F-026

微分可能な活性化関数に基づく誤差逆伝播法の拡張
Extended Back-propagation Based on a Differentiable Activation Function

宮内 敦†
Atsushi Miyauchi

1. はじめに

最近の多層ニューラルネットワークでは活性化関数として ReLU を用いることが多い。しかし、この関数は原点で微分値が定義できず導関数は不連続となる。学習過程において誤差逆伝播法を用いる場合、この不連続性によって信号の流れが滑らかに進まず収束過程に悪影響を与える可能性が考えられる。微分可能な ReLU はこれまでもいくつか考案されているが、本稿ではより簡単な構成方法を提案し、加えてそれを用いた誤差逆伝播法の拡張方法を示す。

2. 微分可能な活性化関数

活性化関数には従来 sigumoid のような有界かつ微分可能な関数が用いられてきた。しかし多層ネットワークにおいて勾配消失による学習効果の飽和が明らかとなり、現在では非有界関数の一つである ReLU[1]の使用が主流となっている。他方、誤差逆伝播法[2]は深層学習の代表的手法である CNN や RNN おいても学習手段として頻繁に用いられるが、そのアルゴリズムには活性化関数の導関数が必要である。ReLU は原点で微分値が不定であるうえに不連続にその値が変化する。中間的な値が存在しないために学習の滑らかな収束過程が阻害され、処理時間が長大化する懸念がある。共役勾配法による大規模行列の緩和解法においても、丸め誤差によって残差が不規則に変動する現象はしばしば経験される[3]。そこで本稿では ReLU にパラメータを導入することによって微分可能な近似関数を提案する。これまでにも同様の試みはいくつか存在するが、本手法はより簡単な式を用い、かつパラメータによる制御が可能なことが特徴である。はじめに ReLU を以下にように変形する。

$$ReLU(x) \equiv \max(0,x) = \frac{1}{2}(x+|x|) = \frac{1}{2}\left(x+\sqrt{x^2}\right)$$

ここで上式にパラメータ ε を導入し、新たな活性化関数 f を次式で定義する。

$$f(x,\varepsilon) = \frac{1}{2}\left(x+\sqrt{x^2+4\varepsilon}\right)$$

この関数は明らかに $\varepsilon \to 0$ の極限において ReLU に一致し、任意の x と $\varepsilon > 0$ に対して微分可能な滑らかな関数である。図 1 は f とその導関数 f' を $\varepsilon = 0$ と $\varepsilon - 1$ に対してプロットしたグラフである。上記の定義式を少し変形すると、逆関数が次にように簡単な式で与えられる。

$$x(f,\varepsilon) = f - \frac{\varepsilon}{f}$$

この逆関数を用いることにより、導関数 f_ε' とパラメータ微分 $\dot{f}_\varepsilon$ もそれぞれ以下のような簡単な式で表される。

$$f'(x,\varepsilon) \equiv \frac{\partial f}{\partial x} = \frac{f}{f+\varepsilon/f}, \qquad \dot{f}(x,\varepsilon) \equiv \frac{\partial f}{\partial \varepsilon} = \frac{1}{f+\varepsilon/f}$$

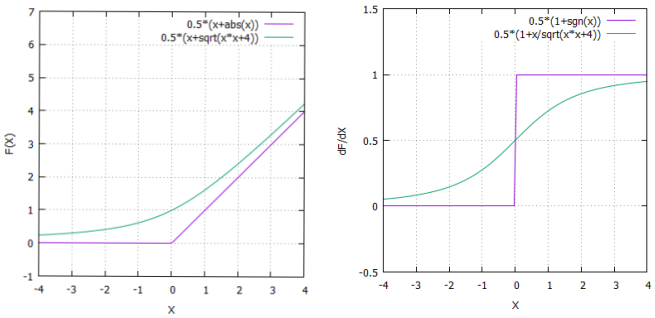

図 1　f とその導関数

3. 誤差逆伝播法の拡張

新たに導入したパラメータを誤差関数の最小化に利用することでアルゴリズムを拡張することが可能になる。ここでは一般的な多層ネットワークを考える。ここで各記号の定義は図 2 に示した通りである。バイアス $b_i^{(l)}$ は定出力ユニットの係数として $w_{ji}^{(l)}$ に含める。また微分可能化のために導入したパラメータを $\varepsilon_i^{(l)}$ とする。ここでサンプル x_n に対する誤差関数を E_n とすると、勾配降下法による学習は一般的に次式で与えられる。

$$w_{ji}^{(l)} \Leftarrow w_{ji}^{(l)} - \xi\frac{\partial E_n}{\partial w_{ji}^{(l)}}, \qquad \varepsilon_i^{(l)} \Leftarrow \varepsilon_i^{(l)} - \eta\frac{\partial E_n}{\partial \varepsilon_i^{(l)}}$$

