知能プログラミング演習 I

第 5 回: 再帰型ニューラルネットワーク

梅津 佑太
2 **号館** 404A: umezu.yuta@nitech.ac.jp

課題のダウンロード

前回作ったディレクトリに移動して今日の課題のダウンロードと解凍

step1: `cd ./DLL`

step2: `wget http://www-als.ics.nitech.ac.jp/~umezu/DLL19/Lec5.zip`

step3: `unzip Lec5.zip`

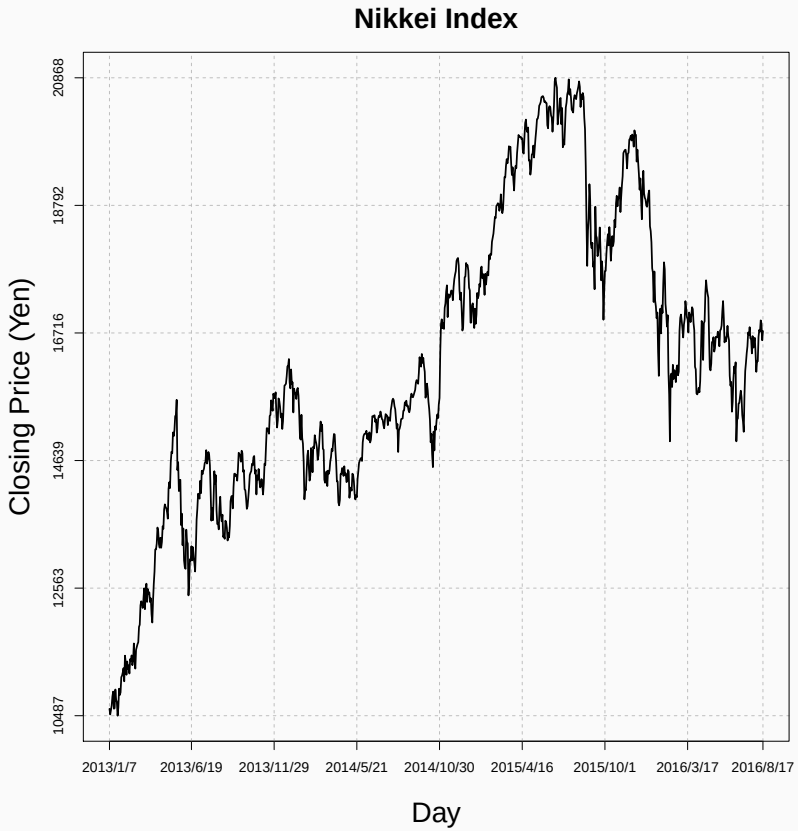
- ✓ **まだ** DLL **のフォルダを作ってない人は**, step1 **の前に**
  `mkdir -p DLL`
  でフォルダを作成する

