

# 13. 系列データの識別

- 本章の説明手順
  - ◆ 系列識別問題の分類
  - ◆ 入力長と出力長が  $n$  対  $n$  の場合
  - ◆ 入力長と出力長が  $n$  対  $1$  の場合
  - ◆ 入力長と出力長が  $n$  対  $m$  の場合

# 13.1 ラベル系列に対する識別

- ラベル系列に対する識別問題の分類
  - ◆ 入力の系列長と出力の系列長が等しい
    - 例) 形態素解析、固有表現抽出
    - 系列ラベリング問題 ⇒ CRF
  - ◆ 入力の系列長に関わらず出力の系列長が1
    - 例) 動画像の分類、文書分類
    - 系列識別問題 ⇒ HMM
  - ◆ 入力の系列長と出力の系列長に対応関係がない
    - 例) 連続音声認識、機械翻訳
    - 系列変換問題 ⇒ RNN, Seq2Seq+Attention, Transformer

## 13.2 系列ラベリング問題 — CRF —

- 系列ラベリング問題の事例

- ◆ 形態素解析

入力	系列	で	入力	さ	れる	各	要素
出力	名詞	助詞	名詞	動詞	接尾辞	接頭辞	名詞

- ◆ 固有表現抽出

入力	Apple is looking at buying U.K. startup for \$1 billion												
出力	B-ORG	O	O		O	O		B-GPE	O		O	B-MONEY	I-MONEY

