HMM Theory and Practice

Training & Recognition

- Major advantage of HMMs is the availability of a ‘toolkit’ of powerful, well-founded mathematical methods for HMM manipulation
- The **Baum-Welch** algorithm is used to train the parameters of a set of HMMs given a set of training data
- **Viterbi Decoding** is used to classify an unknown speech pattern in terms of the sequence of HMMs which is most likely to have produced it

The Recognition Problem

- Given a sequence of acoustic feature vectors

  $Y = \{y_1,\ldots,y_T\}$

  we want to find the sequence of words

  $W = \{w_1,\ldots,w_L\}$

  such that the probability

  $P(W \mid Y)$

  is maximized.
- If $M = \{M_1,\ldots,M_K\}$ is the sequence of HMMs which represents W, then $P(W \mid Y) = P(M \mid Y)$

Bayes' Theorem

- Computation of the probability *P(M | Y)* is made possible using **Bayes' Theorem**

$$P(W \mid Y) = \frac{p(Y \mid W)P(W)}{p(Y)}$$

- *P(W)* is the "language model probability"
- *p(Y | W)* is the "acoustic model probability"
- Bayes Theorem has been referred to as the "fundamental theorem of speech recognition"!

The Baum-Welch Algorithm

- The Baum-Welch algorithm is the method which is normally used for HMM parameter estimation
- Given a set of HMMs M_0 and a set of speech patterns Y, Baum's theorem defines how to produce a new model set M_1 such that

$$P(Y \mid M_1) \geq P(Y \mid M_0)$$

- Baum-Welch algorithm applies this method repeatedly until a HMM M_n is found which (locally) maximizes $P(Y \mid M)$
- Baum's theorem only valid for particular classes of state output PDF

Notes on B-W Reestimation

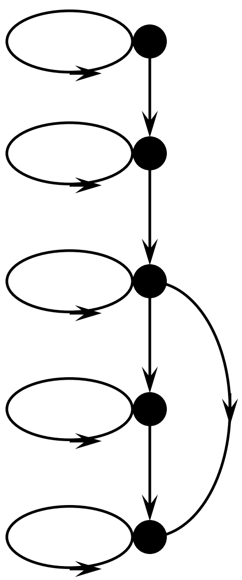
- The Baum-Welch algorithm is only guaranteed to find a **locally** optimal HMM set - hence choice of M_0 can be important
- Baum-Welch is a **supervised** training algorithm which requires labelled speech data
- The labelling need **not** be at the same level as the HMM set - phoneme level HMMs can be trained using data labelled orthographically at the phrase or sentence level
- For large applications B-W reestimation can be **very** computationally expensive

Viterbi Decoding

- Viterbi Decoding is the algorithm which is used to find the sequence of HMMs which is most likely to have generated a given speech pattern
- Based on **Dynamic Programming**
- Viterbi Decoding illustrates the type of computation typically done with HMMs

Viterbi Decoding (1)

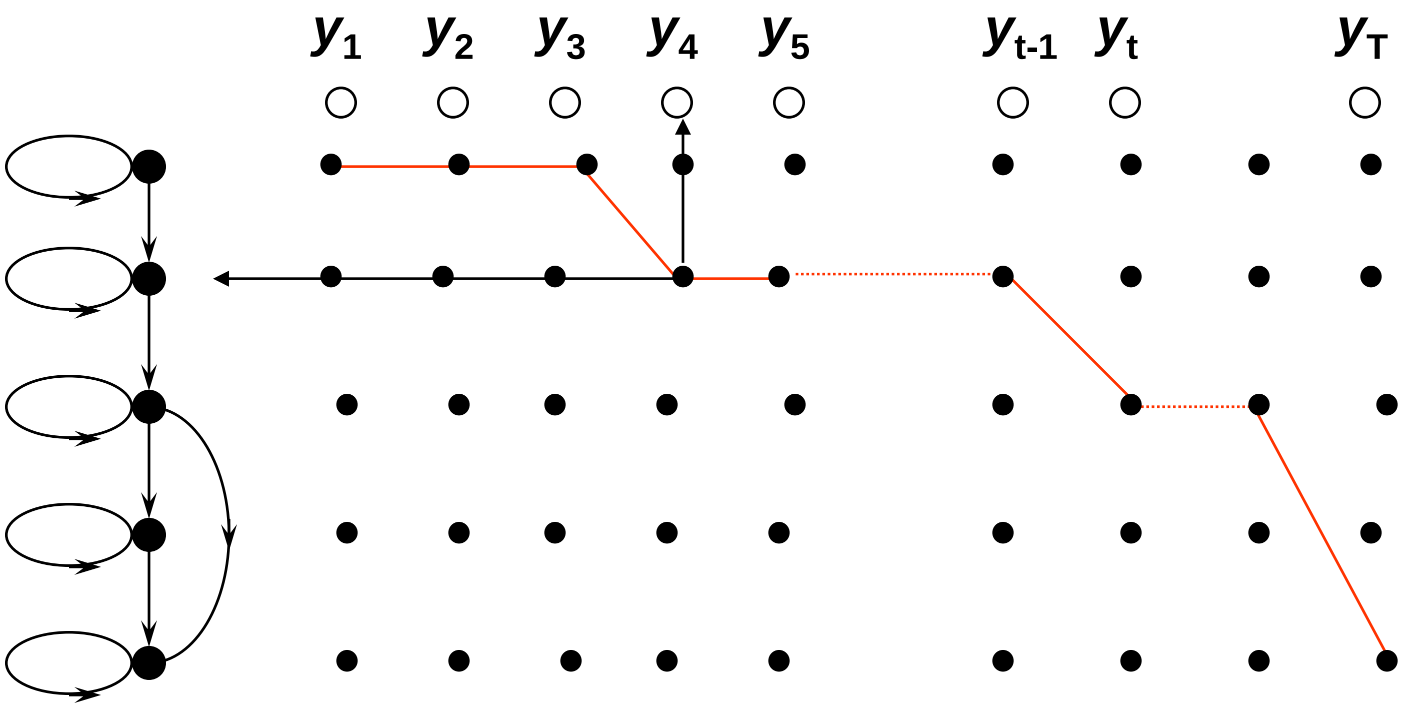
y_1 y_2 y_3 y_4 y_5 y_{t-1} y_t y_T



Q: How can *M* have generated *Y*?

