

目次

第 1 章	貢献度分配問題とニューラルネットワーク	3
1.1	再帰型ニューラルネットワークと経時的貢献度分配問題	3
1.2	勾配法に基づく経時的貢献度分配	4
1.2.1	RNN の構造と損失関数	4
1.2.2	未来方向・過去方向の勾配和	4
参考文献		8

第 1 章

貢献度分配問題とニューラルネットワーク

1.1 再帰型ニューラルネットワークと経時的貢献度分配問題

本節では再帰型ニューラルネットワーク (recurrent neural networks; RNN), すなわち再帰的結合を持った発火率モデルの学習則について取り扱う. 前節までで扱った前方向結合のみのニューラルネットワークでは通常, 入力と出力の間の遅延は考慮せず, 即時的に学習することを想定する. しかし RNN や運動制御, 強化学習などの動的な系では, シナプス結合, 神経活動や行動の変化が, 誤差や報酬という形で観測されるまでに時間的な遅れを伴う場合がある. このような状況において, 「ある時点のシナプス結合, 神経活動, 行動等の変化が, 後になって得られる誤差や報酬にどれだけ寄与したか」を明らかにし, 各変化に対して貢献度を適切に割り当てる問題を経時的貢献度分配問題 (temporal credit assignment problem; TCAP) と呼ぶ.

勾配法に基づいて経時的貢献度分配をする代表的な手法として, 実時間再帰学習 (real-time recurrent learning; RTRL) と経時的誤差逆伝播法 (backpropagation through time; BPTT) の 2 種類がある. 本節ではまず, RTRL と BPTT を勾配和の方向に基づいた統合的な視点から説明し, なぜ勾配法から 2 種類の学習則が得られるのかについて説明する. 次に, RTRL と BPTT を用いてパラメータの勾配を具体的に計算する. 最後に, RTRL と BPTT を踏まえたうえで, 生理学的妥当性の高い学習則について説明を行う.

1.2 勾配法に基づく経時的貢献度分配

1.2.1 RNN の構造と損失関数

まず、本節で扱う RNN の定義を行う。時刻 t ($1 \leq t \leq T$)^{*1} における入力を $\mathbf{x}_t \in \mathbb{R}^n$ 、隠れ状態を $\mathbf{h}_t \in \mathbb{R}^d$ 、出力を $\mathbf{y}_t \in \mathbb{R}^m$ とすると、隠れ状態と出力は

$$\mathbf{u}_t = \mathbf{W}_{\text{rec}}\mathbf{h}_{t-1} + \mathbf{W}_{\text{in}}\mathbf{x}_t + \mathbf{b}_{\text{rec}} \quad (1.1)$$

$$\mathbf{h}_t = (1 - \alpha)\mathbf{h}_{t-1} + \alpha f(\mathbf{u}_t) \quad (1.2)$$

$$\mathbf{v}_t = \mathbf{W}_{\text{out}}\mathbf{h}_t + \mathbf{b}_{\text{out}} \quad (1.3)$$

$$\mathbf{y}_t = g(\mathbf{v}_t) \quad (1.4)$$

与えられる。ただし、 $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d \times n}$, $\mathbf{W}_{\text{rec}} \in \mathbb{R}^{d \times d}$, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{m \times d}$ はシナプス結合重み, $\mathbf{b}_{\text{rec}} \in \mathbb{R}^d$, $\mathbf{b}_{\text{out}} \in \mathbb{R}^m$ は定常項である。 $f(\cdot), g(\cdot)$ は活性化関数であり、入力ベクトルの要素ごとに作用する。 $\mathbf{u}_t, \mathbf{v}_t$ は膜電位に対応し、勾配を書き下す際に使用する。 $\alpha := \frac{1}{\tau}$ は状態の更新率（時定数 τ の逆数）であり^{*2}、 $\alpha = 1$ の場合は Elman ネットワーク (Elman, 1990) と同一である。なお、状態の初期値を $\mathbf{h}_0 = \mathbf{0}$ とする。時刻 t での教師信号を \mathbf{y}_t^* とすると、損失 \mathcal{L} は各時刻における損失 \mathcal{L}_t の和を取り、

$$\mathcal{L} = \sum_{t=1}^T \mathcal{L}_t(\mathbf{y}_t, \mathbf{y}_t^*) \quad (1.5)$$

として与えられる。

1.2.2 未来方向・過去方向の勾配和

この RNN を学習させる際の目標は、損失 \mathcal{L} を最小化するようにパラメータ θ を最適化することである。ここでは、 $\theta \in \{\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{rec}}, \mathbf{W}_{\text{out}}, \mathbf{b}_{\text{rec}}, \mathbf{b}_{\text{out}}\}$ である。

