- 入力層

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \tag{2}$$

- 隠れ層の活性化関数

$$\boldsymbol{h}^{(t)} = tanh(\boldsymbol{a}^{(t)}) \tag{3}$$

- 出力層

$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \tag{4}$$

- 損失関数

$$L = \frac{1}{2}(\boldsymbol{o}^{(t)} - \boldsymbol{y}^{(t)})^2 \tag{5}$$

ここで，パラメータベクトルを $\boldsymbol{b}$，$\boldsymbol{c}$ とする．また，損失 L は出力 $\boldsymbol{o}^{(t)}$ と正解 $\boldsymbol{y}^{(t)}$ の二乗誤差とする．最終的に，各重み行列の勾配は以下のような計算結果を得る．

$$\nabla_{\boldsymbol{V}} L = \sum_t (\nabla_{\boldsymbol{o}^{(t)}} L)\boldsymbol{h}^{(t)\top} \tag{6}$$

$$\nabla_{\boldsymbol{W}} L = \sum_t (1 - (\boldsymbol{h}^{(t)})^2)((\nabla_{\boldsymbol{h}^{(t)}} L)\boldsymbol{h}^{(t-1)\top} \tag{7}$$

$$\nabla_{\boldsymbol{U}} L = \sum_t (1 - (\boldsymbol{h}^{(t)})^2)((\nabla_{\boldsymbol{h}^{(t)}} L)\boldsymbol{x}^{(t)\top} \tag{8}$$

計算コストが高くなる理由は，$\nabla_{\boldsymbol{h}^{(t)}} L$ にある．ここで，$\boldsymbol{h}^{(t)}$ についての勾配計算を式 (9) に示す．このように，時間ステップ t について勾配計算を行う際に $\nabla_{\boldsymbol{h}^{(t+1)}} L$ の計算結果を用いる必要があるため，計算コストが高くなる．

$$\begin{aligned}\nabla_{\boldsymbol{h}^{(t)}} L = &(\boldsymbol{V}^{(t)})^\top (\boldsymbol{o}^{(t)} - \boldsymbol{y}^{(t)}) \\ &+ (\boldsymbol{W}^{(t+1)})^\top (1 - (\boldsymbol{h}^{(t+1)})^2)(\nabla_{\boldsymbol{h}^{(t+1)}} L)\end{aligned} \tag{9}$$

3 具体的な RNN

具体例を 5 つあげ，概要と利点，応用例を説明する．

3.1 出力から隠れ層へ回帰接続する RNN

出力から隠れ層へ回帰接続する RNN について説明する．計算グラフを Figure 4 に示す．概要として，テスト時において，重み行列 $\boldsymbol{W}$ は出力から隠れ状態へ重み付けされる．また，各時間ステップで出力が生成され，同じ系列長の入出力処理が可能である．学習には，教師強制と呼ばれる方法が用いられる．

利点として，通時的誤差逆伝播法を用いる必要がない点があげられる．代わりに教師強制を用いることで，各時間ステップで RNN が分離されるため，学習の並列処理が可能となる．応用例として，文章生成に用いられる．

教師強制は，学習時に正解 $\boldsymbol{y}$ を次の時間ステップの入力として与えることである．つまり，テスト時は出力 $\boldsymbol{o}$ から隠れ状態 $\boldsymbol{h}$ へ $\boldsymbol{W}$ を重み付け，学習時は正解 $\boldsymbol{y}$ から隠れ状態 $\boldsymbol{h}$ へ $\boldsymbol{W}$ を重み付ける．

3.2 出力を 1 つ生成する RNN

出力を 1 つ生成する RNN について説明する．計算グラフを Figure 5 に示す．概要として，各時間ステップで入力は行われるが，出力は最後の時間ステップのみ生成される．学習には，通時的誤差逆伝播法を用いる．