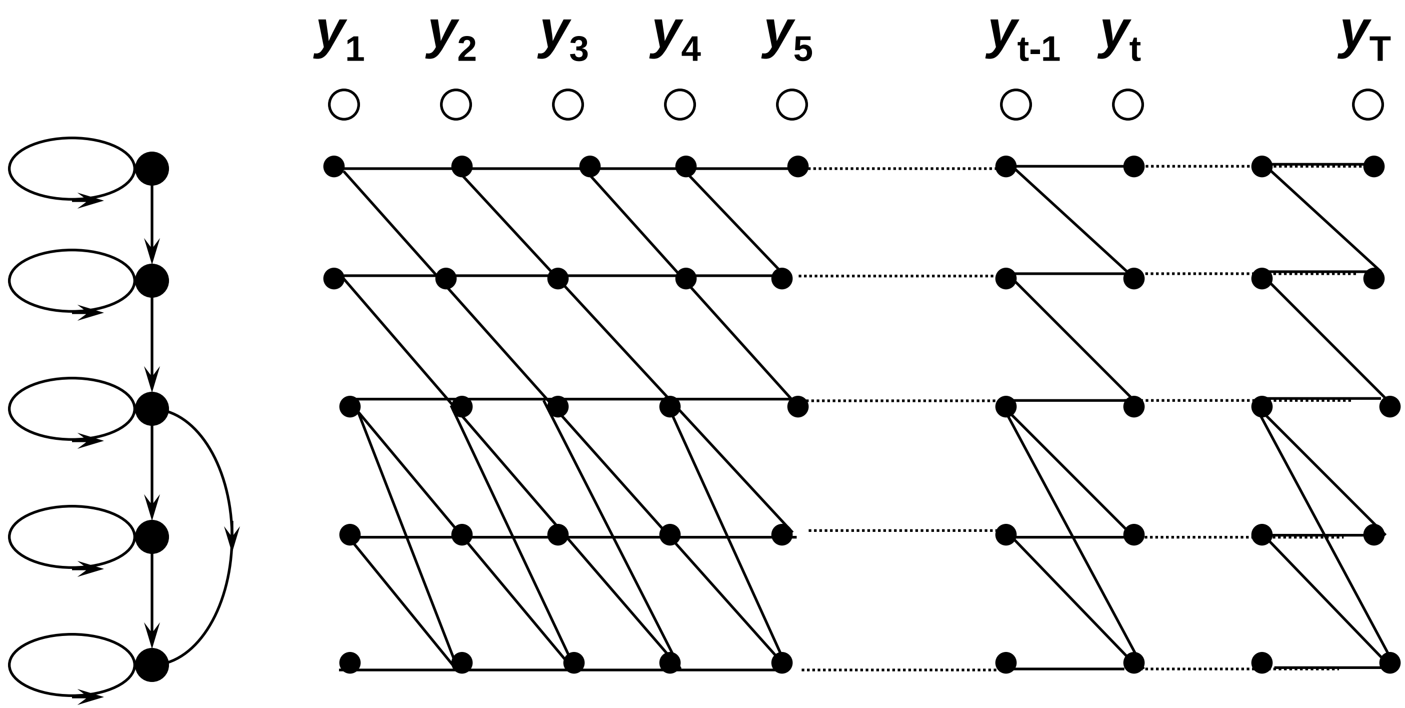
A: Via a state sequence of length T

Function of State Sequence



Viterbi Decoding (2)

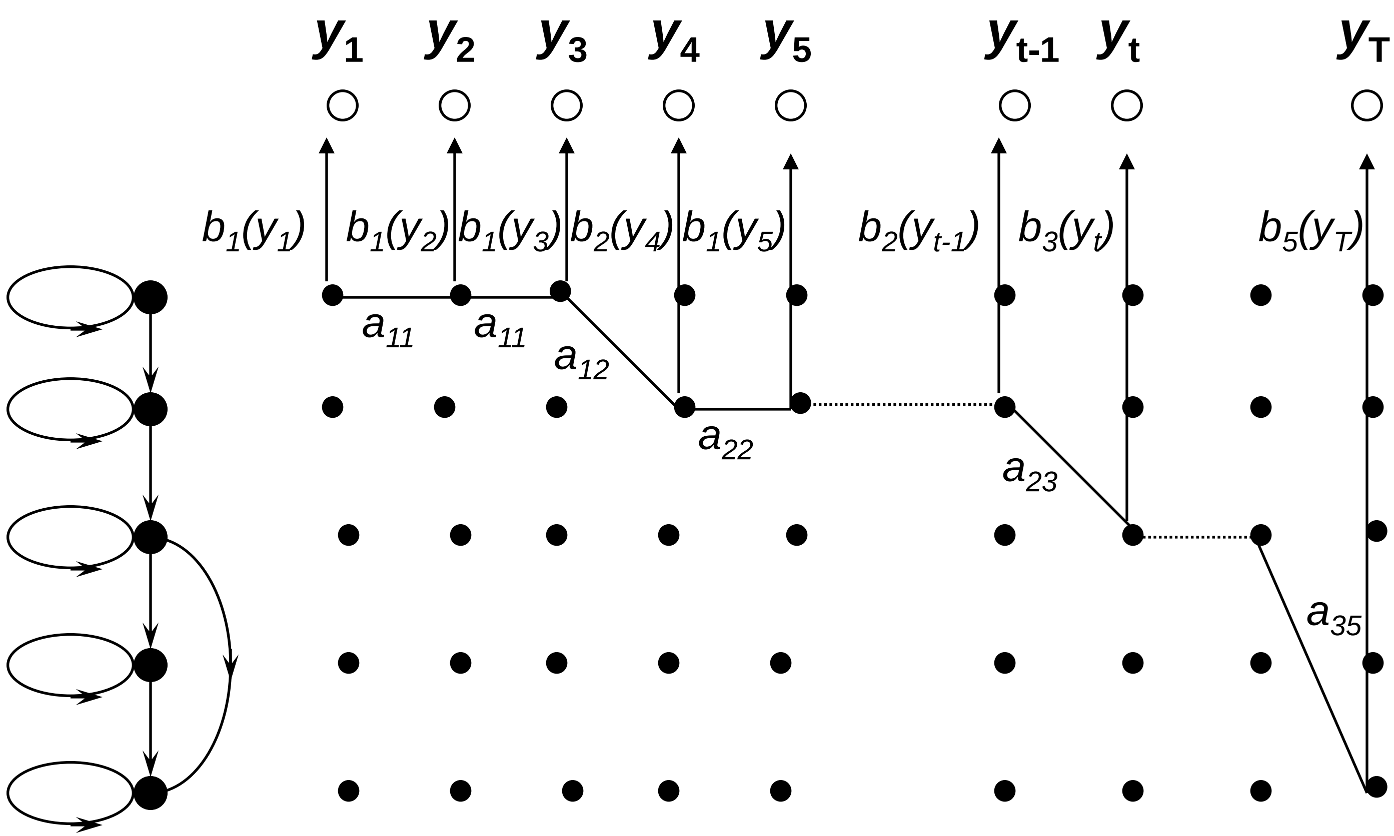
- **Construction of 'state-time trellis'**



