人工知能学事典 再校のお願い

次ページ以降の内容につきまして，著者校正をお願いいたします．

- 締切は 2017 年 2 月 21 日です．
- 修正できるのは**修正必須の誤り**だけです．**文章の推敲や追加はできません**．
- 校正が終わりましたら，修正がない場合は「修正なし」，修正がある場合は修正内容をメールでお知らせください．
- メールでは伝わりにくい修正指示は，この PDF を印刷して赤字を入れ，スキャンまたは写真撮影の上，メールに添付してください．
- **共著の項目は，代表者がまとめて**ご回答ください．

ご連絡事項

- 再校では，aidic ウェブシステム上での初校の結果を反映した原稿データを，最終的な**紙版の状態に近づけた組版状態**でご覧いただきます（初校の状態を確認するには，再校通知メール（この PDF を開いたメール）の下部に記載したリンクを開いてください）．
- 再校通知メールに個別の連絡を記載している場合があります．**再校通知メールの■本項目固有のご連絡■を必ずお読みください**．
- 組版は，**旧版の器に原稿データを流し込んで現段階でできる調整・補正を施した状態**です．
- 後の工程で，ページデザインやフォントサイズを変更して総ページ数を圧縮するため，版面はこのあと変化します．**配置・空きの不具合や，長い数式の飛び出しなど，組版上の見た目の問題はその後に修正しますので，個々のご指摘は不要です**．
- 内容的には，全項目の再校が済んだ後の**全体処理（術語の統一など）を経て変化**します．
- 索引 **は出るべきものが出ているか**を確認してください（**英語の大文字/小文字による索引重複は無視**）．**aidic ウェブシステムと紙版とでポリシーが異なるため，並び順や対訳・略語の表示方法などは後にまとめて調整・変更**します（稀に先行して直しているところもあります）．
- 項目タイトル部の項目番号は，現状どの項目も 1（例えば 15 章なら「15-1」）（コラムの場合，章番号も出ず「0-a」）と出ていますが問題ありません．一方，他項目参照の項目番号は，再校 PDF を生成した時点での aidic ウェブシステム上の項目ラベルになっています．以下に一覧があります．

  `http://www.gravel.co.jp/pdfs/aidic_itemlist.htm`
- aidic ウェブシステムによる紙版向けの TeX データで，**稀に数式や索引語句などの再現における事故**が発生しています．別途共立出版にて不整合のチェックを行いますが，お気づきの点がありましたらお知らせください．

校正終了の通知やお問合せは，再校通知メール（この PDF を開いたメール）への返信を使い，下記宛てお願いいたします．

株式会社グラベルロード
学術コンテンツ編集： 伊藤裕之，山田ひとみ
aidic@gravel.co.jp

7–1 リカレントニューラルネットワーク

Recurrent Neural Network

はじめに

リカレントニューラルネットワークとは，フィードバック結合（feedback connection）を有するネットワークである．ネットワーク内に再帰結合が存在すると，現時刻における情報を処理する際に過去の情報を再帰結合から受け取り利用する．これにより，リカレントニューラルネットワークは時系列情報を扱うことが可能となる．フィードバック結合の時間遅れを変動項と考えることで，種々の実際のニューロンの活動をモデル化することが可能である．リカレントニューラルネットワークは離散時間を仮定した場合と連続時間を仮定する場合とがあるが，ここでは主に前者を取り上げる．図 1 (a) はリカレントニューラルネットワークの簡単な例である．再帰結合行列を $\mathbf{W}$ とする．この時間発展を考えると，再帰結合行列を共有する多層ニューラルネットワーク（図 1 (b)）となる．

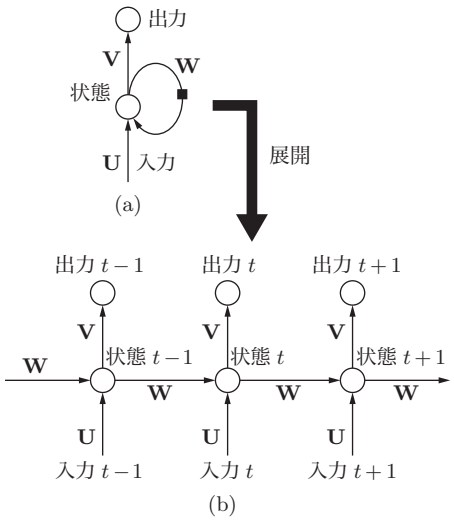

図 1 リカレントニューラルネットワーク (a) とその時間発展 (b)

