A SMOOTH OPTIMISATION PERSPECTIVE ON TRAINING FEEDFORWARD NEURAL NETWORKS

Hao Shen
Department of Electrical and Computer Engineering
Technical University of Munich, Germany
hao.shen@tum.de

ABSTRACT

We present a smooth optimisation perspective on training multilayer Feedforward Neural Networks (FNNs) in the supervised learning setting. By characterising the critical point conditions of an FNN based optimisation problem, we identify the conditions to eliminate local optima of the cost function. By studying the Hessian structure of the cost function at the global minima, we develop an approximate Newton FNN algorithm, which demonstrates promising convergence properties.

1 INTRODUCTION

Despite the recent great success of deep neural networks in various applications, training a deep neural network is still among the greatest challenges in the field, cf. (Glorot & Bengio, 2010). In this abstract, we focus on the study of training the Feedforward Neural Networks (FNNs) to solve supervised learning problems. One major reason for the difficulty in training an FNN is that their performance is highly dependent on various factors, such as the architecture of an FNN (Hornik, 1991; Sun et al., 2016), the specific activation function (Mhaskar & Micchelli, 1993), and the choice of error functions (Falas & Stafylopatis, 1999), in a very complicated way.

The most popular FNN training algorithm is the backpropagation (BP) algorithm, cf. (Widrow & Lehr, 1990). Although the BP algorithm shares a great convenience of being very simple, early works argue that problems with the BP algorithm are essentially its nature of being a gradient descent algorithm, cf. (Sutton, 1986). Since a cost function for training an FNN is often in a large scale and highly non-convex, BP algorithms often suffer from two major problems, namely, (i) potential existence of undesired local minima, and (ii) slow convergence speed. Although BP algorithms are suspected to be sensitive to initialisations, c.f. (Kolen & Pollack, 1990), recent results reported in (Goodfellow et al., 2015) suggest that modern FNN learning algorithms can overcome the problem of local optima quite conveniently. Such an observation could be explained by the previous works in (Yu, 1992; Yu & Chen, 1995; Gori & Tesi, 1992; Kawaguchi, 2016), which developed conditions on the structure of FNNs to eliminate undesired local minima. On the other hand, to deal with slow convergence speed, various modified BP algorithms have been developed, such as momentum based BP algorithm (Vogl et al., 1988), conjugate gradient algorithm (Charalambous, 1992), and BFGS algorithm (Le et al., 2011). Heuristic approximations of the Hessian matrix, such as a diagonal approximation structure (Battiti, 1992) and a block diagonal approximation structure (Wang & Lin, 1998), have also been proposed to construct approximate Newton's method. However, without a true evaluation of the Hessian, performance of these heuristic approximations is hardly convincing.

2 REVISITING THE BACKPROPAGATION ALGORITHM

We denote by L the number of layers in an FNN structure, and by n_l the number of processing units in the l-th layer with $l = 1, \ldots, L$. Specifically, by letting $l = 0$, we refer to the input layer. Let $\phi_{l-1} \in \mathbb{R}^{m_l}$ denote the output from the $(l-1)$-th layer, $w_{l,k} \in \mathbb{R}^{m_l}$ the parameter vector associated with the (l,k)-th unit function $f_{l,k}(w_{l,k}, \phi_{l-1}) \in \mathbb{R}$ in the l-th layer. By stacking all unit functions together, we can define the l-th layer evaluation mapping as

$$F_l \colon \mathbb{R}^{m_l \times n_l} \times \mathbb{R}^{m_l} \to \mathbb{R}^{n_l}, \qquad (W_l, \phi_{l-1}) \mapsto \left[f_{l,1}(w_{l,1}, \phi_{l-1}), \ldots, f_{l,n_l}(w_{l,n_l}, \phi_{l-1})\right]^\top, \quad (1)$$

with $W_l := [w_{l,1}, \ldots, w_{l,n_l}] \in \mathbb{R}^{m_l \times n_l}$ being the l-th parameter matrix. Specifically, let us denote by $\phi_0 \in \mathbb{R}^{n_0}$ the input, then we define $\phi_l := F_l(W_l, \phi_{l-1})$ iteratively. By composing all the layer-

wise mappings together, the overall network mapping is defined as

$$F\colon \mathcal{W}\times\mathbb{R}^{n_0}\to\mathbb{R}^{n_L},\qquad (\mathbf{W},\phi_0)\mapsto F_L(W_L,\cdot)\circ\ldots\circ F_2(W_2,\cdot)\circ F_1(W_1,\phi_0), \tag{2}$$

where $\mathbf{W}:=(W_1,\ldots,W_L)\in\mathcal{W}:=\mathbb{R}^{m_1\times n_1}\times\ldots\times\mathbb{R}^{m_L\times n_L}$. For a specific learning task, one often deploys a suitable error function $E\colon\mathbb{R}^{n_L}\to\mathbb{R}$, which is assumed to be differentiable in this abstract. For supervised learning tasks, given a dataset with T samples, denoted by $(x_i,y_i)_{i=1}^T$, we define the overall *FNN learning cost function* as

$$\mathcal{J}\colon\mathcal{W}\to\mathbb{R},\qquad \mathcal{J}(\mathbf{W}):=\sum_{i=0}^{T}(E\circ F)\,(\mathbf{W},x_i), \tag{3}$$

which is by construction differentiable in FNN parameters $\mathbf{W}$.