Constructing the State-Time Trellis

- Simple Rule:
  - Connect node (*i,t*) of the trellis to node (*j,t+1*) if and only if there is a transition between state *i* and state *j* in the HMM with probability a_{ij} greater than zero

Basic Probability Calculation



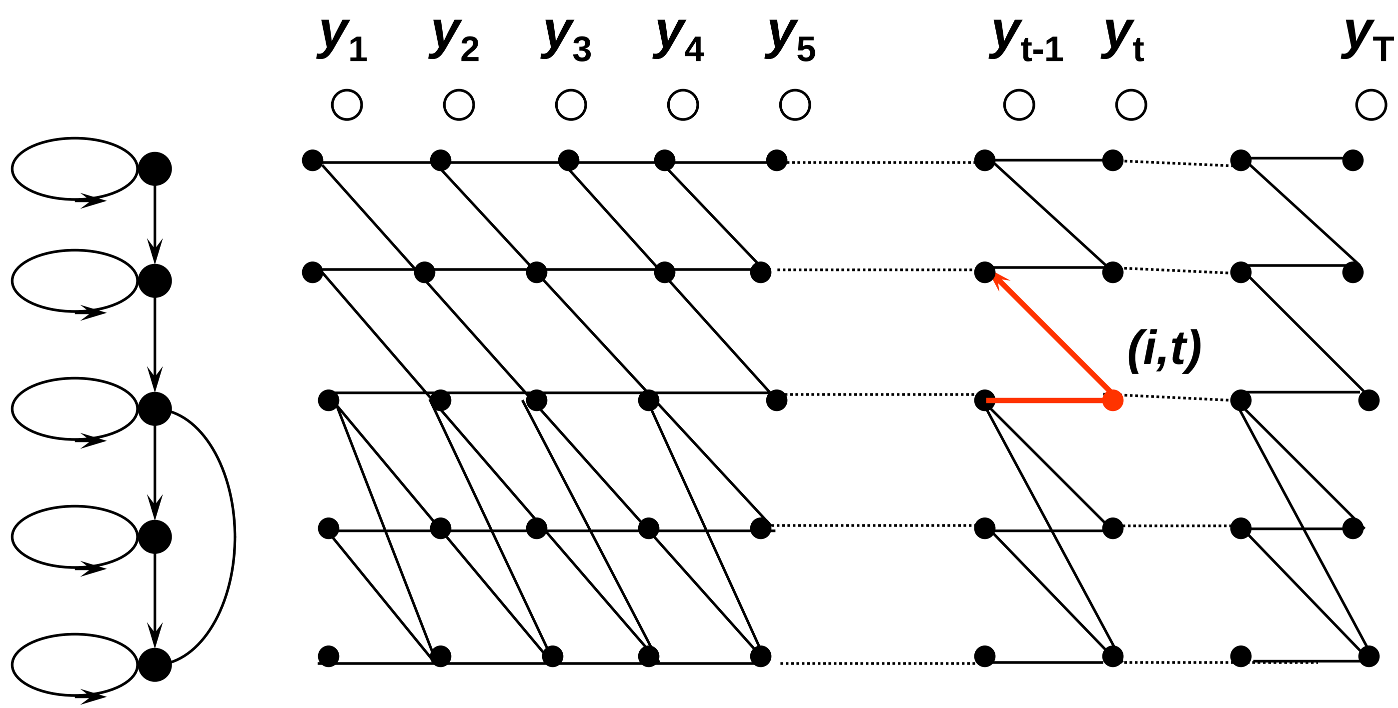
Viterbi Decoding (3)

- Let $X = \{x_1, \ldots, x_T\}$ be a state sequence of length T
- The joint probability of Y and X is given by:

$$p(Y, X) = b_{x_1}(y_1) \prod_{t=2}^{T} a_{x_{t-1} x_t} b_{x_t}(y_t)$$

- i.e. the product of the state-output and state transition probabilities along the state sequence
- The <u>optimal</u> state sequence is the sequence X such that $p(Y,X)$ is maximized
- $p(Y)$ is the sum of $P(Y,X)$ over all sequences X

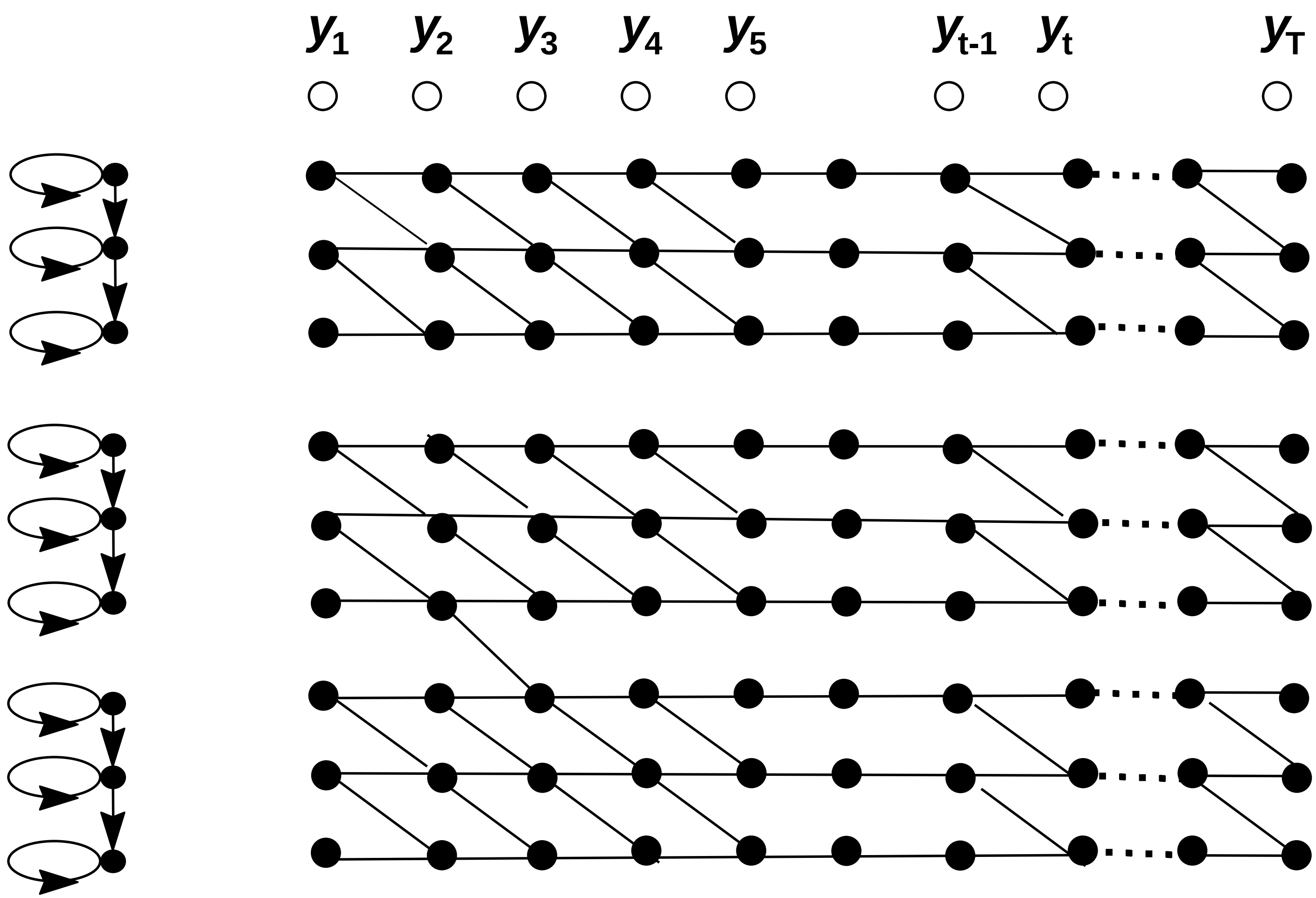
Viterbi Decoding (4)