後者は力学系（dynamical system）として記述される [1]．歴史的には，各時刻に単位時間幅の入力情報を与え，再帰結合を仮定しないモデルであるネットトーク（NETtalk）[2] が始まりである．時刻 $t+1$ におけるユニット i の出力 $x_i(t+1)$ は，バイアス項を無視すれば，x の生起確率を p として，

$$x_i(t+1) = \operatorname{argmax} p\left(x_i(t+1) \,|x_t(t), c(t); \theta\right) \quad (1)$$

$$= f_i\left(\sum_{j \in U} w_{ij} x_j(t)\right) \quad (2)$$

と表記される．ここで，f_i は任意の出力関数，w_{ij} は i, j 間の結合係数，$x_j(t)$ は y_i への入力信号である．$c(t)$ は文脈層と呼び，1 時刻前の状態を保持する．θ はニューラルネットワークを定めるパラメータ集合であり，結合係数やバイアス項を含む．

初期のモデルとしては，ジョーダンネットワーク（Jordan network）[3]，エルマンネットワーク（Elman network）[4] がある．両モデルとも 1 時刻前の状態を文脈層に保存し，通常のバックプロパゲーション法によって学習を行う．一方，時系列情報を一定の時間窓の期間維持し，時間窓内の情報について学習を行うアルゴリズムとして，BPTT (back-propagation through time) [5], RTRL (real time recurrent learning; 実時間リカレント学習) [6] が提案された．BPTT と RTRL は後の節で解説する．2010 年代以降，リカレントニューラルネットワークは時系列情報処理 [7] の発展として，チューリングマシン（Turing machine）との相同性 [8]，擬似プログラムコードの生成 [9]，英語からフランス語へ自動翻訳 [10]，写真からの脚注の生成 [11]，など，応用が盛んな領域となっている．類似した用語で表記されるモデルに，リカーシブニューラルネットワーク（recursive neural network）[12] があるが，これは自然言語処理に特化したモデルである．

単純再帰型ニューラルネットワーク

図 2 (a) にジョーダンネットワーク，(b) にエルマンネットワークを示す．

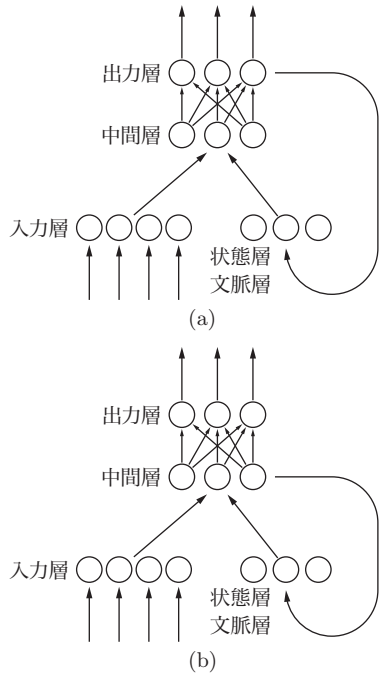

図 2　(a) ジョーダンネットワークと，(b) エルマンネットワーク

両者を併せて単純再帰型ニューラルネットワーク (simple recurrent neural network) と呼ぶ．ジョーダンネットワークは 1 時刻前の出力層の情報を，またエルマンネットワークは 1 時刻前の中間層の内容を保持しておき，現時刻における入力情報と文脈層情報は中間層で計算される．文脈層（context layer）とは，1 時刻前の状態を保持しておく層である．1 時刻前の状態をコピーするのであるから，ジョーダンネットワークにおける出力層ユニット数と文脈層ユニット数と同数であり，同様にエルマンネットワークの中間層ユニット数と文脈層ユニット数は同数である．ジョーダンネットワークの文脈層の信号が出力層へと帰還する図が描かれている文献が存在するが，原典のジョーダンネットワーク [3] では，出力層からの文脈層情報は中間層への帰還信号である．ジョーダンネットワークとエルマンネットワークの相違は，文脈情報として出力表象を用いるか，内部表象を用いるかの違いである．すなわち，1 時刻前の状態を利用する際に出力表象を用いたほうが制御に有用であるロボットアーム，姿勢，運動などの動作制御の場合にはジョーダンネットワークが用いられ，一方，1 時刻前の内部状態を扱ったほうが有利な言語情報処理，文法判断などではエルマンネットワークが用いられることが多かった．