ここで ξ, η は発散しない程度に設定する任意の小さな定数である。通常の勾配降下法では左側の重み係数 $w_{ji}^{(l)}$ に関する学習のみでアルゴリズムは閉じている。一方、本手法ではこれに右側の新たな導入したパラメータに関する学習が加わる。上式中に現れる偏微分は誤差逆伝播法と呼ばれるアルゴリズムによって効率よく計算できることが知られている。ここではその詳細を述べないが、重み関数の偏微分は結果的に次式で与えられる。

$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = \delta_j^{(l)} z_i^{(l-1)}, \qquad \delta_j^{(l)} = \sum\nolimits_k \delta_k^{(l+1)} w_{kj}^{(l+1)} f'(u_j^{(l)}, \varepsilon_j^{(l)})$$

右側の式は漸化式となっており、出力側 $\delta_i^{(L)}$ から入力側 $\delta_i^{(0)}$ へ向かって逐次計算される。また、初期値となる出力側の値はそれぞれ以下で与えられる。

† 一般財団法人高度情報科学技術研究機構, RIST

$$\delta_j^{(L)} = \left(z_j^{(L)} - d_j\right) f'\left(u_j^{(L)}, \varepsilon_j^{(L)}\right)$$

ここで d_j は所望する出力値である。同じアルゴリズムをパラメータ微分にも適用すると次式を得る。

$$\frac{\partial E_n}{\partial \varepsilon_j^{(l)}} = \sigma_j^{(l)}, \qquad \sigma_j^{(l)} = \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} \dot{f}(u_j^{(l)}, \varepsilon_j^{(l)})$$

また出力側の値は以下で与えられる。

$$\sigma_j^{(L)} = \left(z_j^{(L)} - d_j\right) \dot{f}\left(u_j^{(L)}, \varepsilon_j^{(L)}\right)$$

さて、上記の学習を繰り返し適用した場合に$\varepsilon_i^{(l)}$ が負値となる可能性がある。そこで正値性を保証するために変数変換 $\varepsilon_i^{(l)} = \exp\theta_i^{(l)}$ を行い、$\varepsilon_i^{(l)}$ に代えて $\theta_i^{(l)}$ を学習させる。

$$\theta_i^{(l)} \Leftarrow \theta_i^{(l)} - \frac{\eta}{\varepsilon_i^{(l)}} \frac{\partial E_n}{\partial \varepsilon_i^{(l)}}$$

以上に述べた方法は、従来の誤差逆伝播法を内包して拡張する形になっており、既存のプログラムに若干の追加をするだけで新たな制御自由度を増やすことが可能である。

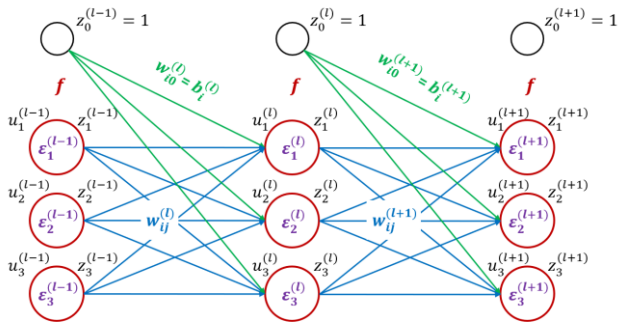

図 2　多層ネットワークの構成例

4. おわりに

多層ネットワークを実用的な問題に適用する場合、正規化、ドロップアウト、モメンタム、AdaGrad といった経験的あるいは対症療法的なテクニックを要求されることが少なくない[4]。新たなパラメータの導入によって学習の自由度を増す本稿の方法により、これらの効果をある程度自動的に取り込むことができれば応用面において有益な場面もあると考えられる。今後はベンチマーク問題に適用してその有効性について検証を進めたい。

参考文献

[1] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks”, Proc. 14th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), **15**, 315-323, (2011).
[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-propagating Errors”, Nature **323**(6088), 533-536, (1986).
[3] H. A. van der Vorst, “Iterative Krylov Methods for Large Linear Systems”, Cambridge University Press, (2003).
[4] 人工知能学会監修, “深層学習”, 近代科学社, (2015)