$p_t(i)$ = Prob($y_1,\ldots,y_t$, opt sequence to (i,t))

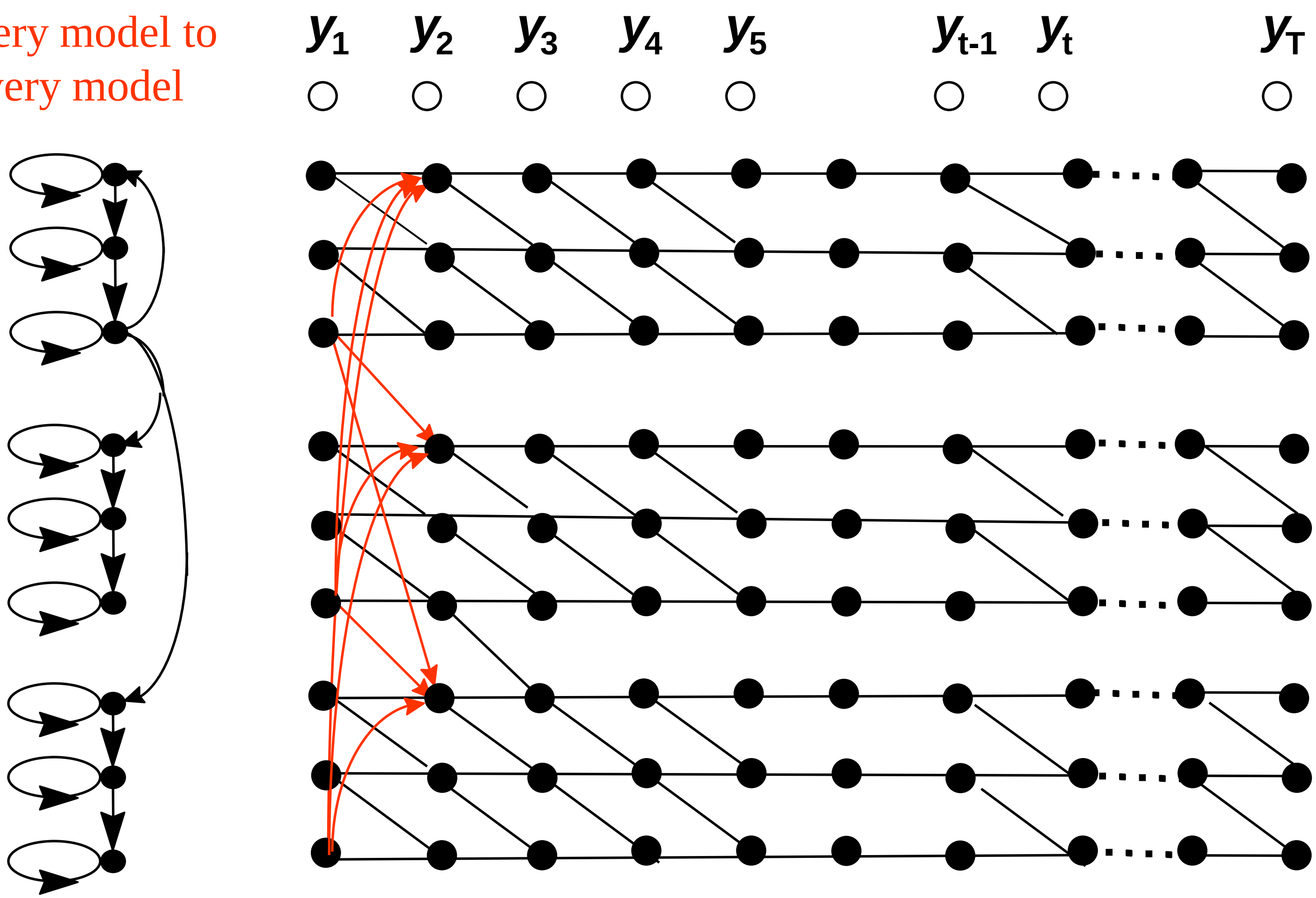
$p_t(i) = \max \{p_{t-1}(i-1)a_{i-1,i}\ ,\ p_{t-1}(i)a_{i,i}\}b_i(y_t)$