We apply the chain rule of multivariable derivative to compute the first derivation of $(E\circ F)$ w.r.t. the l-th parameter matrix W_l in direction $H_l\in\mathbb{R}^{m_l\times n_l}$ evaluated at sample (x_i,y_i) as

$$\mathrm{D}(E\circ F)(W_l)H_l=\mathrm{D}E(\phi_L^{(i)})\cdot\mathrm{D}_2F_L(W_L,\phi_{L-1}^{(i)})\cdot\ldots\cdot\mathrm{D}_2F_{l+1}(W_{l+1},\phi_l^{(i)})\cdot\mathrm{D}_1F_l(W_l,\phi_{l-1}^{(i)})H_l, \tag{4}$$

where $\mathrm{D}_1F_l(W_l,\phi_{l-1}^{(i)})$ and $\mathrm{D}_2F_l(W_l,\phi_{l-1}^{(i)})$ refer to the derivative of F_l w.r.t. the first and second argument, respectively. Let $\phi_l'\in\mathbb{R}^{n_l}$ be the vector of the derivative of the activation function in the l-th layer, and we define diagonal matrices $\Sigma_l^{'(i)}:=\mathrm{diag}(\phi_l^{'(i)})$ for all $l=1,\ldots,L$. Then we can write the gradient of $(E\circ F)$ with respect the l-th parameter matrix $W_l\in\mathbb{R}^{m_l\times n_l}$ as

$$\nabla_{(E\circ F)}(W_l)=\phi_{l-1}^{(i)}\big(\underbrace{\Sigma_l^{'(i)}W_{l+1}\ldots\Sigma_{L-1}^{'(i)}W_L\Sigma_L^{'(i)}\nabla E(\phi_L^{(i)})}_{=:\omega_l^{(i)}\in\mathbb{R}^{n_l}}\big)^\top, \tag{5}$$

which is a rank-one matrix update. By exploring the layer-wise structure of the FNN, the corresponding vector $\omega_l^{(i)}$ can be computed iteratively backwards from the output layer L. Such a backward mechanism in computing the gradient $\nabla_{(E\circ F)}(W_l)$ is referred to as the classic BP algorithm.

3 Main Results

With the gradient computed explicitly as in (5), the critical points of the FNN learning cost $\mathcal{J}$ are characterised by simply setting it to zero, namely, $\nabla_{\mathcal{J}}(\mathbf{W})=0$. More explicitly, by constructing a sequence of matrices as, with $\Psi_L^{(i)}=\Sigma_L^{'(i)}\in\mathbb{R}^{n_L\times n_L}$ and for all $l=L-1,\ldots,1$ as

$$\Psi_l^{(i)}:=\Sigma_l^{(i)}W_{l+1}\Psi_{l+1}^{(i)}\in\mathbb{R}^{m_l\times n_L}, \tag{6}$$

we can write the critical point condition explicitly as an equation system in $\nabla_E(\phi_L^{(i)})$ as

$$\nabla_{\mathcal{J}}(\mathbf{W})=\underbrace{\begin{bmatrix}\Psi_L^{(1)}\otimes\phi_{L-1}^{(1)} & \ldots & \Psi_L^{(T)}\otimes\phi_{L-l}^{(T)}\\ \vdots & \ddots & \vdots\\ \Psi_1^{(1)}\otimes\phi_0^{(1)} & \ldots & \Psi_1^{(T)}\otimes\phi_0^{(T)}\end{bmatrix}}_{=:\mathbf{P}\in\mathbb{R}^{N_{net}\times(T\cdot n_L)}}\begin{bmatrix}\nabla_E(\phi_L^{(1)})\\ \vdots\\ \nabla_E(\phi_L^{(T)})\end{bmatrix}=0, \tag{7}$$

where $\otimes$ denotes the Kronecker product of matrices, and $N_{net}=\sum_{l=1}^{L}m_l\cdot n_l$ is the total number of variables in an FNN. Obviously, if the rank of matrix $\mathbf{P}$ is equal to $T\cdot n_L$, then the trivial solution of $\nabla_E(\phi_L^{(i)})=0$ for all $i=1,\ldots,T$ is the only solution of the parameterised linear system (7). If the error function E is chosen to be strictly convex, then such a trivial zero solution is corresponding to the global minimum of E. Hence, we present the following theorem.