学習については，ジョーダンネットワークとエルマンネットワークとも 1 時刻前の状態を文脈層にコピーするので，中間層から見ると入力情報が増えたこと以外に，再帰結合のない通常の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との間で相違はない．任意の目標関数を各結合係数で微分したバックプロパゲーションによる学習も行われる．

BPTT と RTRL

BPTT [13][6] は back-propagation through time の略，RTRL [5] は real time recurrent learning の略である．上記のジョーダンネットワークとエルマンネットワークの両モデルは，再帰（帰還信号）が出力層あるいは中間層から戻ってくることを区別したモデルである．この両者を区別せず，一般にリカレントニューラルネットワークを構成することを考える．このとき，学習について時間をどのように考慮するかについて BPTT と RTRL の二つのモデルが提案された．BPTT と RTRL とを統一して表記することも可能である [14]．RTRL は “real time” との命名から連続時間を考慮したモデルと誤解される場合もあるが，学習時に「即時」に学習できるという意味であり，離散時間を仮定したモデルである．一方，“through” を用いる BPTT は状態変動の履歴を全時刻について保持し，時刻を “through” すなわち串刺しにして計算するため，このような命名で区別されている．したがって，BPTT は学習時に全時刻の全状態を保持しておく必要がある．このため，扱う系列が長くなると，学習のために必要となる記憶容量や学習時間が問題となる．実際には，一定の時間幅で切断したり，過去の影響を時間幅に応じて減衰させたりする実装もある．

ある時刻における出力は，入力と 1 時刻前の状態に依存する．しかし，1 時刻前の状態は，そのときの入力とさらに 1 時刻前の状態とに依存するので RTRL も時間情報を扱いうる．換言すれば，学習時に考慮する時間幅を h とすると，RTRL は $\mathrm{BPTT}_{h=1}$ でもある．さらに，$\mathrm{BPTT}_{h=1}$，かつ，帰還信号の発生源に制約を設けたモデルは，ジョーダンネットワークやエルマンネットワークと見なしうる．時刻 t における出力層ユニット k の誤差を，d_k を教師信号として以下のように定義する．

$$e_k = d_k(t) - y_k(t) \tag{3}$$

ここで，$y_k(t)$ は出力信号を表す．システムの 2 乗誤差の総和 $J(t)$ を

$$J(t) = -\frac{1}{2}\sum_{i\in U}[e_i(t)]^2 \tag{4}$$

とすれば，時間幅 $(t', t]$ についての総誤差 $J^{\text{total}}(t', t)$ は，以下のように表記できる．

$$J^{\text{total}}(t', t) = \sum_{\tau=t'+1}^{t} J(\tau) \tag{5}$$

学習則は，学習係数を η として，

$$\nabla_w J^{\text{total}}(t', t) = -\sum_{\tau=t'+1}^{t} \nabla_w J(\tau) \tag{6}$$

$$\Delta w_{ij} = \eta \frac{\partial J^{\text{total}}(t', t)}{\partial w_{ij}} \tag{7}$$

と書くことができる．

BPTT

BPTT では過去の状態を記憶バッファに保持し，保持された記憶状態への結合についてもバックプロパゲーション法を適用して結合係数を更新する．時刻 t における誤差 $J(t)$ の勾配を計算するために

$$\delta_i(t) = f_i' \left(\sum_{j \in U} w_{ij} x_j \right) e_i(t) \tag{8}$$

$$e_i(t-1) = \sum_{j \in U} w_{ji} e_i(t) \tag{9}$$

とし，上式に従って時間を遡り $t_0 + 1$ まで達すれば

$$\frac{\partial J(t)}{\partial w_{ij}} = \sum_{\tau=t_0+1}^{t} e(\tau) x_j(\tau - 1) \tag{10}$$