Isolated Speech Recognition

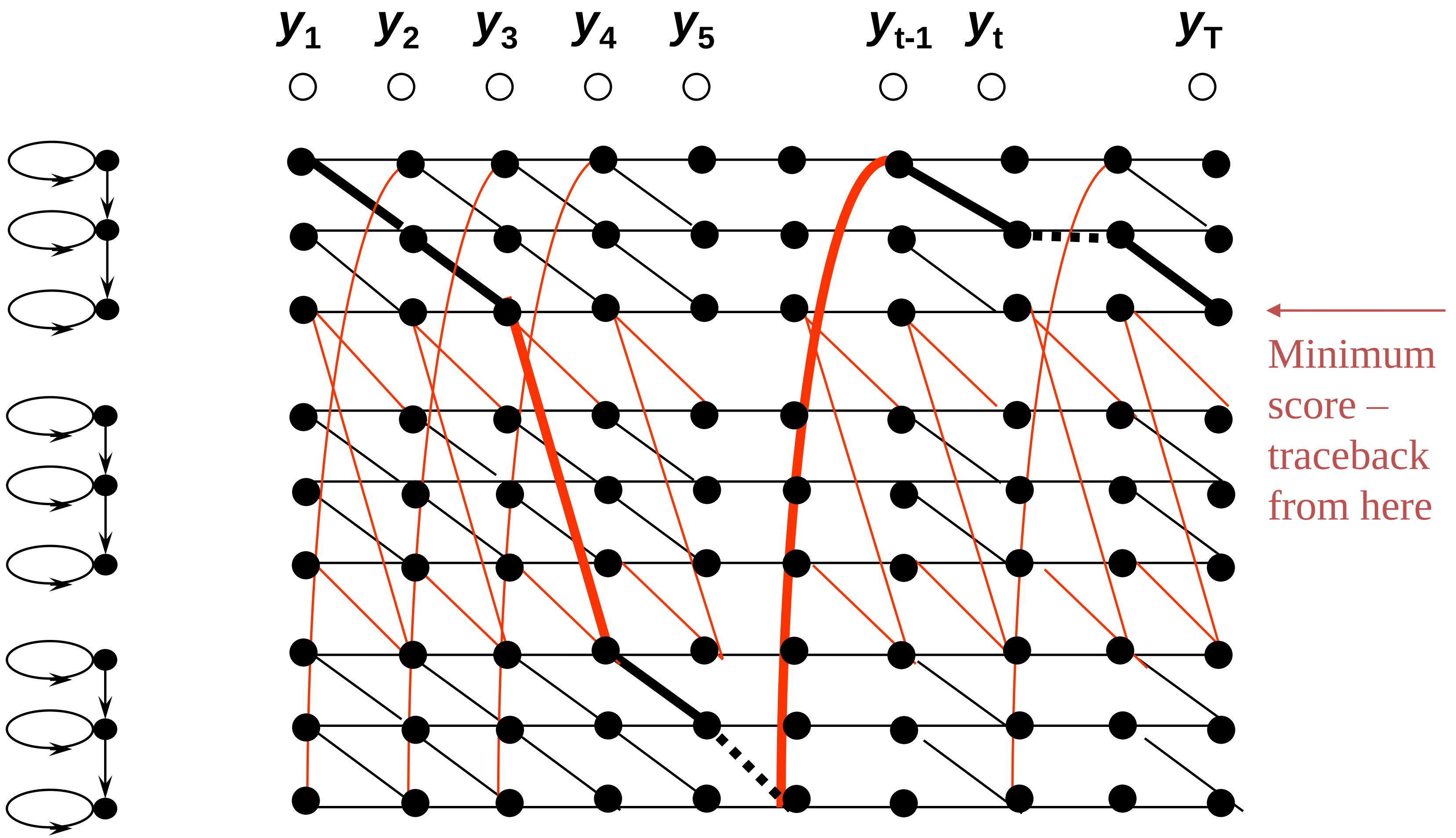


Connected Speech Recognition

New transitions connect end of every model to start of every model



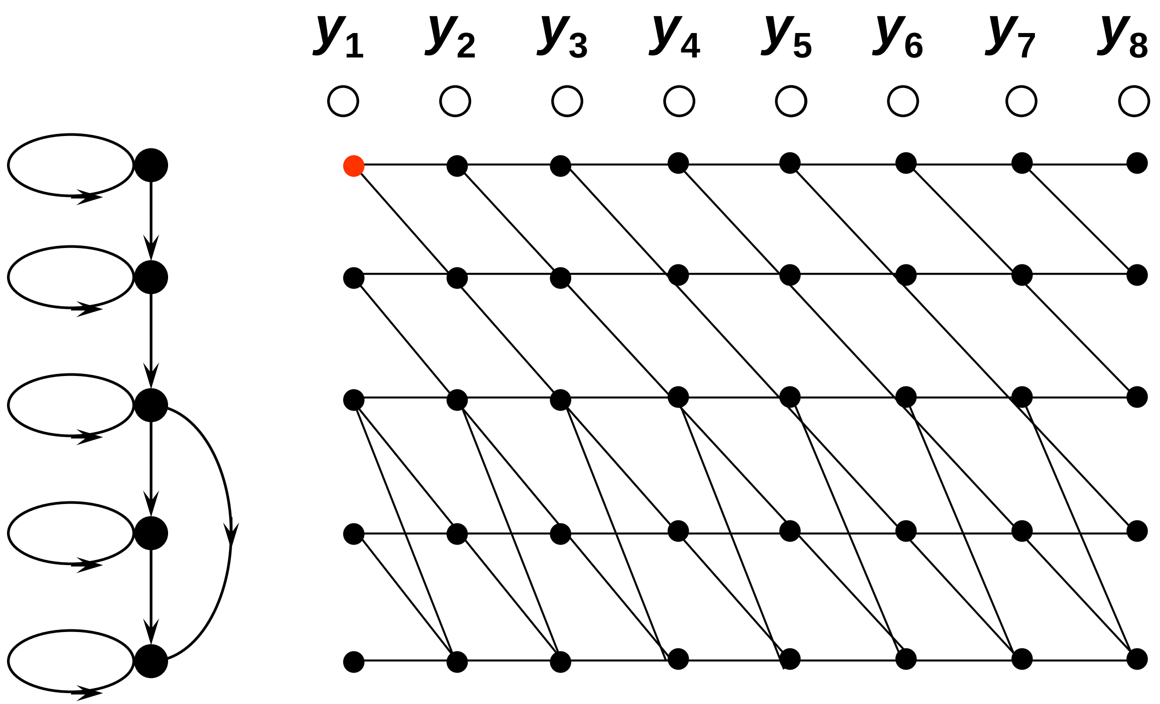
Connected Speech Recognition