勾配法の観点では、損失のパラメータに対する勾配 $\frac{\partial \mathcal{L}}{\partial \theta}$ が求まれば最適化が可能である。損失は $\mathcal{L} = \sum_t \mathcal{L}_t$ と時間方向に分解できるので、勾配は

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \theta} = \sum_{t=1}^T \sum_{s=1}^T \frac{\partial \mathcal{L}_t}{\partial \theta_s} \underbrace{\frac{\partial \theta_s}{\partial \theta}}_{=\mathbf{I}} = \sum_{t=1}^T \sum_{s=1}^t \frac{\partial \mathcal{L}_t}{\partial \theta_s} \quad (1.6)$$

と時間方向に分解できる。ここで、 s, t はいずれも時刻を表し、 $1 \leq s, t \leq T$ である。また、便宜的に「時刻 s に用いられたパラメータ θ 」を θ_s と表記した。従って、 $\frac{\partial \mathcal{L}_t}{\partial \theta_s}$ は「時刻 t における損失 \mathcal{L}_t の時

*1 隠れ状態 \mathbf{h}_t についてのみ $t = 0$ を定義する。

*2 $\alpha < 1$ である RNN は、重み共有をした残差結合 (residual/skip connection) のある順伝播モデル (ResNet など) に展開することが可能である (Liao and Poggio, 2016)。

刻 s に用いられたパラメータ θ_s に対する勾配」を意味する。また、オンライン学習でパラメータを毎時刻更新する場合であっても、勾配計算においては θ_s の微小変化 $\delta\theta_s$ はそのまま現在の θ の微小変化 $\delta\theta$ に等しいと見なせるため、 $\frac{\partial\theta_s}{\partial\theta} = \mathbf{I}$ が成立する。さらに現在のパラメータの状態は過去の損失に影響を与えないため、 $s > t$ では $\frac{\partial\mathcal{L}_t}{\partial\theta_s} = \mathbf{0}$ となり、上式では $s \leq t$ の範囲の勾配のみが残っている。

ここで、なぜ BPTT と RTRL の 2 種類の学習法が存在するのかを考えると、 $\sum_t \sum_{s \leq t} \frac{\partial\mathcal{L}_t}{\partial\theta_s}$ という二重和において、どちらの変数に対して先に和を取るかに 2 通りの方法があるためである。すなわち、現在時刻を t または s のいずれかを基準にとるかによって、内側の和を未来方向 (future-facing) または過去方向 (past-facing) に進めることができ、これに応じて勾配の和の取り方も区別される (Marschall et al., 2020) :

$$\frac{\partial\mathcal{L}}{\partial\theta} = \begin{cases} \sum_{s=1}^T \frac{\partial\mathcal{L}}{\partial\theta_s} = \sum_{s=1}^T \sum_{t=s}^T \frac{\partial\mathcal{L}_t}{\partial\theta_s} & (\text{未来方向勾配和; e.g., BPTT}) \\ \sum_{t=1}^T \frac{\partial\mathcal{L}_t}{\partial\theta} = \sum_{t=1}^T \sum_{s=1}^t \frac{\partial\mathcal{L}_t}{\partial\theta_s} & (\text{過去方向勾配和; e.g., RTRL}) \end{cases} \quad (1.7)$$

未来方向勾配和では、「現在時刻 s におけるパラメータ θ_s が、未来のすべての損失 \mathcal{L}_t ($t \geq s$) に与える影響」を合算する。したがって、この形式では、「現在のパラメータの微小な変化が将来の損失にどれだけ影響を及ぼすか」を評価することになる。未来方向勾配和を利用する手法の代表例が BPTT である。一方、過去方向勾配和では、「現在時刻 t における損失 \mathcal{L}_t に対して、過去のすべてのパラメータ θ_s ($s \leq t$) が及ぼした影響」を合算する。すなわち、この形式では、「現在の損失が過去のパラメータの微小な変化にどれだけ応答するか」を評価することになる。過去方向勾配和を利用する手法の代表例が RTRL である。これらの勾配和についての概念図が図 1.1 である。

勾配和をいずれの方向で取る場合であっても、二重和をそのまま計算すれば、計算量は $\mathcal{O}(T^2)$ となり、非効率である。このため、動的計画法 (dynamic programming; DP) を用いて、計算量を $\mathcal{O}(T)$ に削減するのが一般的である。ここで、動的計画法とは、問題を部分問題に分割し、それらの部分問題の解を再利用することで、全体の計算量を削減するアルゴリズム設計手法である。誤差逆伝播法 (backpropagation) も動的計画法の一種に位置づけられ、また、運動制御や強化学習など、さまざまな分野で動的計画法は頻繁に用いられている。