今日の講義内容

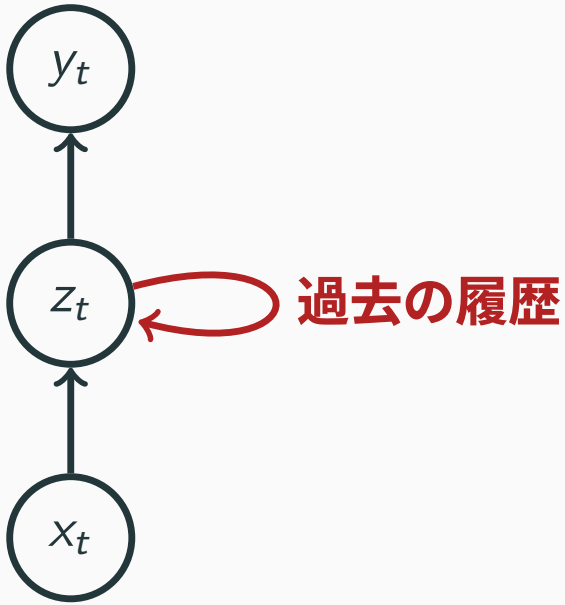
1. **再帰型ニューラルネットワークの学習**

空間依存性

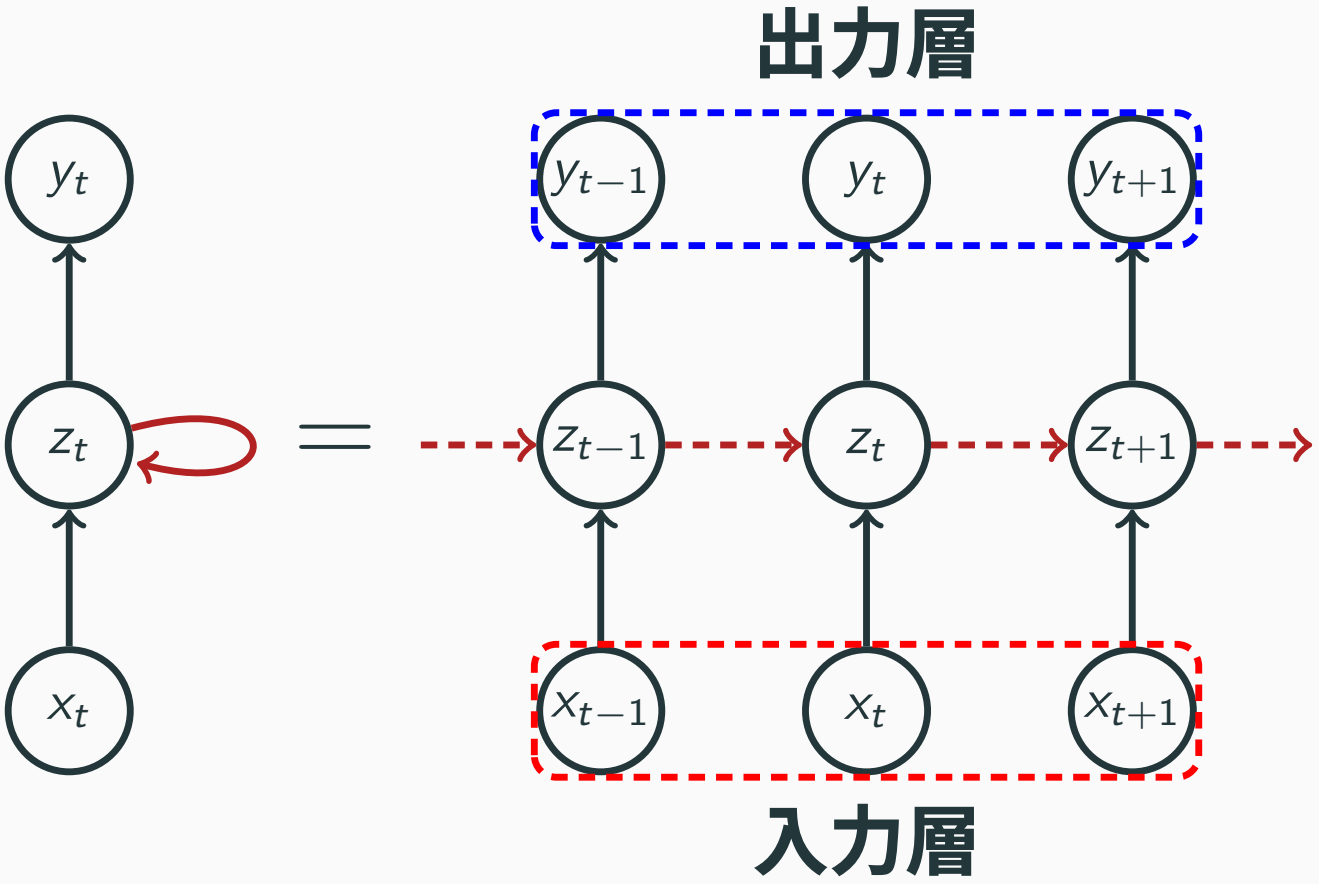
- 時系列，動画，テキストなどは，“現在の値”が “過去の値”に依存



ネットワーク表現 (モデル)