Viterbi Decoding

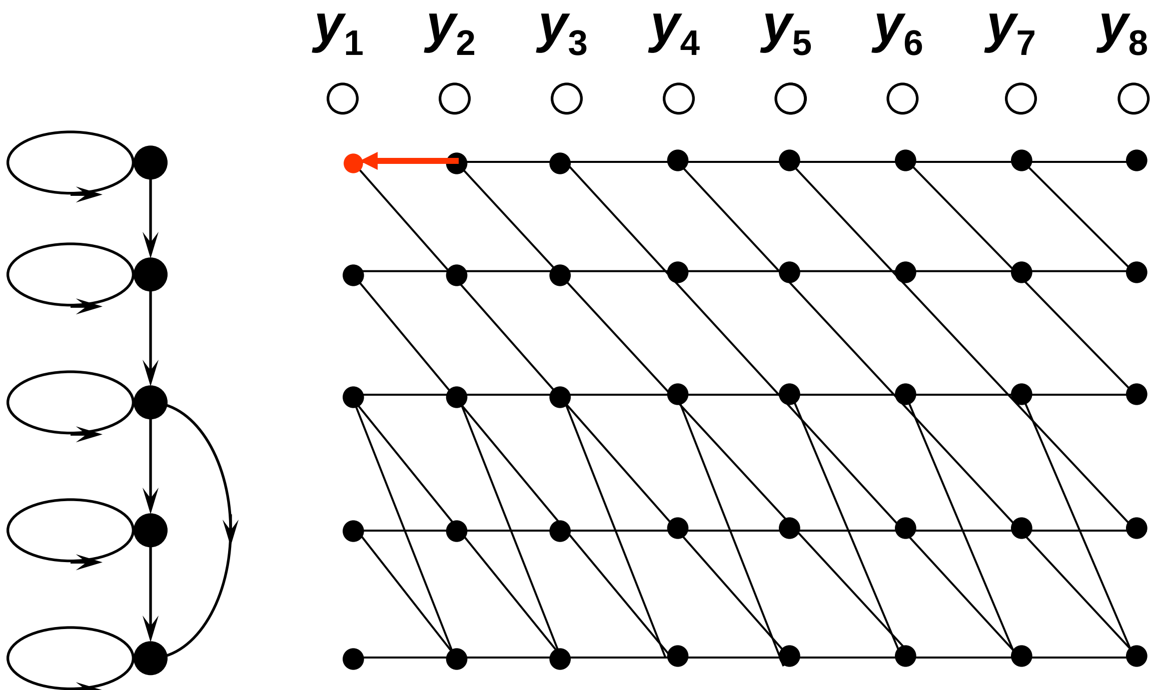
Further explanation of Viterbi decoding

Viterbi Decoding



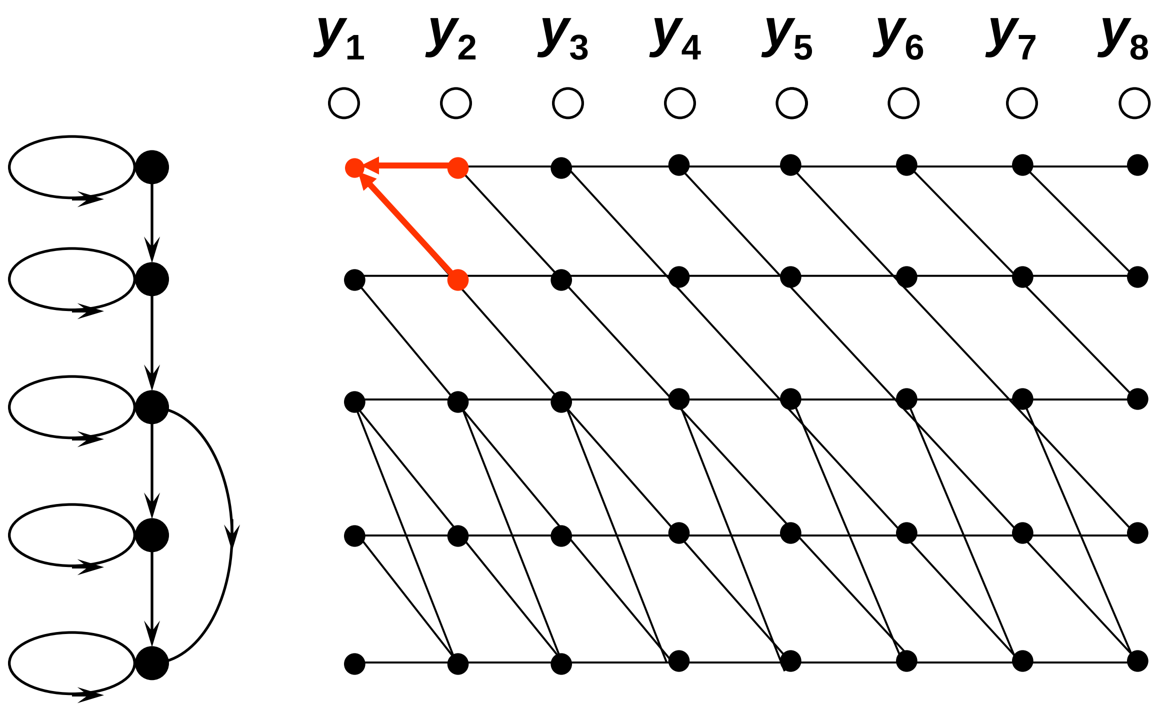
$$\alpha_1(1) = b_1(y_1)$$

Viterbi Decoding



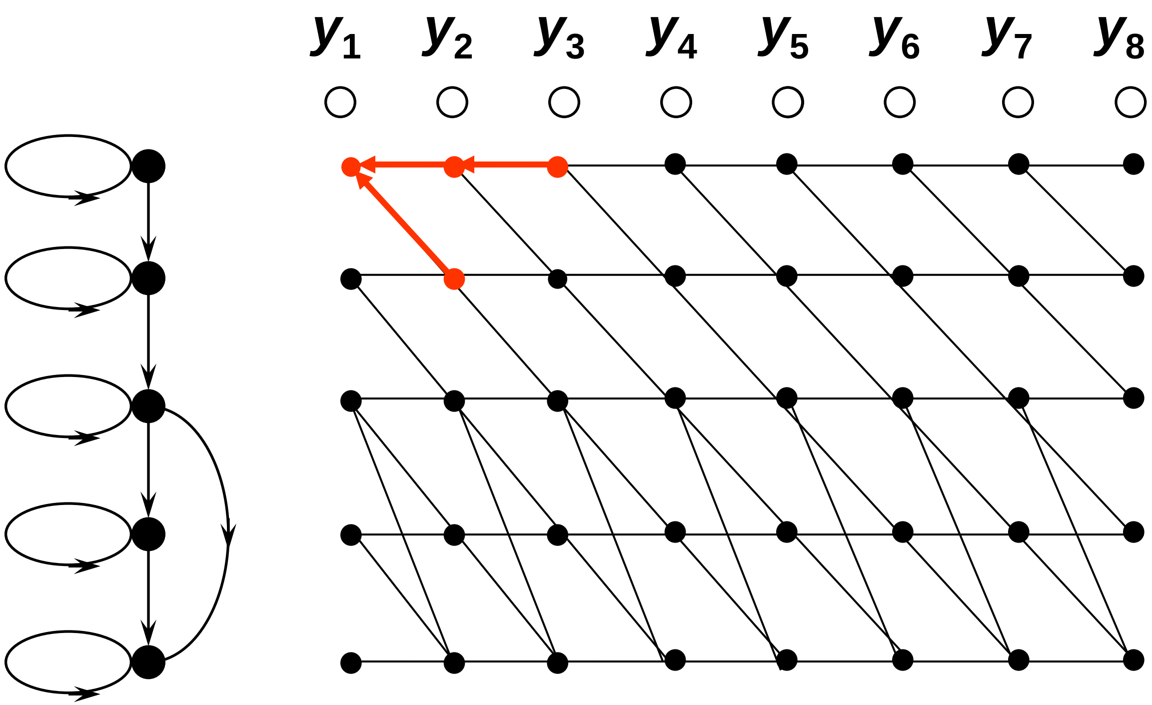