Theorem 1 (Local minima free condition)**.** *Let the error function $E\colon\mathbb{R}^{n_L}\to\mathbb{R}$ be strictly convex, and a global minimum $\mathbf{W}^*$ of the FNN learning cost be reachable. If the rank of matrix $\mathbf{P}$ as constructed in* (7) *is equal to $T\cdot n_L$, i.e., $\mathrm{rank}(\mathbf{P})=T\cdot n_L$, then the FNN learning cost function $\mathcal{J}$ is free of local minima.*

Remark 1 (Choice of the number of NN variables)**.** *Given the number of rows of $\mathbf{P}$ being N_{net}, the theorem suggests that the total number of variables in an FNN, i.e., N_{net}, needs to be greater than or equal to $T\cdot n_L$.*

The analysis of the Hessian is critically important for designing efficient numerical algorithms. The Hessian form of the FNN learning cost function $\mathcal{J}$ is a bilinear operator $\mathsf{H}_{\mathcal{J}}\colon\mathbb{R}^{N_{net}}\times\mathbb{R}^{N_{net}}\to\mathbb{R}$, computed by computing the second derivative of $\mathcal{J}$. Specifically, if a global minimum $\mathbf{W}^*$ is reachable, the Hessian form $\mathsf{H}_{\mathcal{J}}$ evaluated at $\mathbf{W}^*$ in direction $\mathbf{H}\in\mathcal{W}$ is computed by

$$\mathsf{H}_{\mathcal{J}}(\mathbf{W}^*) = \tfrac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{J}(\mathbf{W}+t\mathbf{H})\big|_{t=0} = \sum_{i=1}^{T} \mathbf{\Psi}^{*(i)} \odot \mathbf{\Phi}^{*(i)} \in \mathbb{R}^{N_{net}\times N_{net}}, \tag{8}$$

where $\odot$ is the Khatri-Rao product of two identically partitioned $(L \times L)$ matrices $\mathbf{\Psi}^{*(i)}$ and $\mathbf{\Phi}^{*(i)}$

$$\mathbf{\Psi}^{*(i)} := \begin{bmatrix} \Psi_L^{(i)} \\ \vdots \\ \Psi_1^{(i)} \end{bmatrix} \mathsf{H}_E(\phi_L^{*(i)}) \begin{bmatrix} \Psi_L^{(i)} \\ \vdots \\ \Psi_1^{(i)} \end{bmatrix}^\top, \quad \text{and} \quad \mathbf{\Phi}^{*(i)} := \begin{bmatrix} \phi_{L-1}^{*(i)} \\ \vdots \\ \phi_0^{*(i)} \end{bmatrix} \begin{bmatrix} \phi_{L-1}^{*(i)} \\ \vdots \\ \phi_0^{*(i)} \end{bmatrix}^\top. \tag{9}$$

It is obvious that $\mathrm{rank}(\mathbf{\Psi}^{*(i)}) \leq n_L$ and $\mathrm{rank}(\mathbf{\Phi}^{*(i)}) = 1$. Since both matrices $\mathbf{\Psi}^{*(i)}$ and $\mathbf{\Phi}^{*(i)}$ are positive semi-definite, the Hessian matrix $\mathsf{H}_{\mathcal{J}}(\mathbf{W}^*)$ is simply a sum of T low rank ($\leq n_L$) positive semi-definite matrices. We can conclude the following result.

Theorem 2. *If a global minimum $\mathbf{W}^*$ of the FNN learning cost is reachable, then the rank of the Hessian matrix of $\mathcal{J}$ is bounded from above by*

$$\mathrm{rank}(\mathsf{H}_{\mathcal{J}}(\mathbf{W}^*)) \leq T \cdot n_L. \tag{10}$$

Remark 2. *According to the result in Theorem 1, it is easy to that* $\mathrm{rank}(\mathsf{H}_{\mathcal{J}}(\mathbf{W}^*)) \leq T \cdot n_L \leq N_{net}$. *In other words, the rank of the Hessian at the global minima have the largest possible rank of $T \cdot n_L$. When a specific FNN is constructed from scratch without insightful knowledge regarding the data, then it is very likely that the Hessian is degenerate, i.e., gradient based algorithms can suffer significantly from slow convergence speed.*

It is important to notice that the Hessian $\mathsf{H}_{\mathcal{J}}(\mathbf{W}^*)$ is neither diagonal nor block diagonal, which demotivates the existing approximate strategies of the Hessian in (Battiti, 1992; Wang & Lin, 1998). With our explicit characterisation of the Hessian at global minima, we propose to approximate the Hessian of J at arbitrary point $\mathbf{W}$ with the structure as shown in Eq. (8).