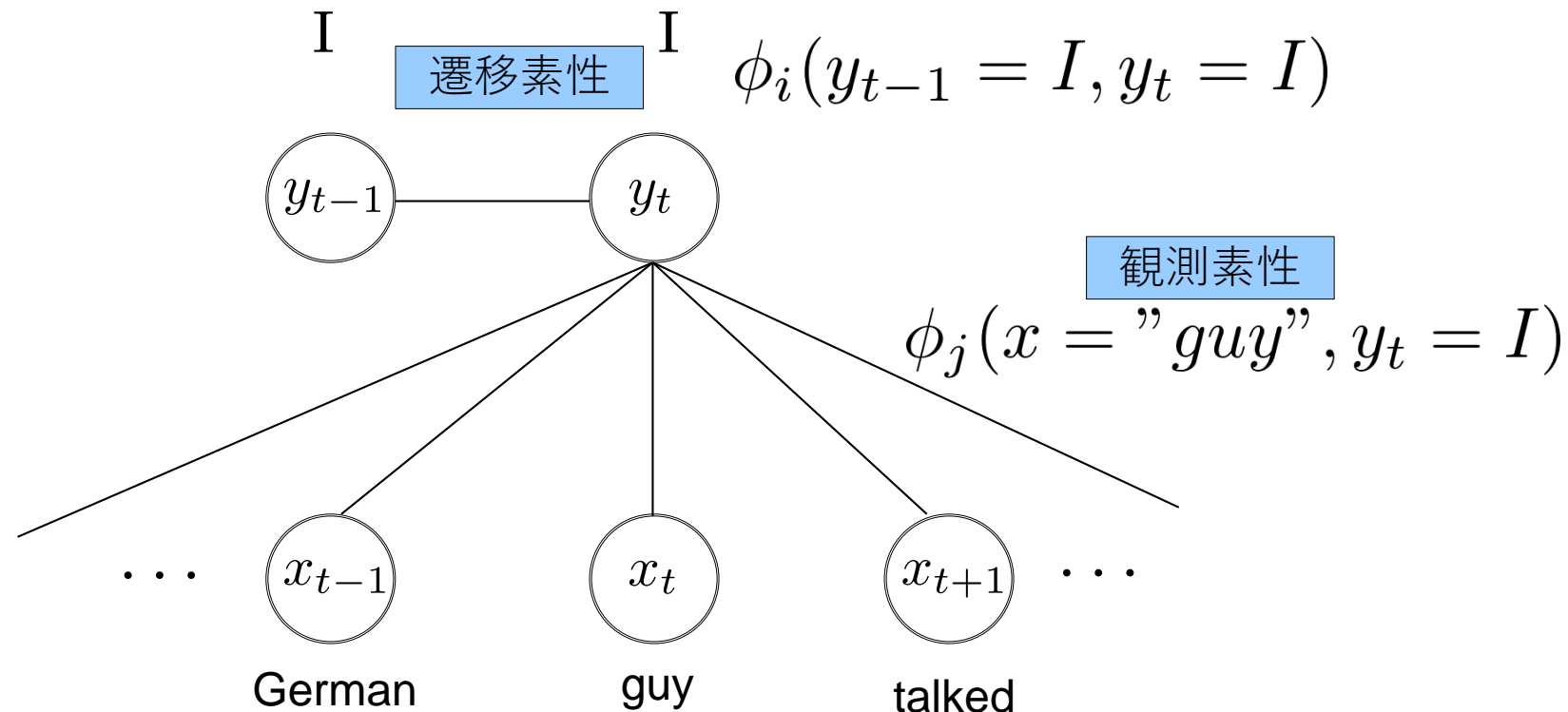
B: begin  
I: inside  
O: outside

## 13.2 系列ラベリング問題 — CRF —

- 系列ラベリング問題とは
  - ◆ 入力系列の個々の要素に対して出力ラベルを付与する
  - ◆ ラベルの出現確率は前後のラベル系列に依存
    - 1入力1出力の識別器を連続的に適用する方法では性能が低い  
⇒ 入力や出力の系列としての特徴を使う
  - ◆ 可能なラベル系列は膨大な数
    - すべてのラベル系列をリストアップすることは現実的に不可能  
⇒ 探索によって最適解を求める

## 13.2 系列ラベリング問題 — CRF —

- 対数線型モデルによる系列ラベリング
  - ◆ 素性関数の導入
    - 入力系列  $\mathbf{x}$  と出力系列  $\mathbf{y}$  との間に定義される関数
    - 関係が成立すれば1、不成立なら0を値とする



## 13.2 系列ラベリング問題 — CRF —

- 対数線型モデル（多クラスロジステック回帰）

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z_{\boldsymbol{x},\boldsymbol{w}}} \exp(\boldsymbol{w}^T \phi(\boldsymbol{x}, \boldsymbol{y}))$$

$$Z_{\boldsymbol{x},\boldsymbol{w}} = \sum_{\boldsymbol{y}} \exp(\boldsymbol{w}^T \phi(\boldsymbol{x}, \boldsymbol{y}))$$

- 出力の決定

$$\boldsymbol{y}^* = \arg \max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})$$

$$= \arg \max_{\boldsymbol{y}} \frac{1}{Z_{\boldsymbol{x},\boldsymbol{w}}} \exp(\boldsymbol{w}^T \phi(\boldsymbol{x}, \boldsymbol{y}))$$

$$= \arg \max_{\boldsymbol{y}} \boldsymbol{w}^T \phi(\boldsymbol{x}, \boldsymbol{y})$$

## 13.2 系列ラベリング問題 — CRF —

- 素性関数の制限：出力系列を隣接するものに限定

$$y^* = \arg \max_y \sum_t w^T \phi(x, y_t, y_{t-1})$$

- ビタビアルゴリズムによって探索が可能

---

Algorithm 12.1 ビタビアルゴリズム

---

for  $t = 2$  to  $|x|$  do

  for all  $y_t$  do

$$\alpha(t, y_t) = \max_{y_{t-1}} \{w \cdot \phi(x, y_t, y_{t-1}) + \alpha(t-1, y_{t-1})\}$$

$$B(t, y_t) = \arg \max_{y_{t-1}} \{w \cdot \phi(x, y_t, y_{t-1}) + \alpha(t-1, y_{t-1})\}$$

