
- [10] ZENG Xiaoqin, YEUNG D S. Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure[J]. Neurocomputing, 2006, 69(7/8/9): 825-837.
- [11] ZHONG Shuiming, ZENG Xiaoqin, WU Shengli, et al. Sensitivity-based adaptive learning rules for BFNNs[J]. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(3): 480-491.
- [12] 高玉琴, 张利昕, 吴焕霞. 改进 GA-BP 神经网络评价算法及其应用[J]. 水利经济, 2012, 30(6): 7-10. (GAO Yuqin, ZHANG Lixin, WU Huanxia. Evaluation algorithm of improved GA-BP neural network and its application[J]. Journal of Economics of Water Resource, 2012, 30(6): 7-10. (in Chinese))
- [13] HORNIK K M, STINCHCOMBEA M, WHITE H. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359-366.
- [14] RICARDO P, TREFETHEN L N. Barycentric-Remez algorithms for best Polynomial approximation in the chebfun system[J]. BIT Numerical Mathematics, 2009, 49(4): 721-741.
- [15] BLAKE C L, MERZ C J. UCI repository of machine learning databases[DB]. [2013-01-20]. <http://archive.ics.uci.edu/ml/index.html>.