なお、BPTT と RTRL をある程度ご存じの読者であれば、「BPTT は過去方向で RTRL は未来方向ではないのか」という疑問が浮かぶであろう。ここで「勾配和の評価において時間を進める方向」と「動的計画法において計算を進める方向」は逆になることに注意していただきたい。例えば、未来向き勾配和のように、将来に発生する損失や報酬を考慮して損失関数を評価する場合、動的計画法を適用すると、計算は未来から過去へと逆向きに進める必要がある。このように、将来の結果をもとに現在の値を推定する構図は、強化学習の TD 学習における状態価値の推定にも見られる。すなわち、状態価値を将来の累積

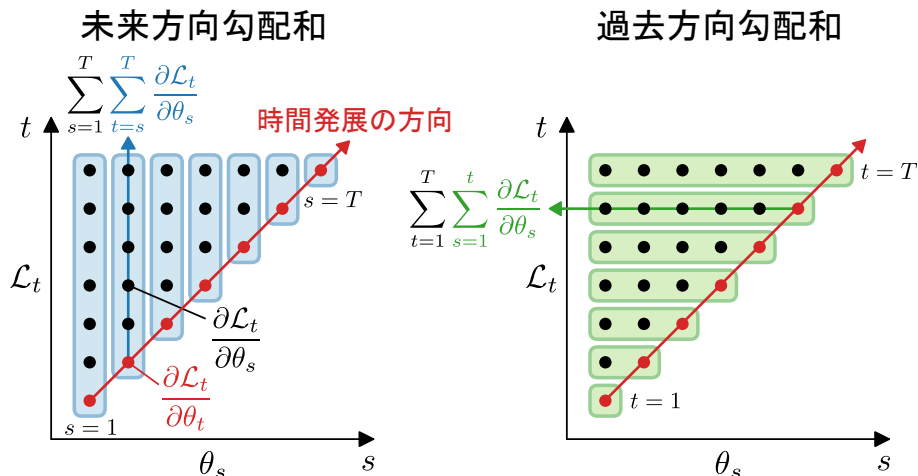


図 1.1 未来方向勾配和（左）と過去方向勾配和（右）の模式図．縦軸の t は損失関数 \mathcal{L} の時刻を表し，横軸の s はパラメータ θ の時刻を表す．モデルの時間発展は赤点に従って行われる．各点は $\frac{\partial \mathcal{L}_t}{\partial \theta_s}$ を意味する．対角線においては $s = t$ であるため，赤点は $\frac{\partial \mathcal{L}_t}{\partial \theta_t}$ を意味する．囲まれた点の集まりは，時刻 t において，どの勾配を考慮するか，ということの意味する．

報酬の期待値として定義した上で、その推定は未来から過去へと逆向きに進められる^{*3}。

以上を踏まえ、勾配和を動的計画法を用いて計算しよう。動的計画法を適用するためには、まず勾配を適切に展開し、現在の勾配を、ひとつ前（または後）の時刻における勾配との関係で再帰的に表現する必要がある。このとき留意すべき基本原則は、「現在の状態やパラメータは、過去の損失、状態、パラメータに影響を及ぼさない」という事実である。すなわち、現在の変数に対する過去の変数の勾配は常に 0 となる。これは例えば $\frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{h}_t} = \mathbf{0}$ が成り立つ。一方で、未来の変数は現在の変数に依存するため、現在の変数が未来の損失に及ぼす影響は意味を持つ。したがって、勾配を展開する際には、未来の損失に向かう方向で連鎖律を適用していくことになる。

^{*3} TD 学習を含め、強化学習は第 11 章で詳解する。

まず、未来方向勾配和の場合を考える。即時的なパラメータに対する損失の勾配は、

$$\frac{\partial \mathcal{L}}{\partial \theta_t} = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \theta_t} = \left(\sum_{s \geq t} \frac{\partial \mathcal{L}_s}{\partial \mathbf{h}_t} \right) \frac{\partial \mathbf{h}_t}{\partial \theta_t} \quad (1.8)$$

$$= \left(\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} + \sum_{s \geq t+1} \frac{\partial \mathcal{L}_s}{\partial \mathbf{h}_t} \right) \frac{\partial \mathbf{h}_t}{\partial \theta_t} \quad (1.9)$$

$$= \left(\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} + \sum_{s \geq t+1} \frac{\partial \mathcal{L}_s}{\partial \mathbf{h}_{t+1}} \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t} \right) \frac{\partial \mathbf{h}_t}{\partial \theta_t} \quad (1.10)$$

$$= \left(\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} + \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{t+1}} \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t} \right) \frac{\partial \mathbf{h}_t}{\partial \theta_t} \quad (1.11)$$