  end for

end for

$y^* = \alpha$ の最大値に対応する  $B$  を逆に辿る

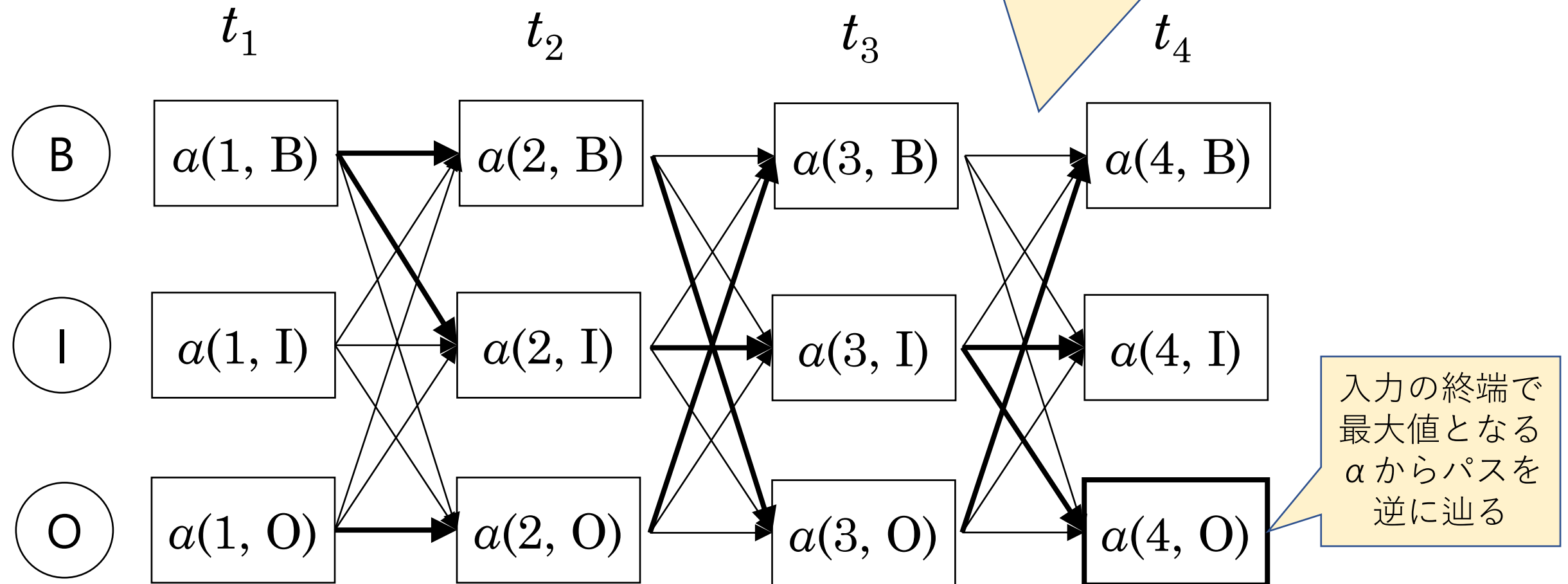
---

順次求める最適スコア

バックポインタ

## 13.2 系列ラベリング問題 — CRF —

- ビタビアルゴリズム





## 13.2 系列ラベリング問題 — CRF —

- CRFの学習
  - ◆ 基本的には多クラスロジステック回帰と同様の手順
    - 最急勾配法などで対数尤度を最大化する
    - L1, L2の正則化項も導入可能
  - ◆ 系列に対する拡張
    - ある時点での重みの更新は、その前後の系列の出力確率に影響を与えるが、その計算を動的計画法で行う（forward-backward アルゴリズム）

# 13.3 系列識別問題 — HMM —

- 系列識別問題の事例

- ◆ PC操作系列による熟練度の判定

- k: キーボード、g: マウス、e: エラー

- 初心者の入力系列例

k e k g k e k g g k g k k e g e e k e e e g e

- 熟練者の入力系列例

k k e k g k k k e k g k g g g e g k g

- 判定したい入力系列

k g e k g k k g e k g e k e e k e g e k

## 13.3 系列識別問題 — HMM —

- 生成モデルによるアプローチ
  - ◆ 系列識別問題ではクラスの事前確率が得られることが多い

$$\begin{aligned} y^* &= \arg \max_y P(y|\mathbf{x}) \\ &= \arg \max_y \frac{P(\mathbf{x}, y)}{P(\mathbf{x})} \\ &= \arg \max_y \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \arg \max_y P(\mathbf{x}|y)P(y) \end{aligned}$$