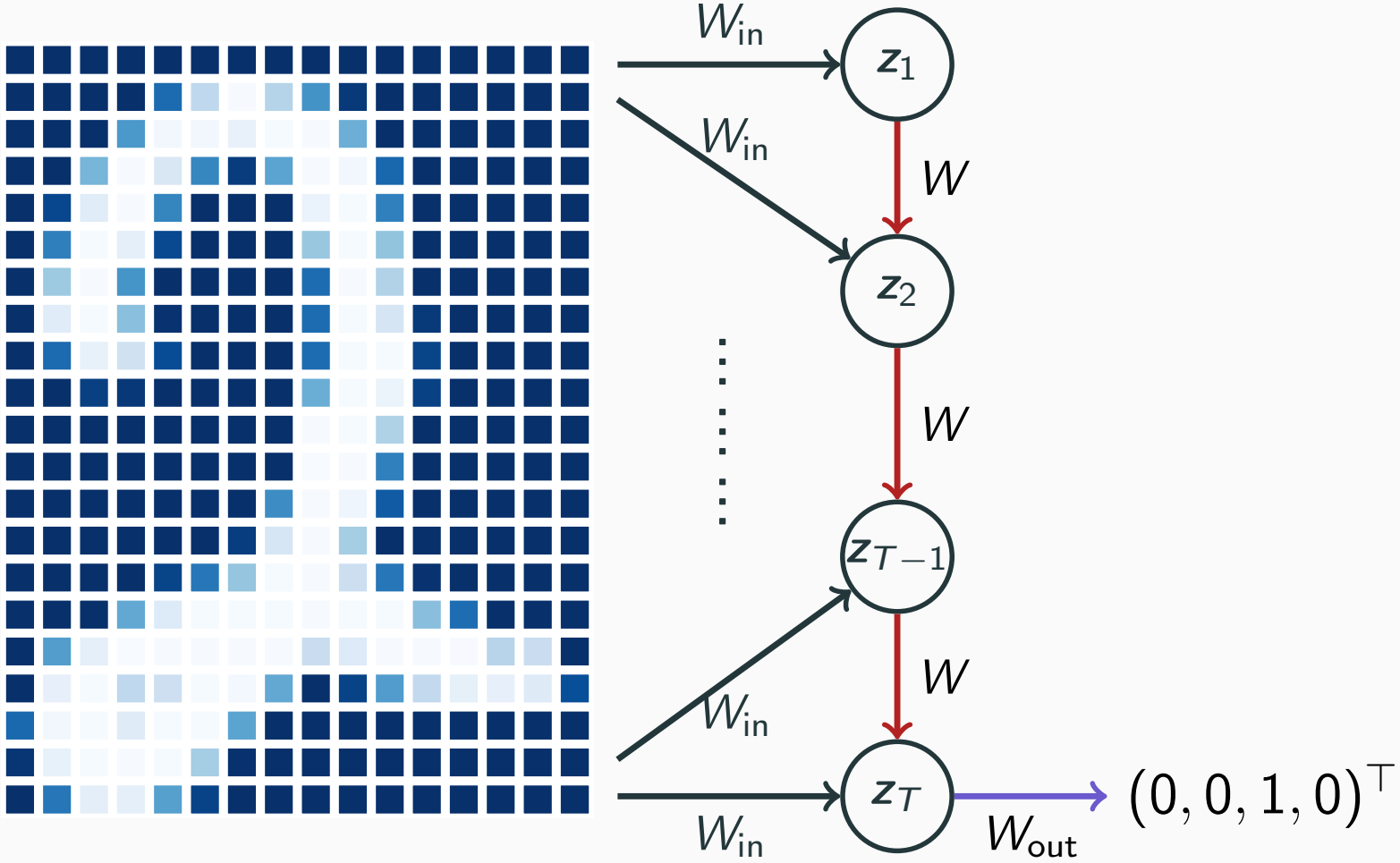


再帰型ニューラルネットワーク



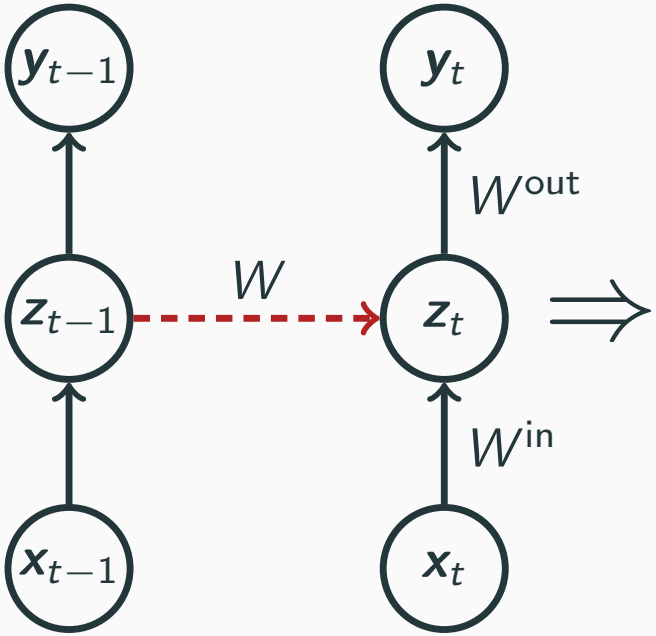
- 入力ごとに出力を計算する再帰型ニューラルネットワークを many-to-many ネットワークと呼ぶ
- 各時点における入出力関係は 3 層ニューラルネットワーク
- 中間層のネットワークにより，全体として “深い” モデルを構成

画像の分類: many-to-one ネットワーク



- 行ごとにデータをスキャンし，最終的な出力を用いて予測する
- 分類問題の場合，誤差関数はクロスエントロピー，最後の中間層から出力層への活性化関数はソフトマックス関数を用いる

再帰型ニューラルネットワークの関数表現



$t = 1, \ldots, T$ **に対して**,

$$z_t = f(W^{\text{in}} x_t + W z_{t-1})$$
$$y_t = g(W^{\text{out}} z_t)$$

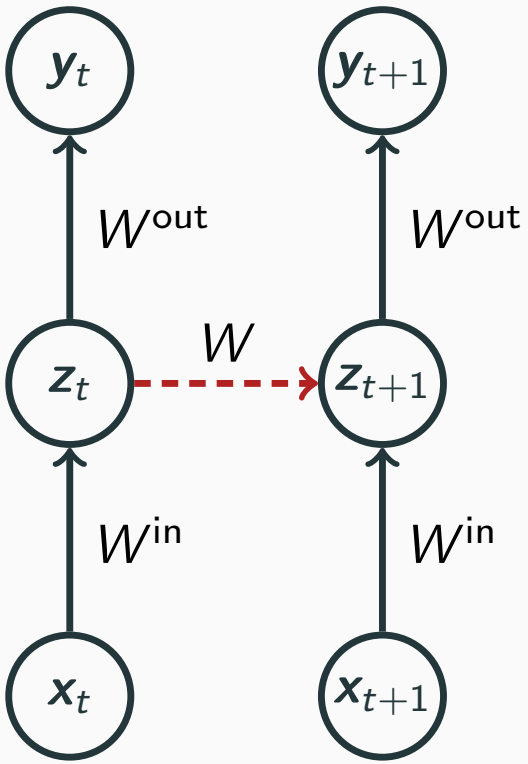
注意: $z_0 = \mathbf{0}$ **とし**, $W^{\text{in}} \in \mathbb{R}^{q \times (T+1)}$, $W^{\text{out}} \in \mathbb{R}^{m \times (q+1)}$ **にはバイアス項も含まれるものとする**

順伝播は, **線形結合** + **活性化関数の繰り返し**. **簡単のため**, **誤差関数は引数を省略して書くと** $E = \sum_{t=1}^{T} E_t$. **ただし**,

(**回帰**) $E_t = \frac{1}{2}(y_t - g(W^{\text{out}} f(W^{\text{in}} x_t + W z_{t-1})))^2$