となる。ここで貢献度分配ベクトル (credit assignment vector) を $\delta_t := \frac{\partial \mathcal{L}}{\partial \mathbf{h}_t} \in \mathbb{R}^{1 \times d}$ 、即時的貢献度分配ベクトルを $\tilde{\delta}_t := \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \in \mathbb{R}^{1 \times d}$ 、状態遷移のヤコビ行列を $\mathbf{J}_t := \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \in \mathbb{R}^{d \times d}$ とすると、 δ_t は次のように未来から過去に向かう再帰的な関係式で表せる：

$$\delta_t = \tilde{\delta}_t + \delta_{t+1} \mathbf{J}_{t+1} \quad (1.12)$$

ただし、境界条件として $\delta_{T+1} = \mathbf{0}$ とする。この式を用いて、 δ_t を逐次的に求め、 $\frac{\partial \mathbf{h}_t}{\partial \theta_t}$ を即時的に計算して δ_t に乗じれば、 $\frac{\partial \mathcal{L}}{\partial \theta_t}$ が求まる。

次に過去方向勾配和の場合を考える。パラメータに対する即時的な損失の勾配は

$$\frac{\partial \mathcal{L}_t}{\partial \theta} = \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \theta} = \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \left(\sum_{s \leq t} \frac{\partial \mathbf{h}_t}{\partial \theta_s} \right) \quad (1.13)$$

$$= \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \left(\frac{\partial \mathbf{h}_t}{\partial \theta_t} + \sum_{s \leq t-1} \frac{\partial \mathbf{h}_t}{\partial \theta_s} \right) \quad (1.14)$$

$$= \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \left(\frac{\partial \mathbf{h}_t}{\partial \theta_t} + \sum_{s \leq t-1} \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \theta_s} \right) \quad (1.15)$$

$$= \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \left(\frac{\partial \mathbf{h}_t}{\partial \theta_t} + \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \theta} \right) \quad (1.16)$$

となる。ここで感度行列 (sensitivity matrix, influence matrix) を $\mathbf{P}_t := \frac{\partial \mathbf{h}_t}{\partial \theta} \in \mathbb{R}^{d \times |\theta|}$ 、即時的感度行列を $\tilde{\mathbf{P}}_t := \frac{\partial \mathbf{h}_t}{\partial \theta_t} \in \mathbb{R}^{d \times |\theta|}$ とする。ただし、 $|\theta|$ はパラメータの次元数を意味する。この場合、 \mathbf{P}_t は次のように過去から未来に向かう再帰的な関係式で表せる：

$$\mathbf{P}_t = \tilde{\mathbf{P}}_t + \mathbf{J}_t \mathbf{P}_{t-1} \quad (1.17)$$

ただし、境界条件として $\mathbf{P}_0 = \mathbf{0}$ とする。この式を用いて、 \mathbf{P}_t を逐次的に求め、 $\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t}$ を即時的に計算して \mathbf{P}_t に乗じれば、 $\frac{\partial \mathcal{L}_t}{\partial \theta}$ が求まる。

連鎖律に基づき理論的な勾配導出を行ったが、数値計算においては、誤差逆伝播法における処理と同様に、自動微分 (automatic differentiation, AD) の枠組みを適用して勾配を効率的に計算することができる。自動微分では、微分の累積過程に応じて2つのモードがあり、それぞれが勾配和の2つの方向と対応する。すなわち、未来方向に対する勾配和の計算は逆方向累積 (reverse accumulation) または逆モード (reverse-mode) 自動微分によって、過去方向に対する勾配和の計算は順方向累積 (forward accumulation) または順モード (forward-mode) 自動微分によって、それぞれ実行される。

本節の最後に、パラメータに対する損失勾配の展開を整理する。すなわち、未来方向 (BPTT に相当) および過去方向 (RTRL に相当) での勾配和は、それぞれ次のように表される。

$$\begin{aligned} \text{未来方向勾配和: } \frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{t=1}^T \frac{\partial \mathcal{L}}{\partial \theta_t} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \theta_t} = \sum_{t=1}^T \left(\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} + \frac{\partial \mathcal{L}}{\partial \mathbf{h}_{t+1}} \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t} \right) \frac{\partial \mathbf{h}_t}{\partial \theta_t} \\ \text{過去方向勾配和: } \frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \theta} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \theta} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \left(\frac{\partial \mathbf{h}_t}{\partial \theta_t} + \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \theta} \right) \end{aligned}$$

次節 (BPTT) および次々節 (RTRL) では、ここで導出した関係式に基づいて、各パラメータの勾配を具体的に計算する。

参考文献

- Elman, J. L. (1990). “Finding structure in time”. *Cognitive science* 14.2, pp. 179–211.
- Liao, Q. and Poggio, T. (2016). “Bridging the gaps between residual learning, recurrent neural networks and visual cortex”. *arXiv preprint arXiv:1604.03640*.
- Marschall, O., Cho, K., and Savin, C. (2020). “A unified framework of online learning algorithms for training recurrent neural networks”. *Journal of machine learning research* 21.135, pp. 1–34.