生成モデル

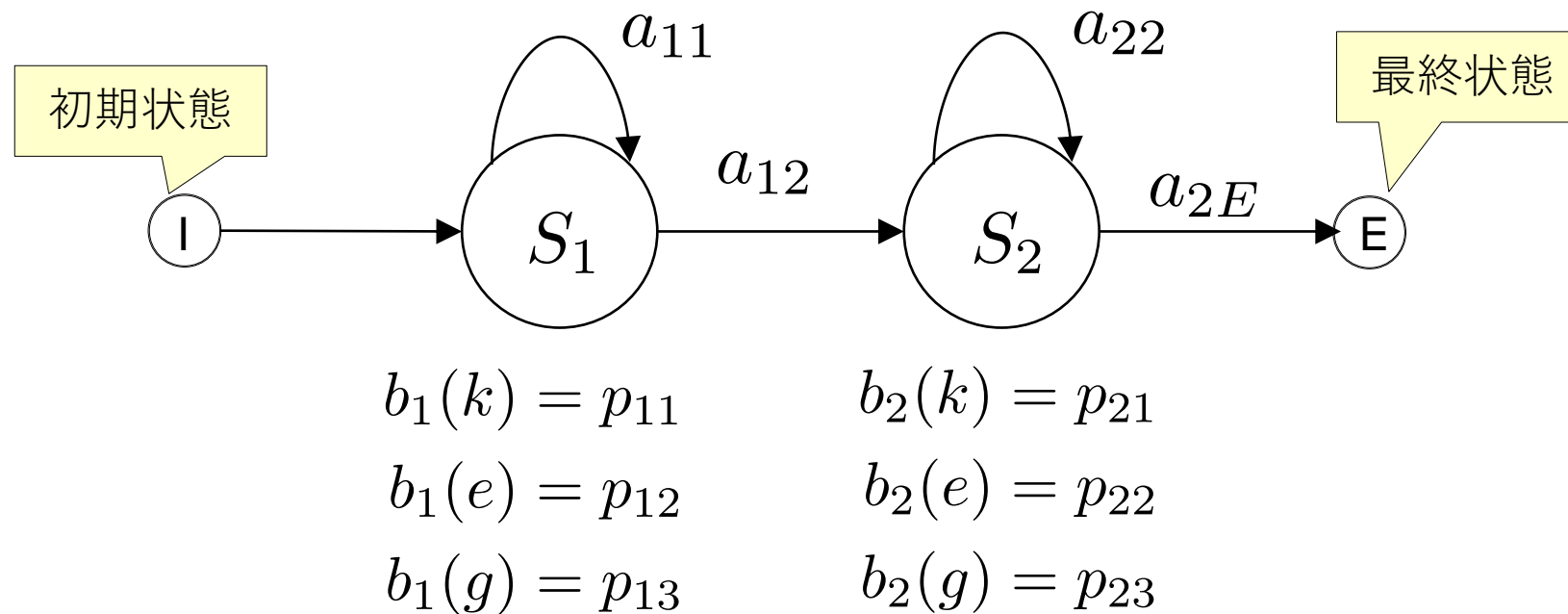
尤度は、あるクラスの確率モデルを、他のクラスとは無関係に求めている

尤度

事前確率

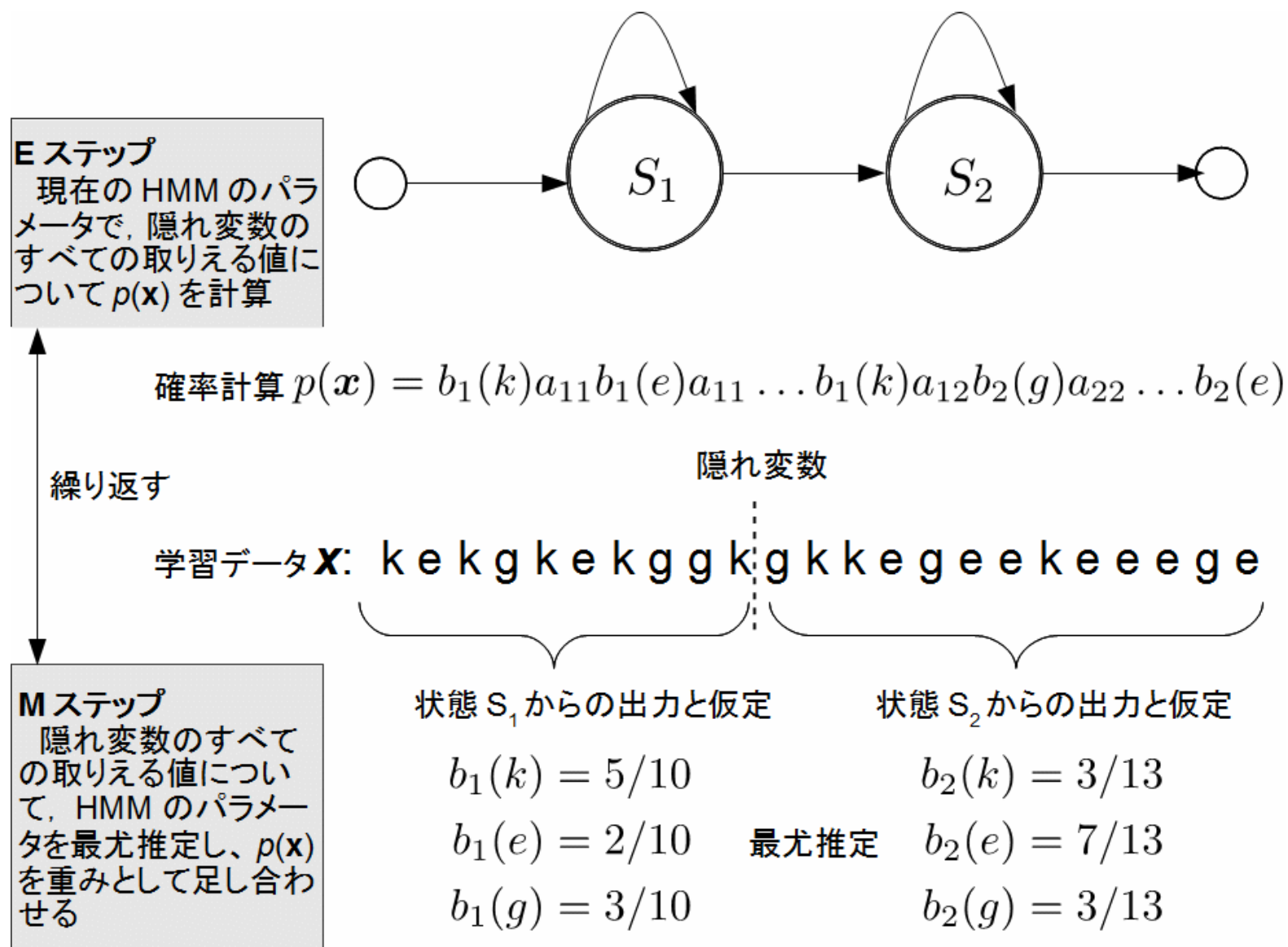
## 13.3 系列識別問題 — HMM —

- 不定長入力に対する尤度計算法
  - ◆ 自己遷移を持つ確率オートマトンを用いる
  - ◆ 尤度計算はビタビアルゴリズム



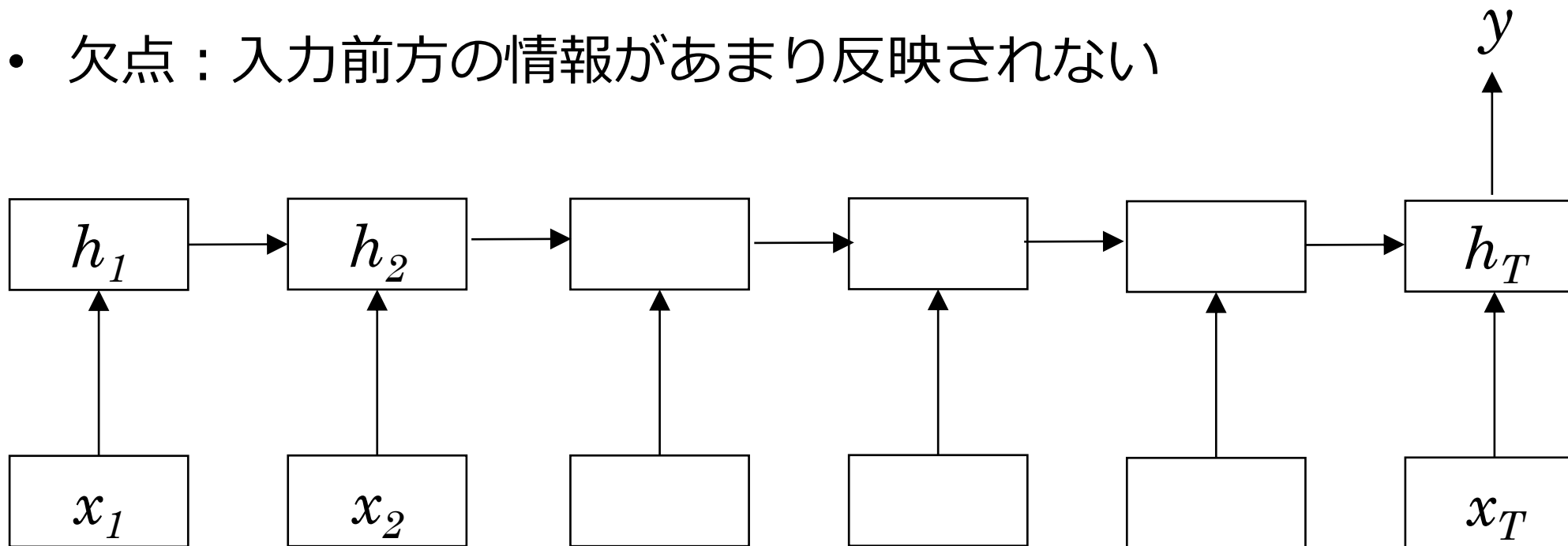