として誤差を計算できる．上記をまとめると，

1. 現在の状態を入力とを履歴バッファに保存する
2. 現在の誤差をバックプロパゲーション法によって計算する
3. 全時刻における誤差の総和を計算する
4. 重みを更新する

を，全データで収束基準に達するまで繰り返すこととなる．BPTT の計算量のオーダーは $O(TN^2)$ となる．ここで，T は時間，N は中間層のユニット数である．BPTT は全時間について考慮するが，実際の計算においては時間幅 h を定める必要がある．これを BPTT_h と表記すれば，RTRL は BPTT の $h = 1$ なる特別な場合と見なしうる [14]．ジョーダンネットワークもエルマンネットワークも $\text{BPTT}_{h=1}$ に相当する．図 3 の実線矢印は情報の流れを，破線矢印は誤差の伝播を示している．各時刻において入出力は異なるが，各時刻における状態は保持され結合係数の更新のために用いられる．図中の数字は誤差の伝播されていく順番を示している．

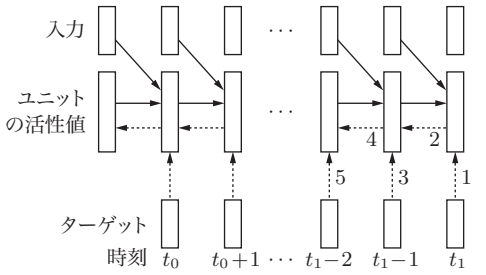

図 3 BPTT の概念図（文献 [6], p.448, Figure 4 を改変）

RTRL

BPTT は誤差勾配の計算に誤差情報の時間に関する逆伝播を用いた．これに対し，誤差勾配を順伝播させる方法が RTRL である．結合係数 w_{ij} に対するユニット k の影響を p_{ij}^k とし，すべてのユニット $(i, j, k \in U)$ について，

$$p_{ij}^k(t) = \frac{\partial y_k(t)}{\partial w_{ij}} \tag{11}$$

を定義する．時刻 t における誤差 $J(t)$ を各結合係数で微分した量を次式に従うと仮定する．

$$\frac{\partial J(t)}{\partial w_{ij}} = \sum_{k \in U} e_k(t) p_{ij}^k(t) \tag{12}$$

次の時刻の $p_{ij}^k(t+1)$ は，次式で表される．

$$p_{ij}^k(t+1) = f_k'\left(\sum(t+1)\right) \left[\sum_{l \in U} w_{kl} p_{ij}^t(t) + \delta_{ik} x_j(t) \right] \tag{13}$$

ここで，δ_{ij} はクロネッカーのデルタである．さらに，初期状態 t_0 においては，$p_{ij}^k(t_0) = \partial y_k(t_0) / \partial w_{ij} = 0$ と仮定すれば各時刻ステップにおける $p_{ij}^k(t)$ の量を計算可能である．この値と誤差との積によって誤差の勾配を求めることができる．この意味で，BPTT とは異なり RTRL は “即時” 的に計算可能である．

勾配消失問題と勾配爆発問題

バックプロパゲーション法における多層パーセプトロンの学習においては，勾配消失問題（gradient vanishing problem）により学習が進まないことが以前か

ら指摘されてきた [15]．Bengio ら [16] はこの問題を定式化した．

バックプロパゲーション法では，任意の課題における特定の出力層ユニットの誤差が下位層の全ユニットに伝播する．したがって，多層化されたニューラルネットワークで，活性化関数にシグモイド関数（$\sigma(x) = (1 + \exp -x)^{-1}$）を用いると，誤差関数を各結合係数で微分した値にシグモイド関数の微分が入る．

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x)) \tag{14}$$

$0 \leq \sigma(x) \leq 1$ であるので，$d\sigma(x)/dx$ は微分するごとに，すなわち層を下るごとに小さくなってしまう．これが勾配消失問題の一因である．

一方，リカレントニューラルネットワークにおける学習の際，BPTT の時間窓が広くなると勾配が発散する場合がある．最小化すべき損失関数 L の勾配をパラメータ θ で時間窓 t_w まで計算すると次式のようになる．

$$\frac{\partial L}{\partial \theta} = \sum_{1 \leq k \leq t} \frac{\partial L}{\partial x_t} \frac{\partial x_t}{\partial x_k} \frac{\partial x_k}{\partial \theta} \tag{15}$$