(**分類**) $E_t = -\sum_{k=1}^{m} y_{tk} \log g(w_k^{\text{out}\top} f(w_k^{\text{in}\top} x_t + w_k^{\top} z_{t-1}))$

誤差逆伝播



$\boldsymbol{u}_t = W^{\text{in}}\boldsymbol{x}_t + W\boldsymbol{z}_{t-1}$, $\boldsymbol{v}_t = W^{\text{out}}\boldsymbol{z}_t$, **および以下を定義する**:

$$\boldsymbol{\delta}_t = \frac{\partial E}{\partial \boldsymbol{u}_t}, \quad \boldsymbol{\delta}_t^{out} = \frac{\partial E}{\partial \boldsymbol{v}_t} = g(\boldsymbol{v}_t) - \boldsymbol{y}_t$$

このとき,

$$\begin{aligned}\boldsymbol{\delta}_t &= \frac{\partial E}{\partial \boldsymbol{u}_{t+1}}\frac{\partial \boldsymbol{u}_{t+1}}{\partial \boldsymbol{u}_t} + \frac{\partial E}{\partial \boldsymbol{v}_t}\frac{\partial \boldsymbol{v}_t}{\partial \boldsymbol{u}_t} \\ &= (W^\top \boldsymbol{\delta}_{t+1} + \tilde{W}^{\text{out}\top}\boldsymbol{\delta}_t^{\text{out}}) \odot \nabla f(\boldsymbol{u}_t)\end{aligned}$$

で誤差を逆伝播する.