## 13.3 系列識別問題 — HMM —

### • HMMの学習：EMアルゴリズム



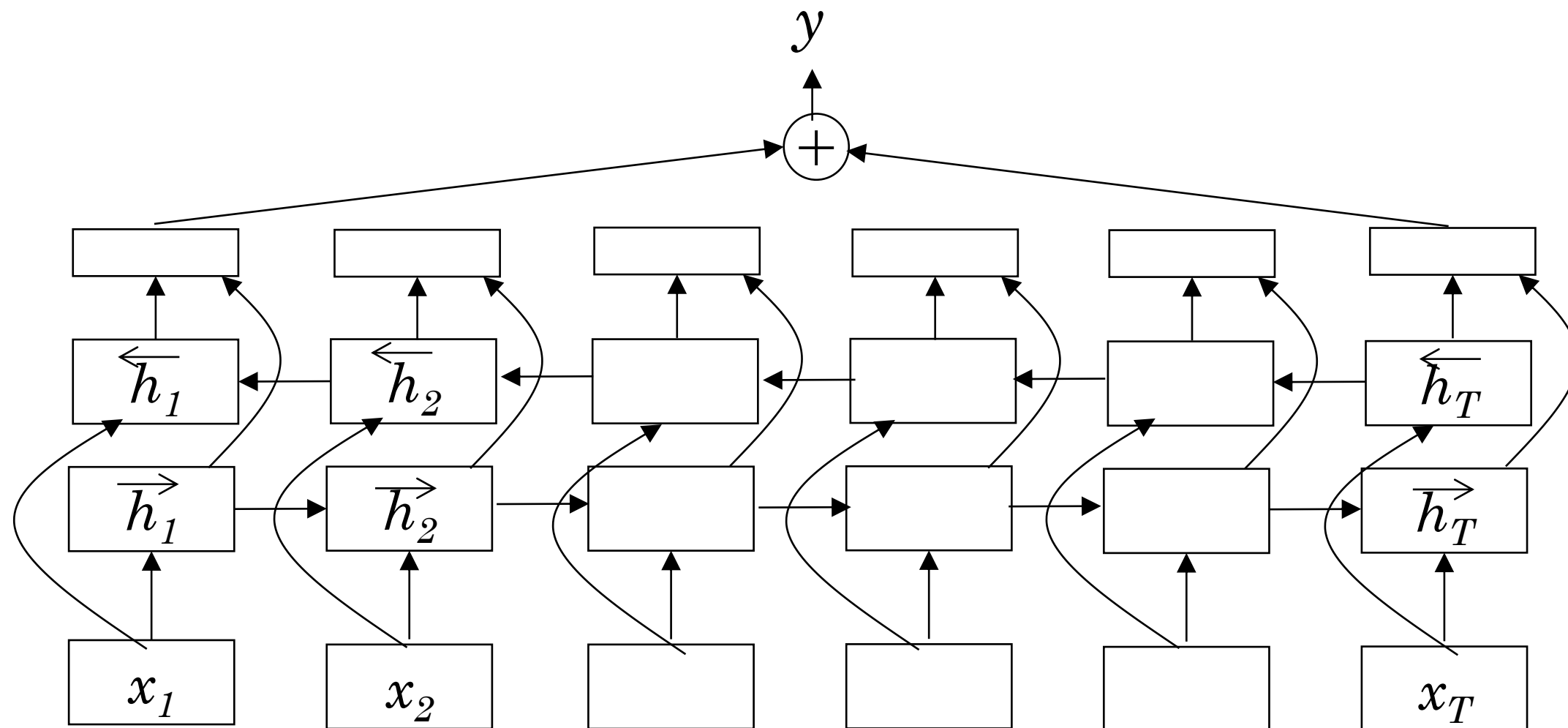
## 13.3 系列識別問題 — RNN —

- RNNによる系列識別
  - ◆ 隠れ層にはLSTMやGRUを使う
  - ◆ 最終入力以外の出力は使わない
  - ◆ 最終入力に対する出力を識別結果とする
    - 欠点：入力前方の情報があまり反映されない



## 13.3 系列識別問題 — RNN —

- bidirectional RNNによる系列識別
  - ◆ 前向き・後向きそれぞれの最終状態を結合