ここで，$\partial x_t/\partial x_k$ は中間層 h での時刻 t における時刻 k による微分である．階層が深くなると，$\partial h_t/\partial h_{t-1}$ のヤコビアン（Jacobian）行列式の最大特異値に応じて拡大縮小が起こる．特異値が 1 より小さければ勾配消失問題となり，1 より大きければ勾配爆発問題（gradient exploding problem）となる [16]．リカレントニューラルネットワークの学習においては，式 (15) で再帰的に勾配を計算した場合，ヤコビアン（Jacobian）の最大特異値分解（singular value decomposition; SVD）に応じて指数関数的に勾配が変化することになる．

勾配正規化

勾配消失問題を回避するために，次のような勾配正則化を行うことが提案されている．誤差関数 E のパラメータ x に関する勾配計算の際に，その 1 時刻後の微分量との比 $\partial x_{k+1}/\partial x_k$ を用いて，次式のような正規化を行う [17]．

$$\sum_k \left(\frac{\left\| \frac{\partial E}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial x_k} \right\|}{\left\| \frac{\partial E}{\partial_{k+1}} \right\|} - 1 \right)^2 \tag{16}$$

勾配クリップ

勾配爆発問題に対する対処には，勾配値を一定範囲に収める勾配クリップ（gradient clip）が提案されている．勾配クリップは，勾配爆発が局在した小領域でしか起こらないという仮定に基づき，一定以上の勾配値に達した場合，強制的にその値を抑える [16]．以下のアルゴリズムではしきい値 θ を設定し，求めた勾配の絶対値がしきい値以上であればしきい値以下に変換している [17]．文献 [18] では $\theta = 1$ が用いられた．

1. $\hat{\mathbf{g}} \leftarrow \frac{\partial \varepsilon}{\partial \theta}$
2. $\hat{\mathbf{g}} \leftarrow \frac{\theta}{\|\hat{\mathbf{g}}\|} \hat{\mathbf{g}}, \quad \text{If } \|\hat{\mathbf{g}}\| \geq \theta$

LSTM

2010 年以降，リカレントニューラルネットワークについては LSTM（long short-term memory）を用いた研究が数多く発表されている．単純再帰型ニューラルネットワークでは，長距離間隙（long-time lag）あるいは長距離依存（long term dependency）を学習することが難しかった（収束に要する繰り返し回数が大きかったため）．これは，直前の情報を適切に利用することが困難だったからである（CEC（constant error carousel）の呪い）．LSTM はこれを解決するために，ゲートを設定し，ゲートの開閉によって 1 時刻前のシステムの状態からの影響を制御する．

図 4 に LSTM の概念図を示す．図で，情報は下から上へと流れる．この図は図 2 と異なり，全ネットワークを示しているのではなく，一つの LSTM セルを描いている．この LSTM セルを複数個用いてニューラルネットワークの 1 層が構成される．図の中央の $\mathbf{c}$ が記憶素子である．$\mathbf{c}$ の前後に二つのゲートが配置され，それぞれ入力ゲート，出力ゲートと呼ばれる．$\mathbf{c}$ の自己フィードバックに関するゲートは忘却ゲートと呼ばれる．$\mathbf{c}$ の出力すなわち過去の状態からの再帰結合は，恒等関数 $f(x) = x, \forall x$ であるため，1.0 と表記している．$\mathbf{c}$ への入力は CEC と忘却ゲートの出力との積である．