利点として，入力系列の要約が可能である点があげられる．応用例として，感情分析に用いられる．

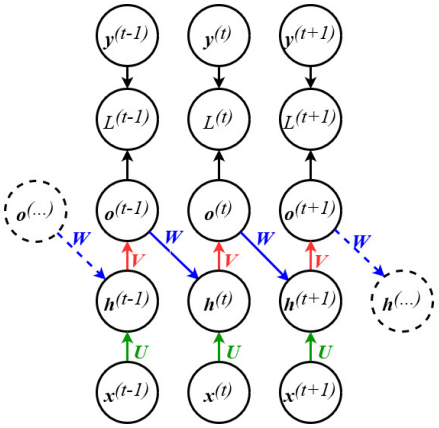

Figure 4　出力から隠れ層へ回帰接続する RNN

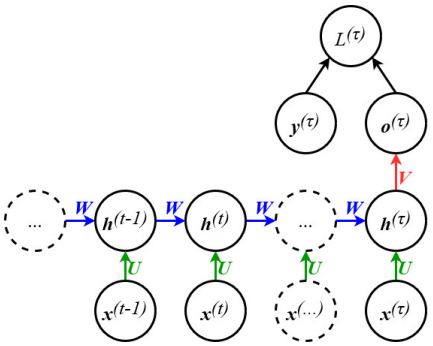

Figure 5　出力を 1 つ生成する RNN

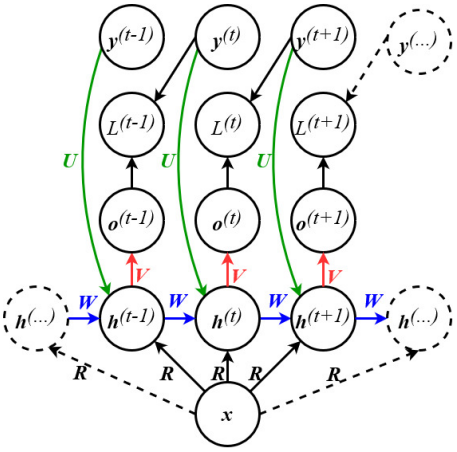

Figure 6　固定長ベクトルを入力する RNN

3.3 固定長ベクトルを入力する RNN

固定長ベクトルを入力する RNN について説明する．計算グラフを Figure 6 に示す．概要として，系列データを入力とするだけでなく，固定長ベクトル $\boldsymbol{x}$ を入力することが可能である．ここで，新たな重み行列 $\boldsymbol{R}$ を導入する．この時，$\boldsymbol{y}^{(t)}$ は入力や正解として振る舞う．

利点として，画像データのような非系列データを利用できる点があげられる．応用例として，画像キャプショニングに利用される．画像キャプショニングは，画像を入力することで画像の状況を説明する単語系列が出力される．

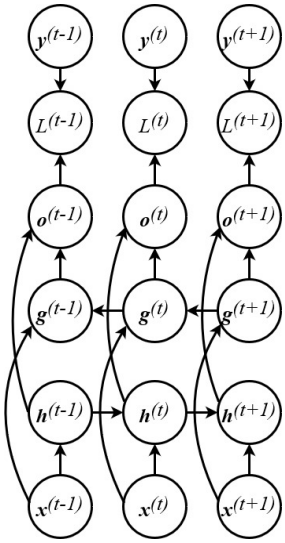

Figure 7　双方向 RNN

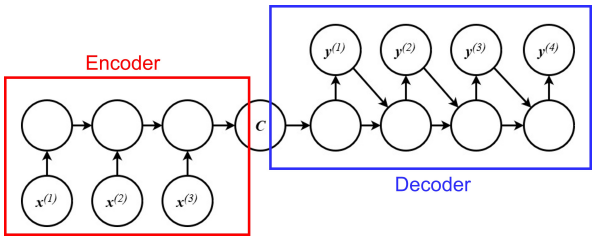

Figure 8　Sequence-to-Sequence

3.4　双方向 RNN

双方向 RNN は，系列の開始から時間と同じ方向で移動する RNN と，系列の終わりから時間と逆の方向で移動する RNN を組み合わせたニューラルネットワークである．計算グラフを Figure 7 に示す．

利点として，過去と未来の情報を利用できる点がある．応用例として，手書き文字認識や音声認識に用いられる．例として，前後の文脈や余韻などの未来の情報が文字や音声の認識に効果を示す．

3.5　Sequence-to-Sequence

構成要素は，Encoder（符号化器）と Decoder（復号化器）である．Encoder は，入力系列をより低次元の情報へ変換する．逆に，Decoder とは，入力系列をより高次元の情報へ変換する．通常，Encoder により実現される符号化 RNN の最終状態を，Decoder により実現される復号化 RNN の入力系列として与える．この符号化 RNN と復号化 RNN を組み合わせた構造を Sequence-to-Sequence と呼ぶ．計算グラフを Figure 8 に示す．

利点として，異なる系列長の入力と出力が処理できる点がある．固定長ベクトル $\boldsymbol{C}$ により実現され，$\boldsymbol{C}$ は Encoder 側で入力系列の要約として出力され，この要約データを Decoder 側で入力として与える．応用例として，機械翻訳や質問応答に用いられる．

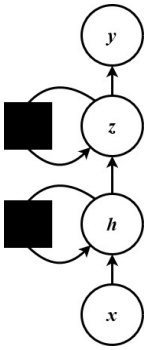

Figure 9　隠れ層の数を増加させる方法

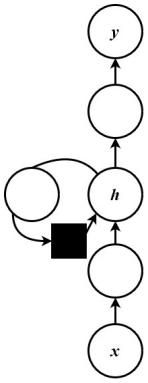

Figure 10　各計算ブロック間に層を増加させる方法

4　RNN における深層化

RNN を深層化することが可能である．層を深くする方法として，2 つあげられる．

まず，隠れ層の数を増加させる方法がある．計算グラフを Figure 9 に示す．次に，各計算ブロック間に層を増加させる方法がある．RNN の計算ブロックは，入力から隠れ層，隠れ層から隠れ層，隠れ層から出力の 3 つに分割できる．計算グラフを Figure 10 に示す．

5　まとめ

RNN は，回帰構造を持ったニューラルネットワークである．特徴として，可変長の系列データを処理でき，すべての時間ステップで同じ重みを共有することができる．基本的に，学習には通時的誤差逆伝播法が用いられる．RNN の例として，同じ系列長の入出力処理が可能な RNN，出力が 1 つのみ生成される RNN，固定長ベクトルを入力する RNN，順方向および逆方向の RNN を組み合わせた双方向 RNN，異なる系列長の入出力処理が可能な Sequence-to-Sequence がある．また，RNN は隠れ層を増やすことや，各計算ブロック間で層を増やすことで深層化することが可能である．

参考文献

1) 竹縄知之，深層学習 part4，東京海洋大学，https://www2.kaiyodai.ac.jp/~takenawa/learning/lecture_Part4.pdf，参照 Apr.26.2024