

誤差逆伝播法 (Backpropagation)

目次

1	概要 (abstract)	1
2	アルゴリズム (algorithm)	1

1 概要 (abstract)

誤差逆伝播法 (Backpropagation) はフィードフォワード型のニューラルネットワークの学習に用いられる方法である。確率的勾配法 (stochastic gradient descent) と呼ばれる方法の一つであり、広い意味では最小値を探索するアルゴリズムである。ニューラルネットワークの学習は、損失関数をパラメータの関数と考えたときに、損失関数を目的関数とした最小化問題であるといえることができる。最小化問題としてのニューラルネットワークに対して確率的勾配法を適用したのが誤差逆伝播法といえることができる。

2 アルゴリズム (algorithm)

まず使用するニューラルネットワークを定義する。対象とするニューラルネットワークの層数を N とする。 k ($k = 1, 2, \dots, N$) 層目のニューロン k ($i = 1, 2, \dots, n_k$) の出力値を $x_i^{(k)}$ 、 $k + 1$ 層目のニューロンの計算に用いられるパラメータを $w_i^{(k)}$ $i = 1, 2, \dots, m_k$ とする。ここで n_k は k 層目のニューロン数、 m_k は $k + 1$ 層目のニューロンの出力値の計算に用いられるパラメータの数である。また、 $k + 1$ 層目のニューロン i の出力値を計算する関数を $f_i^{(k)}$ とする。これらによって順伝播の計算は

$$x_i^{(k+1)} = f_i^{(k)}(x^{(k)}, w^{(k)})$$

として書かれる。また、損失関数を L とすると、 p 個目の教師パターン $t_i^{(p)}$ ($i = 1, 2, \dots, n_N$) とネットワークの出力 $x_i^{(N)}$ との誤差は

$$L(t^{(p)}, x^{(N)})$$

となる。よって教師パターン $t^{(p)}$ に対する各重みの更新量 $\Delta w_i^{(k)}$ は

$$\Delta w_i^{(k)} = -\eta \frac{\partial L}{\partial w_i^{(k)}}$$

として書かれる。更新量は連鎖律によって

$$\Delta w_i^{(k)} = -\eta \partial^{(p)} x_j^{(k)} \partial^{(p)} w_i^{(k)}$$

と変形できる。ここで

$$\begin{aligned} \partial^{(p)} x_i^{(k)} &= \begin{cases} \frac{\partial L}{\partial x_i^{(N)}} = \partial_{x_i} L(t^{(p)}, x^{(N)}), & k = N \\ \frac{\partial L}{\partial x_j^{(k+1)}} \frac{\partial x_j^{(k+1)}}{\partial x_i^{(k)}} = \partial^{(p)} x_j^{(k+1)} \cdot \partial_{x_i} f_j^{(k)}(t^{(p)}, x^{(N)}), & \text{otherwise} \end{cases} \\ \partial^{(p)} w_i^{(k)} &= \partial^{(p)} x_j^{(k+1)} \cdot \partial_{w_i} f_j^{(k)}(t^{(p)}, x^{(N)}) \\ \partial_{x_i} f_j^{(k)} &= \frac{\partial f_j^{(k)}}{\partial x_i^{(k)}} = \frac{\partial x_j^{(k+1)}}{\partial x_i^{(k)}} \\ \partial_{w_i} f_j^{(k)} &= \frac{\partial f_j^{(k)}}{\partial w_i^{(k)}} = \frac{\partial x_j^{(k+1)}}{\partial w_i^{(k)}} \end{aligned}$$

である。

誤差逆伝播法の具体的な手順は以下の通りである。

Step1 教師パターンの組 $(x'^{(p)}, t^{(p)})$ を P 個用意する。学習率 η を定める。

Step2 ある p に対して

$$x_i^{(1)} = x_i'^{(p)}$$

とし、 k を 2 から N まで増やしながら

$$x_i^{(k)} = f_i^{(k-1)}(x^{(k-1)}, w^{(k-1)})$$

を計算する。

Step3 ネットワークの出力と教師パターンとの誤差から

$$\partial^{(p)} x_i^{(N)} = \partial_{x_i} L(t^{(p)}, x^{(N)})$$

を計算し、 k を $N - 1$ から 1 まで減らしながら

$$\begin{aligned}\partial^{(p)} x_i^{(k)} &= \partial^{(p)} x_j^{(k+1)} \cdot \partial_{x_i} f_j^{(k)}(t^{(p)}, x^{(N)}) \\ \partial^{(p)} w_i^{(k)} &= \partial^{(p)} x_j^{(k+1)} \cdot \partial_{w_i} f_j^{(k)}(t^{(p)}, x^{(N)})\end{aligned}$$

を計算する。

Step4 いくつかの p に対して **Step2** から **Step3** を繰り返し行い、重みの更新量

$$\Delta w_i^{(k)} = -\eta \sum_p \partial^{(p)} w_i^{(k)}$$

を計算し重みを更新する。

Step5 繰り返し回数の上限を越えた場合やネットワークの出力と教師パターンの誤差

$$E = \sum_p L(t^{(p)}, x^{(N)})$$

が一定の値以下になった場合に終了する。そうでない場合 **Step2** へ戻る。