ただし, $\tilde{W}^{\text{out}}$ **は** W^{out} **の** 1 **列目を取り除いた行列で**, $\delta_{T+1} = \mathbf{0}$ **とする**.
また, $\boldsymbol{u}_t^{\text{in}} = W^{\text{in}}\boldsymbol{x}_t$ **としたとき**,

$$\frac{\partial E}{\partial \boldsymbol{u}_t^{\text{in}}} = \frac{\partial E}{\partial \boldsymbol{u}_t}\frac{\partial \boldsymbol{u}_t}{\partial \boldsymbol{u}_t^{\text{in}}} = \frac{\partial E}{\partial \boldsymbol{u}_t}$$

なので, **中間層から入力層への逆伝播にも** $\partial E/\partial \boldsymbol{u}_t$ **を利用できる**.

パラメータの更新ルール I

バイアス項を含む $\boldsymbol{x}_t$ **(元データ** $X \in \mathbb{R}^{T \times T}$ **の行ベクトルの** 1 **列目に** 1 **を加えた** $T \times (T+1)$ **行列) に対して**,

$$\frac{\partial E}{\partial W^{\text{in}}} = \sum_{t=1}^{T} \frac{\partial E}{\partial \boldsymbol{u}_t} \frac{\partial \boldsymbol{u}_t}{\partial W^{\text{in}}} = \sum_{t=1}^{T} \boldsymbol{\delta}_t \boldsymbol{x}_t^{\top}$$

となる. $D = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_T)$ **とすれば**,

$$\frac{\partial E}{\partial W^{\text{in}}} = DX$$

となる. **同様に**, $\tilde{Z} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T-1})^{\top}$ $(\boldsymbol{z}_0 = \boldsymbol{0})$ **とすれば**,

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E}{\partial \boldsymbol{u}_t} \frac{\partial \boldsymbol{u}_t}{\partial W} = \sum_{t=1}^{T} \boldsymbol{\delta}_t \boldsymbol{z}_{t-1}^{\top} = D\tilde{Z}$$

となる.

パラメータの更新ルール II

さらに, $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T)$ **とすれば**,

$$\frac{\partial E}{\partial W^{\text{out}}} = \sum_{t=1}^{T} \frac{\partial E}{\partial \boldsymbol{v}_t} \frac{\partial \boldsymbol{v}_t}{\partial W^{\text{out}}} = \sum_{t=1}^{T} (g(\boldsymbol{v}_t) - \boldsymbol{y}_t) \boldsymbol{z}_t^\top = (g(V) - Y) Z$$

が得られるので, **これらを用いてパラメータを更新する**. **ただし**,

$$g(V) - Y = (g(\boldsymbol{v}_1) - \boldsymbol{y}_1, \ldots, g(\boldsymbol{v}_T) - \boldsymbol{y}_T)$$

- many-to-one **ネットワークの場合には**, **上記の** $\frac{\partial E}{\partial W^{\text{out}}}$ **の代わりに**,

$$\frac{\partial E}{\partial W^{\text{out}}} = (g(\boldsymbol{v}_T) - \boldsymbol{y}_T) \boldsymbol{z}_T^\top \in \mathbb{R}^{m \times T}$$