$$\alpha_2(1) = \alpha_1(1) a_{11} b_1(y_2)$$

Viterbi Decoding



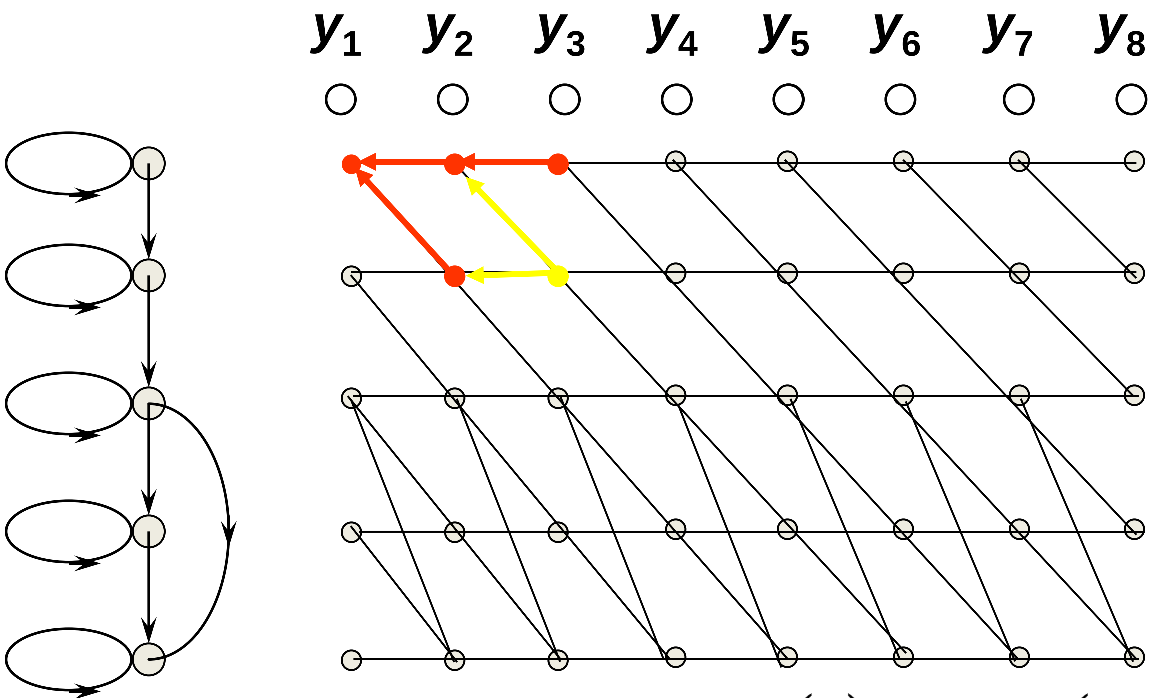
$$\alpha_2(2)=\alpha_1(1)a_{12}b_2(y_2)$$

Viterbi Decoding



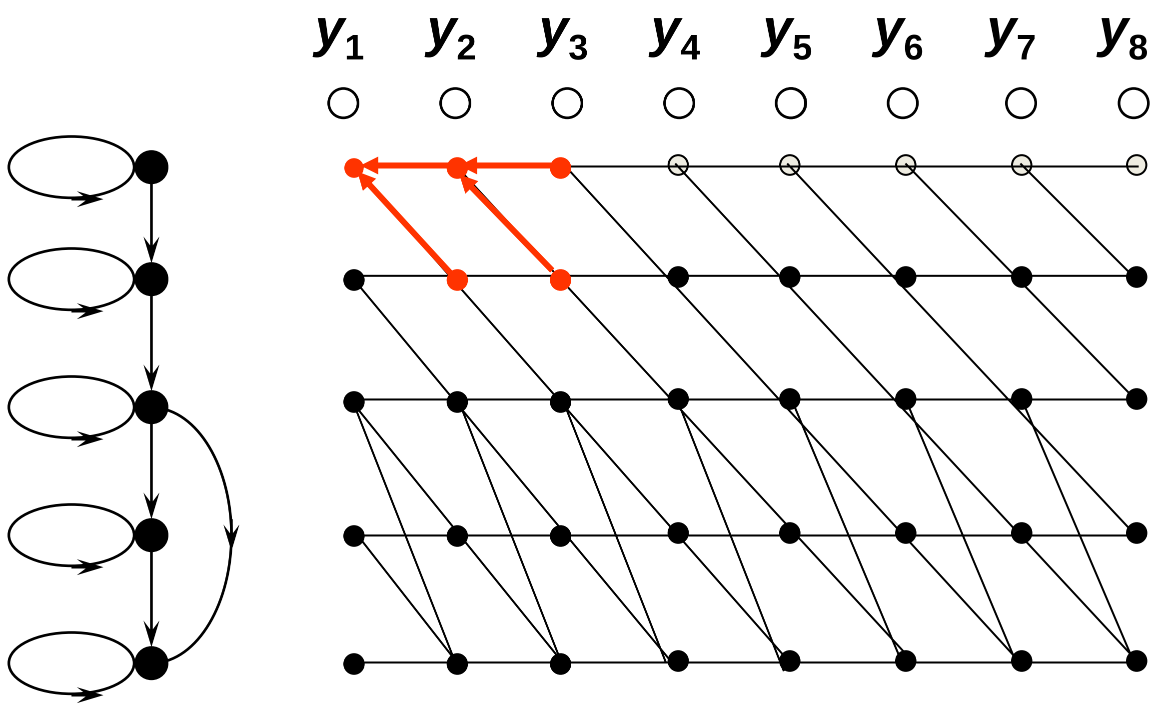
$$\alpha_3(1) = \alpha_2(1) a_{11} b_1(y_3)$$