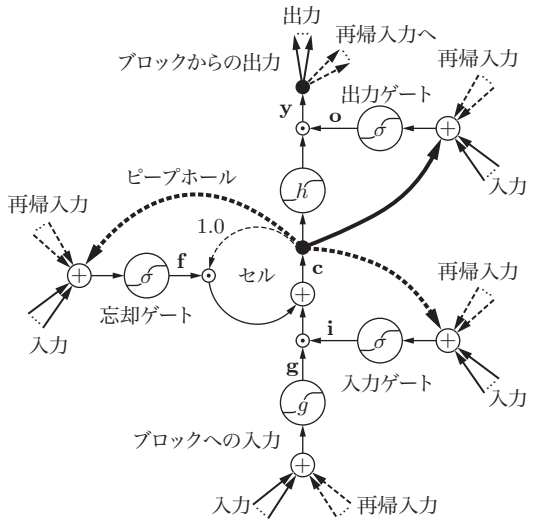

図 4 LSTM の概念図

出力ゲートと入力ゲートの役割は，入出力量をスケーリングすることでメモリセルにどの程度の誤差を学習するかを指示することである．入力ゲートによるスケーリングと出力ゲートのスケーリングにより誤差を開放するか否かの情報を，記憶ユニットが学習することになる．忘却ゲートは，メモリセルに対して，言い換えれば，1 時刻前の自身の保持している情報を利用するか否かを定めるために，ゲート開閉を行う．なお，原典の LSTM [19] においては忘却ゲートは存在せず，Gers ら [20] によって導入された．

LSTM は実用的な繰り返し回数で長期記憶（long-term memory）を制御できる．長期記憶は記憶セル $c^n_{t\in\mathcal{R}}$ のベクトルとして保持される．LSTM はネットワーク全体のアーキテクチャと活性化関数とは独立に長期記憶を保持する能力を持つ．LSTM は記憶セルを上書きし，保持し，引き出すことが可能である．

図 4 内の表記と対応させて，時刻 t における入力を $\mathbf{i}_t$，忘却ゲートを $\mathbf{f}_t$，出力ゲートを $\mathbf{o}_t$，LSTM ユニットの出力を $\mathbf{g}_t$，記憶セルを $\mathbf{c}_t$ とすれば，LSTM のユニットは以下のように表される．

$$\mathbf{i}_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \tag{17}$$
$$\mathbf{f}_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{18}$$
$$\mathbf{o}_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \tag{19}$$
$$\mathbf{g}_t = \phi\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{20}$$
$$\mathbf{c}_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{21}$$
$$\mathbf{y}_t = o_t \odot \phi\left(c_t\right) \tag{22}$$

W は関連する結合係数行列，σ はロジスティック関数

$$\sigma\left(x\right) = \frac{1}{1+\exp\left(-x\right)} \tag{23}$$

であり，ϕ はハイパータンジェント（hyper tangent; tanh）

$$\phi\left(x\right) = \tan\left(x\right) = \frac{\exp\left(x\right)-\exp\left(-x\right)}{\exp\left(x\right)+\exp\left(-x\right)} = 2\sigma\left(2x\right) - 1 \tag{24}$$

である．また，$\odot$ は要素積（アダマール積; Hadamard product）である．LSTM ユニットの学習には，BPTT もしくは RTRL が用いられる．

ゲート付き再帰ユニット（gated recurrent unit; GRU）（図 5）の動作は，以下のように記述できる．

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1}\right) \tag{25}$$
$$\tilde{h}_t = \phi\left(W x_t + U\left(r_t \odot h_{t-1}\right)\right) \tag{26}$$
$$r_t = \sigma\left(W_r x_t + U_r h_{t-1}\right) \tag{27}$$
$$h_t = \left(1 - z_t\right) h_{t-1} + z_t \tilde{h}_t \tag{28}$$
$$y_t = W_y h_t \tag{29}$$

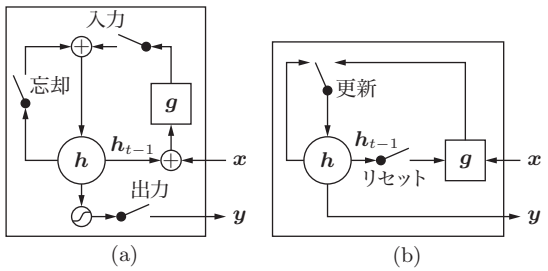

図 5 LSTM (a) とゲート付き再帰ユニット (b) の比較

2010 年代に入ってリカレントニューラルネットワーク，とりわけ LSTM を用いた研究が盛んである．写真からの脚注作成（neural image captioning）では，認識部分に深層学習（deep learning）を用い，出力に LSTM を用いる [21]．Donahue ら [22] は動画へ，Zaremba ら [23] は機械翻訳に，Graves ら [24] は発話解析に，それぞれ LSTM を応用している．