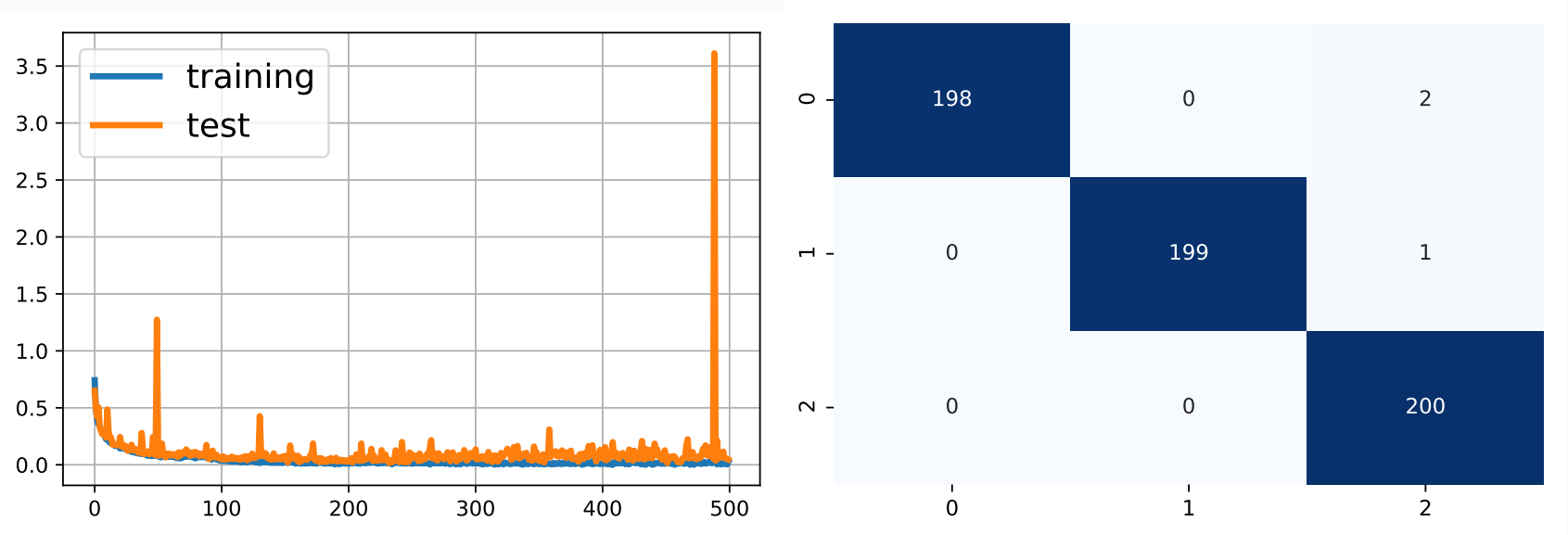
を用いる

疑似コード: many-to-many ネットワークの場合

入力: $\{(\boldsymbol{y}_t, \boldsymbol{x}_t)\}_{t=1}^{T}$ **に関する** n **組のデータ**, **各層における活性化関数** (f, g)

1. **パラメータの初期化**: $W^{\text{in}}, W^{\text{out}}, W$
2. **ランダムにデータを読み込む**: $\{(\boldsymbol{y}_t, \boldsymbol{x}_t)\}$
3. **順伝播**: $t = 1, 2, \ldots, T$ **に対して**,
   - 3.1 $\boldsymbol{z}_t \leftarrow f(W^{\text{in}}\boldsymbol{x}_t + W\boldsymbol{z}_{t-1}) \quad (\boldsymbol{z}_0 = \boldsymbol{0})$
   - 3.2 $\boldsymbol{y}_t \leftarrow g(W^{\text{out}}\boldsymbol{z}_t)$
4. **逆伝播**: $t = T, \ldots, 1$ **に対して**,
   - 4.1 $\boldsymbol{\delta}_t \leftarrow (W^{\top}\boldsymbol{\delta}_{t+1} + \tilde{W}^{\text{out}\top}(g(\boldsymbol{v}_t) - \boldsymbol{y}_t)) \odot \nabla f(\boldsymbol{u}_t) \quad (\boldsymbol{\delta}_{T+1} = \boldsymbol{0})$
5. **重みの更新**: Adam **などを用いてパラメータを更新する**. **通常の確率的勾配法なら下記の通り**.
   - 5.1 $W^{\text{in}} \leftarrow W^{\text{in}} - \eta DX$
   - 5.2 $W \leftarrow W - \eta D\tilde{Z}$
   - 5.3 $W^{\text{out}} \leftarrow W^{\text{out}} - \eta(g(V) - Y)Z$
6. **収束するまで** 2 – 5 **を繰り返す**