# 系列変換問題

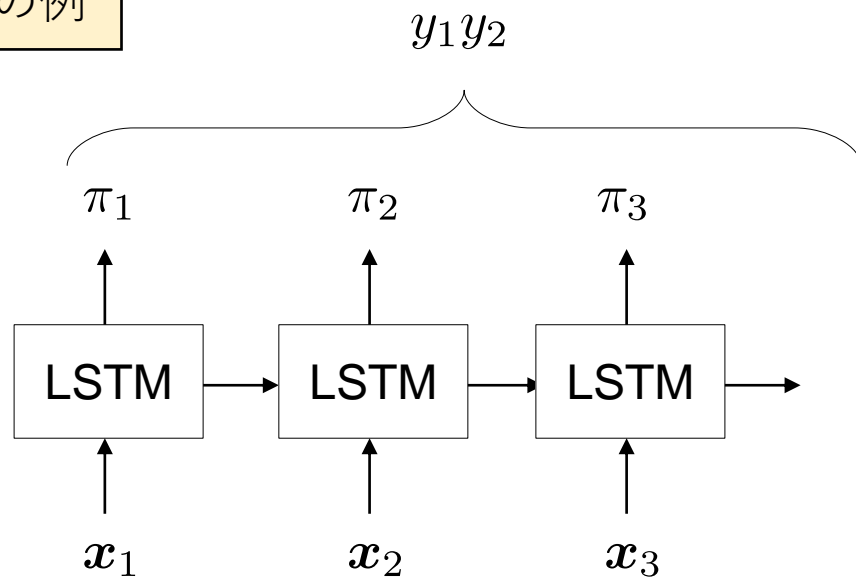
- 系列変換問題の定式化
  - ◆ 入力系列  $\mathbf{x} = \mathbf{x}_1, \dots, \mathbf{x}_T$
  - ◆ 出力系列  $\mathbf{y} = y_1, \dots, y_L$ 
    - 一般に  $T > L$
  - ◆ 系列処理と探索を組み合わせた複雑な処理が必要
- End-to-Endアプローチ
  - ◆ 入力から出力への変換をニューラルネットワークで学習
    - CTC
    - Encoder-Decoderモデル