4 Numerical Experiments

We investigate performance of our proposed approximate Newton's (AN) algorithm on the four regions classification benchmark, as originally proposed in (Singhal & Wu, 1989). In $\mathbb{R}^2$ around the origin, we have a square area $(-4, 4) \times (-4, 4)$, and three concentric circles with their radiuses being 1, 2, and 3. Four regions/classes are interlocked, nonconvex, as shown in Figure 1 (left). We draw randomly $T = 1000$ samples in the box for training, and specify the corresponding output to be the i-th basis vector in $\mathbb{R}^4$. We deploy an FNN architecture with two hidden layers, i.e., $L = 3$. In both hidden layer, there are 10 units each. Hence, we have $n_0 = 2$, $n_1 = n_2 = 10$, and $n_3 = 4$. All activation functions are chosen to be Sigmoid. Finally, the error function is an smooth approximation of the ℓ_1 norm as $E(x) := \sqrt{\|x - y\|_2^2 + \beta}$ where we set $\beta = 10^{-6}$. We test both the classic BP algorithm and the AN algorithm. For running 1000 iterations, the BP algorithm took 61.1 sec., while the AN algorithm spent 1314.1 sec. On average, the running time for each iteration of AN was about 21.4 times as required for an iteration of BP. With the same data and the same random initialisation, we ran BP for 20760 iterations, which took the same amount of time as required for 1000 iterations of AN. As in Figure 1 (right), the first 1000 iterations of BP was highlighted in *red* with the remaining iterations being coloured in *blue*. The AN went up at the beginning, then smoothly decreased to the global minimal value, while the BP demonstrated strong oscillation towards the end. It is worth noticing that the prediction of the trained neural network matches exactly the label in the four region problem as the global minimum was reached.

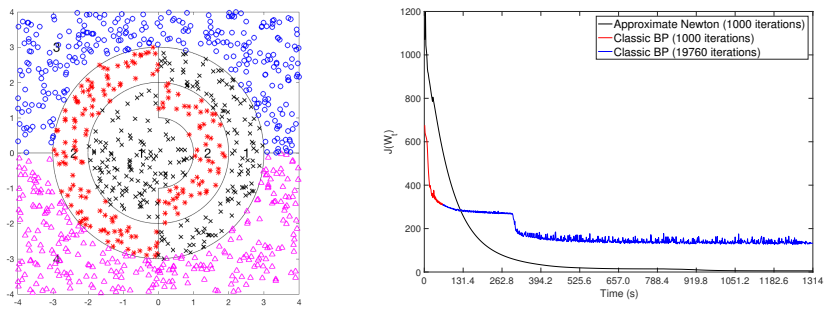

Figure 1: Comparison of convergence in terms of cost function value (step size $\alpha = 0.01$).

REFERENCES

R. Battiti. First- and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4(2):141–166, 1992.

C. Charalambous. Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proceedings G - Circuits, Devices and Systems*, 139(3):301–310, 1992.

T. Falas and A. G. Stafylopatis. The impact of the error function selection in neural network-based classifiers. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 3, pp. 1799–1804, 1999.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, volume 9, pp. 249–256, 2010.

I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. Published at the 5^{th} International Conference on Learning Representations (ICLR). arXiv:1412.6544., 2015.

M. Gori and A. Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257, 1991.

K. Kawaguchi. Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 586–594. 2016.

J. F. Kolen and J. B. Pollack. Backpropagation is sensitive to initial conditions. *Complex Systems*, 4 (3):269–280, 1990.

Q. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Ng. On optimization methods for deep learning. Proceedings of international conference on Machine Learning, 2011.

H. N. Mhaskar and C. A. Micchelli. How to choose an activation function. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, pp. 319–326, 1993.

S. Singhal and L. Wu. Training multilayer perceptrons with the extended Kalman algorithm. In *Advances in Neural Information Processing Systems*, pp. 133–140, 1989.

S. Sun, W. Chen, L. Wang, X. Liu, and T.-Y. Liu. On the depth of deep neural networks: A theoretical view. In *Proceedings of the* 31^{st} *AAAI Conference on Artificial Intelligence*, pp. 2066–2072, 2016.

R. S. Sutton. Two problems with backpropagation and other steepest-descent learning procedures for networks. In *Proceedings of the* 8*-th Annual Conference of the Cognitive Science Society*, pp. 823–831, 1986.

T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59(4):257–263, 1988.

Y.-J. Wang and C.-T. Lin. A second-order learning algorithm for multilayer networks based on block Hessian matrix. *Neural Networks*, 11(9):1607–1622, 1998.

B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

X.-H. Yu. Can backpropagation error surface not have local minima. *IEEE Transactions on Neural Networks*, 3(6):1019–1021, 1992.

X.-H. Yu and Guo-An Chen. On the local minima free condition of backpropagation learning. *IEEE Transactions on Neural Networks*, 6(5):1300–1303, 1995.