リカレントニューラルネットワークの総説論文としては，[25], [26] がある．LSTM の性能を評価した論文には，[27] を挙げることができる．リカレントニューラルネットワークを多層にした場合に，畳み込み深層学習と同じくドロップアウト（dropout）が行われる場合があるが，フィードフォワードネットワーク（feed-forward network; multi-layered network）型の畳み込

み深層学習と異なり，リカレントニューラルネットワークではドロップアウトの効果が認められないとの報告もあった．これに対し，Zaremba ら [23] はフィードフォワード結合のみをドロップアウトし，リカレント結合はドロップアウトを行わないことを提案している．

その他のリカレントニューラルネットワーク

古典的なジョーダンネットワーク，エルマンネットワークは 1 時刻前の状態と現在の入力に基づいて出力を計算する．これは 1 階差分に類比しうることを意味する．LSTM に見られるように，1 階差分の情報を断ち切って，さらに長期の記憶を呼び起こしたりする機構として，ゲートの存在が注目を集めている．すなわち，高階差分情報を活用するためにゲートの果たす役割が大切だとされ，ゲート付き再帰ユニット [28] など，新たなモデルも提案されてきた．どこまで過去の情報を保持し，どのような周期を仮定するのかについては，リカレントニューラルネットワークの持つ内部構造をランダム結合により実現したエコーステートネットワーク（echo state network; ESN）[29] がある．これは互いに疎に（sparse）結合した中間層ユニットを持つリカレントニューラルネットワークである．学習は出力ユニットについてのみ行われる．

系列データが与えられている場合，次刻の出力を予測する際に $t+\alpha$ なる未来のデータ系列も利用可能であれば，双方向リカレントニューラルネットワーク [30] と呼ばれるリカレントニューラルネットワークを用いた訓練が行われることがある．実際の予測をする際には無視すればよい．

参考文献

[1] Grossberg, S. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, Vol. 1, pp. 17–61, 1988.

[2] Sejnowski, T. J. and Rosenberg, C. R. Parallel Networks that Learn to Pronounce Eglish Text. *Complex Systems*, Vol. 1, pp. 145–168, 1987.

[3] Jordan, M. I. Serial Order: A Parallel Distributed Processing Approach. Technical report, University of California, San Diego, 1986.

[4] Elman, J. L. Finding structure in time. *Cognitive Science*, Vol. 14, No. 2, pp. 179–211, 1990.

[5] Williams, R. J. and Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, Vol. 1, No. 2, pp. 270–280, 1989.

[6] Williams, R. J. and Zipser, D. Gradient-based Learning Algorithms for Recurrent Networks and Their Computational Complexity. In Chauvin, Y. and Rumelhart, D. E., editors, *Backpropagation: Theory, Architectures, and Applications*, pp. 434–486. Lawrence Erlbaum Associate, New Jersey, 1995.

[7] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and Tell: A Neural Image Caption Generator. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2015.

[8] Graves, A., Wayne, G., and Danihelka, I. Neural Turing Machines. *arXiv:1410.5401*, Vol. 1410.5401, 2014.

[9] Zaremba, W. and Sutskever, I. Learning to Execute. In Bengio, Y. and LeCun, Y., editors, *Proc. the International Conference on Learning Representations (ICLR)*, 2015.

[10] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to Sequence Learning with Neural Networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 3104–3112, 2014.

[11] Karpathy, A. and Fei-Fei, L. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] Socher, R., Manning, C., and Ng, A. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. In *Proc. Advances in Neural Information Processing Systems*, 2010.

[13] Werbos, P. J. Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1550–1560, 1990.

[14] Hanselmann, T., Zaknich, A., and Attikiouzel, Y., authors, Mastorakis, N., editor. Connection between BPTT and RTRL. *Computational Intelligence and Applications*, Vol. 1, pp. 97–102, 1999.

[15] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. Gradient Flow in Recurrent Nets the Difficulty of Learning Long-Term Dependencies. In Kremer, S. C. and Kolen, J. F., editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.

[16] Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. Advances in Optimizing Recurrent Networks. *arXiv:1212.0901*, 2012.

[17] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. *arXiv:1211.5063*, 2013.

[18] Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850*, 2013.

[19] Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[20] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, Vol. 12, pp. 2451–2471, 2000.

[21] Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv:1502.03044v2*, 2015.