# 系列変換問題

- CTC (Connectionist Temporal Classification)
  - ◆ 出力記号にblank記号\_を加えて、入力長と出力長を合わせる
  - ◆ 正解系列に変換可能な出力系列の確率の和を求める

音声認識の例



- haiという正解系列に対して

\_h\_\_a\_\_i  
hhh\_aaaa\_i  
hh\_\_\_\_\_ai\_\_

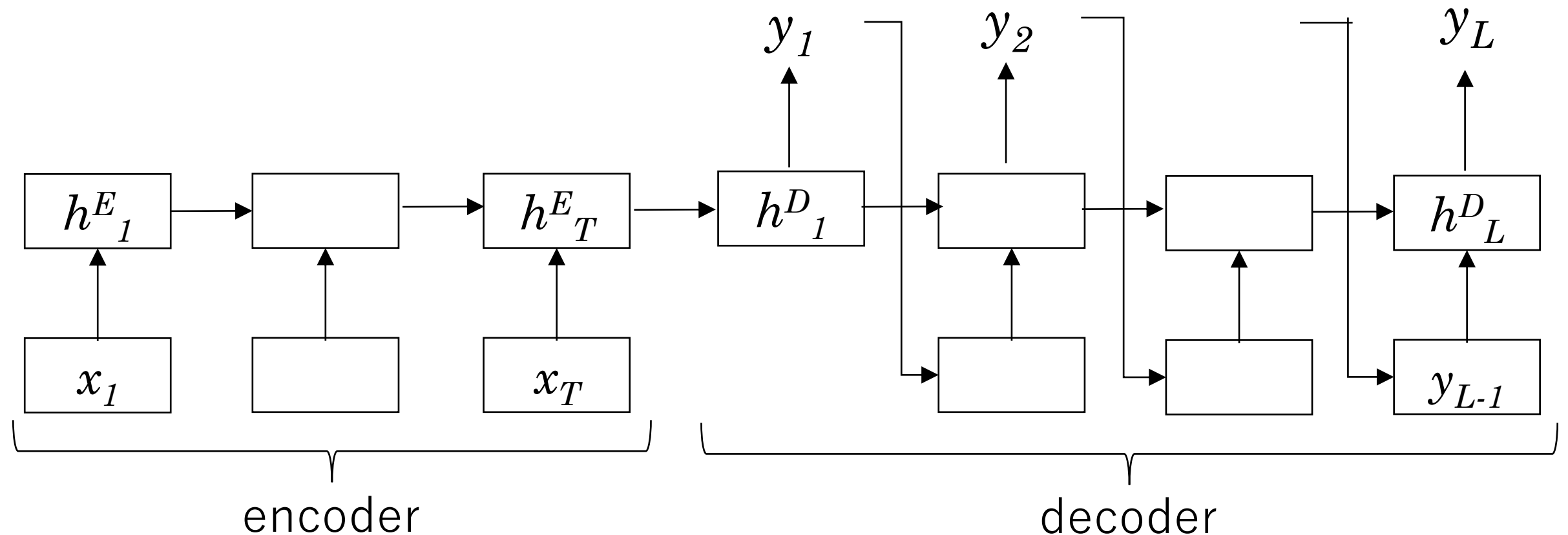
などの出力系列をすべて正解とみなす

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\boldsymbol{\pi} \rightarrow \mathbf{y}} p(\boldsymbol{\pi}|\mathbf{x}) = \sum_{\boldsymbol{\pi} \rightarrow \mathbf{y}} \prod_{i=1}^T p(\pi_i|\mathbf{x}_i)$$