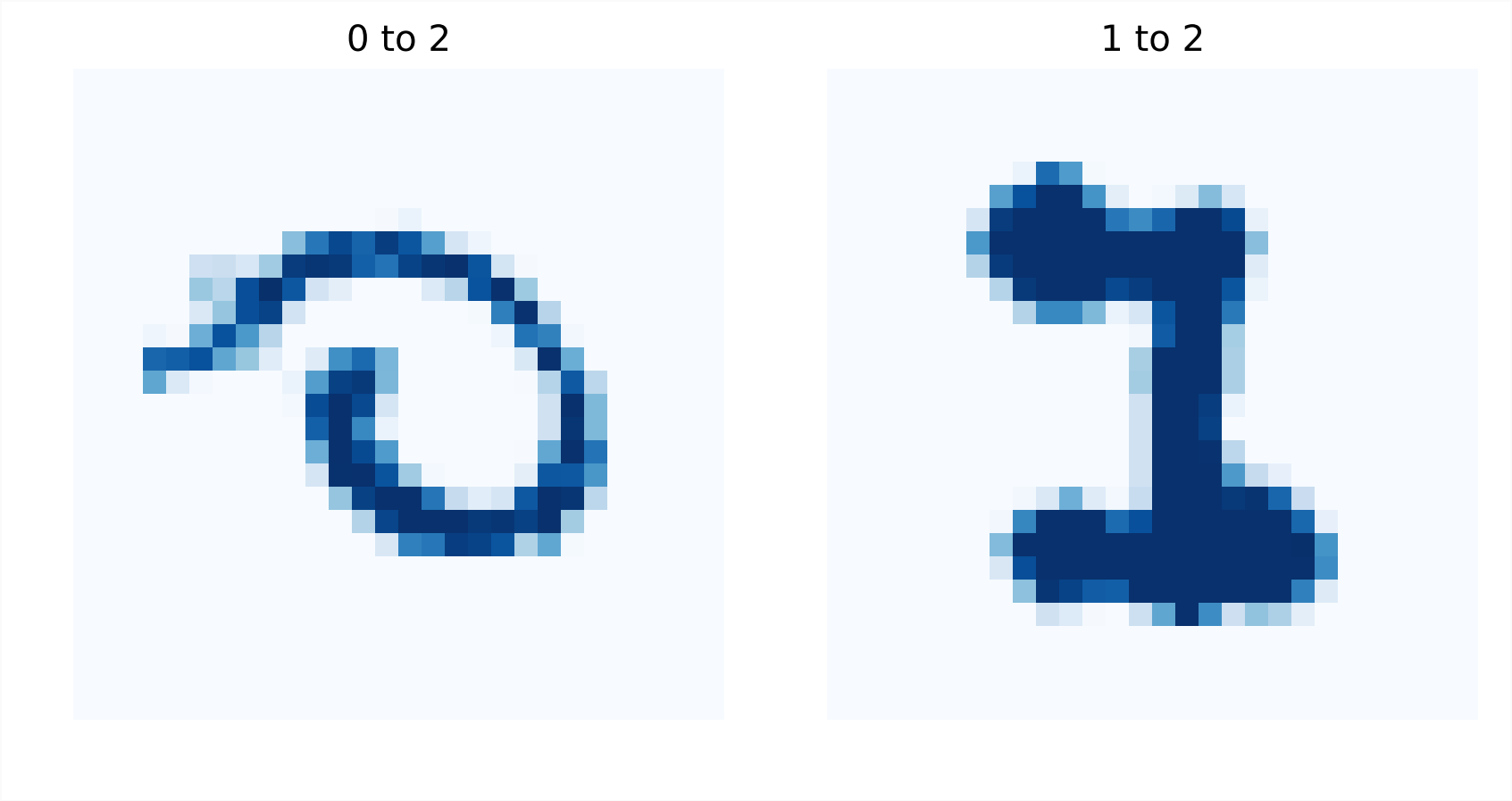
結果の例 I



(a) 誤差の推移　　**(b)** confusion matrix

Figure 1: 500 エポックでの誤差関数の推移と confusion matrix. 活性化関数はシグモイド関数を用いた.

結果の例 II



(a) 0 を 2 と誤ったもの　　(b) 1 を 2 と誤ったもの

Figure 2: 誤って分類したデータの図示.