[22] Donahue, J., Hendricks, Anne, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent Neural Network Regularization. In Bengio, Y. and LeCun, Y., editors, *Proc. the International Conference on Learning Representations (ICLR)*, 2015.

[24] Graves, A., Mohamed, A., and Hinton, G. Speech recognition with deep recurrent neural networks. In Ward, R. K., editor, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649, 2013.

[25] Lipton, Z. C., Berkowitz, J., and Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019*, 2015.

[26] Jozefowicz, R., Zaremba, W., and Sutskever, I. An Empirical Exploration of Recurrent Network Architectures. In Pineau, J., Bach, F., and Blei, D., editors, *Proc. the 32nd Annual International Conference on Machine Learning (ICML)*, 2015.

[27] Greff, K., Srivastava, R. K., Koutnìk, J., Steunebrink, B. R., and Schmidhuber, J. LSTM: A Search Space Odyssey. *arXiv*, 2015.

[28] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555*, 2014.

[29] Jaeger, H. A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. Technical report, Fraunhofer Institute for Autonomous Intelligent Systems, 2002.

[30] Schuster, M. and Paliwal, K. K. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2673–2681, 1997.

執筆者：浅川伸一

和英索引

【あ】

RTRL(real time recurrent learning; RTRL)　2

アダマール積 (Hadamard product)　6

【え】

エコーステートネットワーク (echo state network; ESN)　7

LSTM(long short-term memory; LSTM)　5

エルマンネットワーク (Elman network)　2

【こ】

勾配消失問題 (gradient vanishing problem)　4

勾配爆発問題 (gradient exploding problem)　5

【し】

GRU(gated recurrent unit; GRU)　6

実時間リカレント学習 (real time recurrent learning; RTRL)　2

ジョーダンネットワーク (Jordan network)　2

深層学習 (deep learning)　6

【ち】

チューリングマシン (Turing machine)　2

長期記憶 (long-term memory)　6

長距離依存 (long term dependency)　5

長距離間隙 (long-time lag)　5

【と】

特異値分解 (singular value decomposition; SVD)　5

ドロップアウト (dropout)　6

【ね】

ネットトーク (NETtalk)　2

【は】

ハイパータンジェント (hyper tangent)　6

【ひ】

BPTT(back-propagation through time; BPTT)　2

【ふ】

フィードバック結合 (feedback connection)　2

フィードフォワードネットワーク (feed-forward network, multi-layered network)　6

文脈層 (context layer)　3

【や】

ヤコビアン (Jacobian)　5

【り】

リカーシブニューラルネットワーク (recursive neural network)　2

リカレントニューラルネットワーク (Recurrent Neural Network)　2

リカレントニューラルネットワーク (recurrent neural network; RNN)　2

英和索引

【B】

back-propagation through time; BPTT（BPTT） 2

【C】

context layer（文脈層） 3

【D】

deep learning（深層学習） 6

dropout（ドロップアウト） 6

【E】

echo state network; ESN（エコーステートネットワーク） 7

Elman network（エルマンネットワーク） 2

【F】

feed-forward network, multi-layered network（フィードフォワードネットワーク） 6

feedback connection（フィードバック結合） 2

【G】

gated recurrent unit; GRU（GRU） 6

gradient exploding problem（勾配爆発問題） 5

gradient vanishing problem（勾配消失問題） 4

【H】

Hadamard product（アダマール積） 6

hyper tangent（ハイパータンジェント） 6

【J】

Jacobian（ヤコビアン） 5

Jordan network（ジョーダンネットワーク） 2

【L】

long short-term memory; LSTM（LSTM） 5

long term dependency（長距離依存） 5

long-term memory（長期記憶） 6

long-time lag（長距離間隙） 5

【N】

NETtalk（ネットトーク） 2

【R】

real time recurrent learning; RTRL（RTRL） 2

real time recurrent learning; RTRL（実時間リカレント学習） 2

Recurrent Neural Network（リカレントニューラルネットワーク） 2

recurrent neural network; RNN（リカレントニューラルネットワーク） 2

recursive neural network（リカーシブニューラルネットワーク） 2

【S】

singular value decomposition; SVD（特異値分解） 5

【T】

Turing machine（チューリングマシン） 2