# 系列変換問題

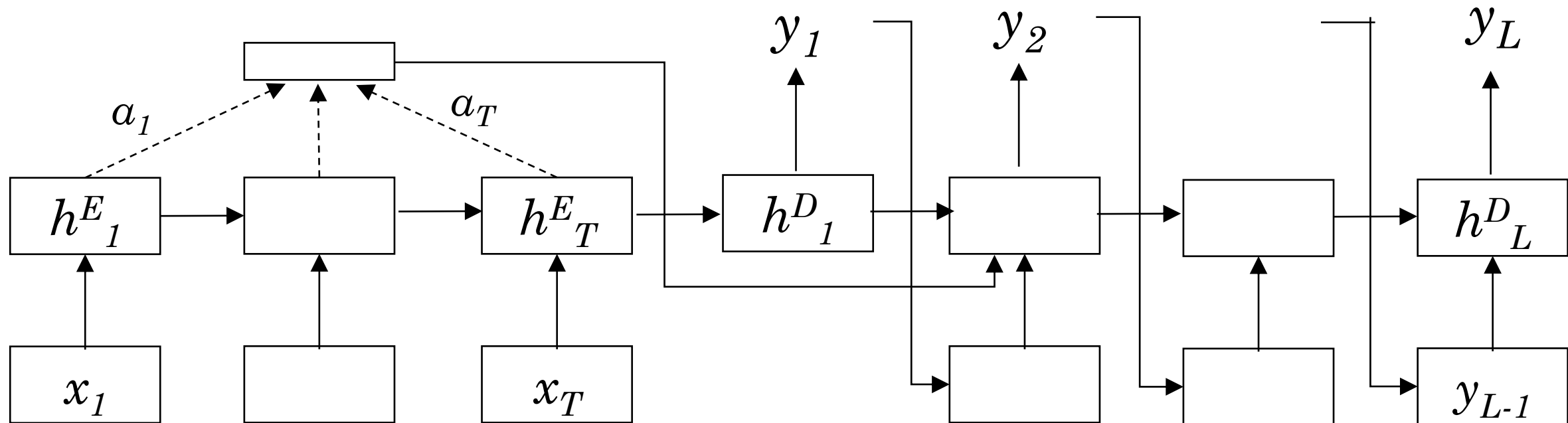
- Encoder-Decoder方式

- ◆ encoder: 入力情報を最後の隠れ層のベクトル  $h^E_T = C$  で表現
- ◆ decoder:  $C$  を初期ベクトルとして出力を逐次生成



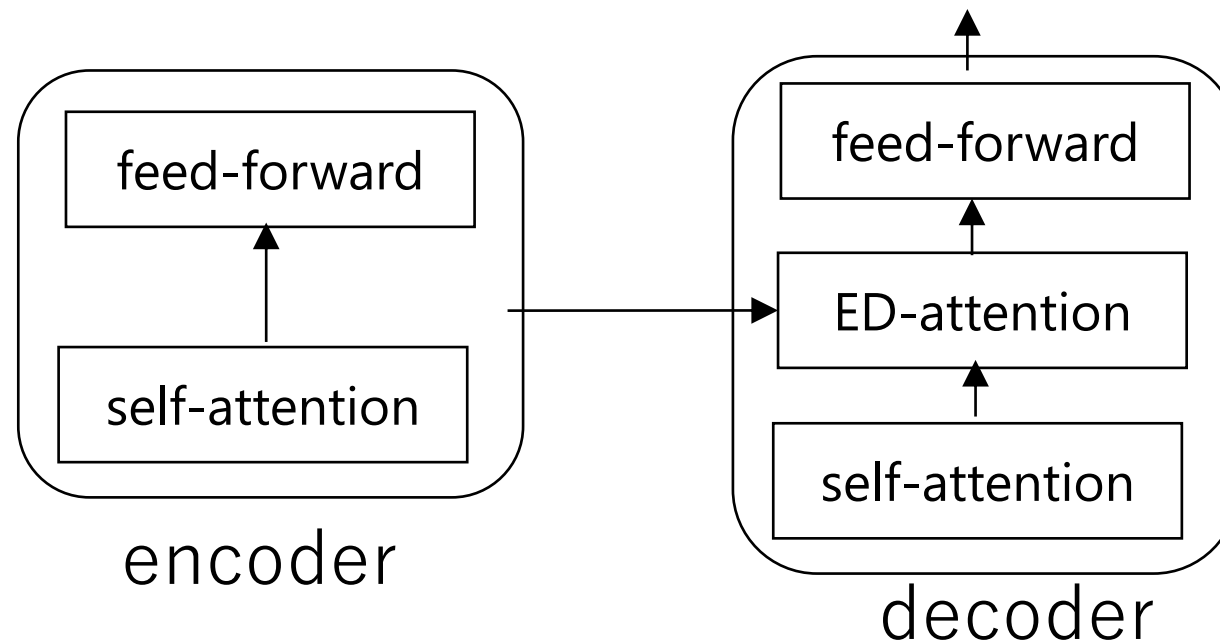
# 系列変換問題

- Encoder-Decoder + Attention
  - ◆ 特定の時刻の出力記号を決めるときに、入力の関連する部分の情報を参照する
  - ◆ decoderの入力に、encoderの隠れ層ベクトルの重み付き和を結合する



# 系列変換問題

- Transformer
  - ◆ 入力系列全体からself-attentionを使って各単語の隠れ層の表現を得る処理を複数段繰り返す
  - ◆ 出力はこれまでの出力系列のself-attentionとencoder-decoder間のattentionを用いた処理を複数段繰り返す



参考  
<http://jalammar.github.io/illustrated-transformer/>

# まとめ

- ラベル系列に対する識別問題
  - ◆ 入力の系列長と出力の系列長が等しい
    - 識別モデルCRFが有効
  - ◆ 入力の系列長に関わらず出力の系列長が1
    - HMMで可変長系列の処理が可能
    - RNNやTransformerも有効
  - ◆ 入力の系列長と出力の系列長に対応関係がない
    - RNNやTransformerが有効