Viterbi Decoding



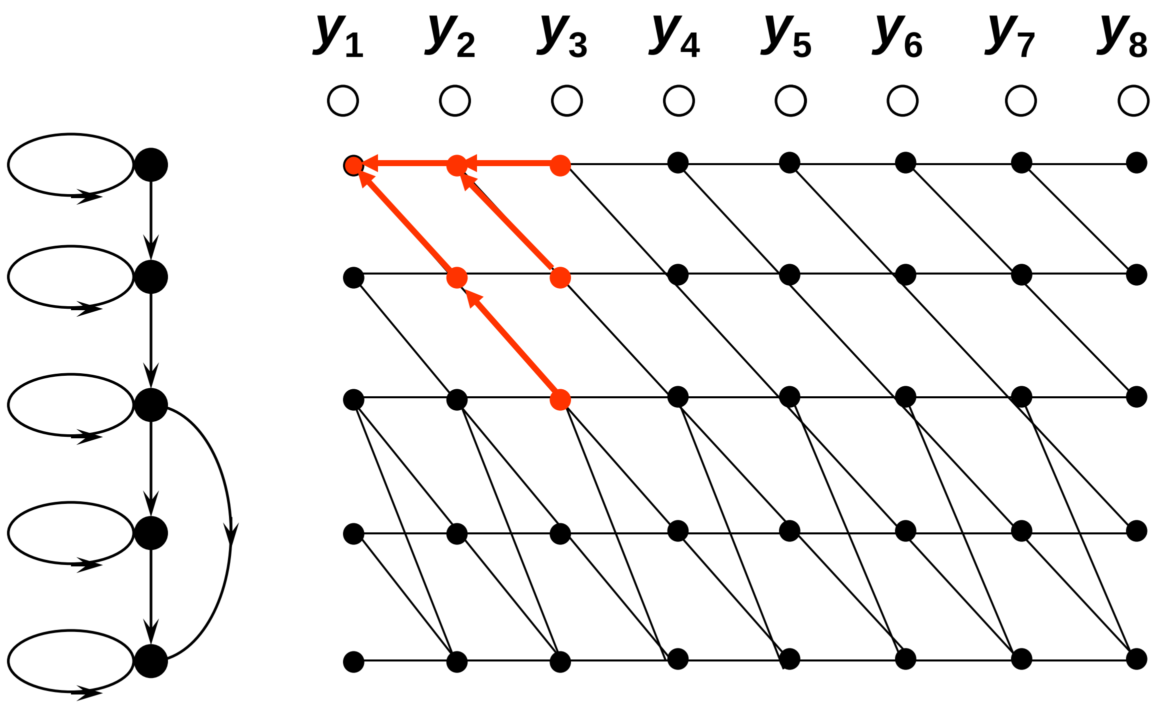
$$\alpha_3(2) = \max \begin{cases} \alpha_2(1)a_{12}b_2(y_3) \\ \alpha_2(2)a_{22}b_2(y_3) \end{cases}$$

Viterbi Decoding



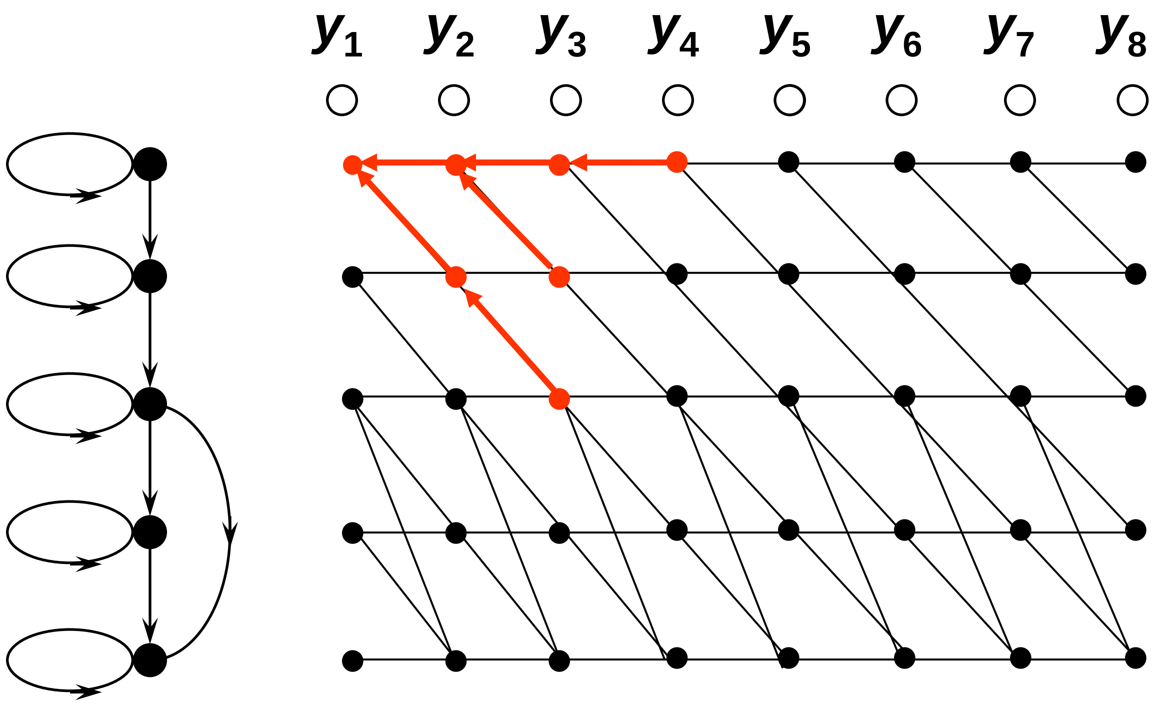
$$\alpha_3(2) = \max \begin{cases} \alpha_2(1) a_{12} b_2(y_3) \\ \cancel{\alpha_2(2) a_{22} b_2(y_3)} \end{cases}$$

Viterbi Decoding

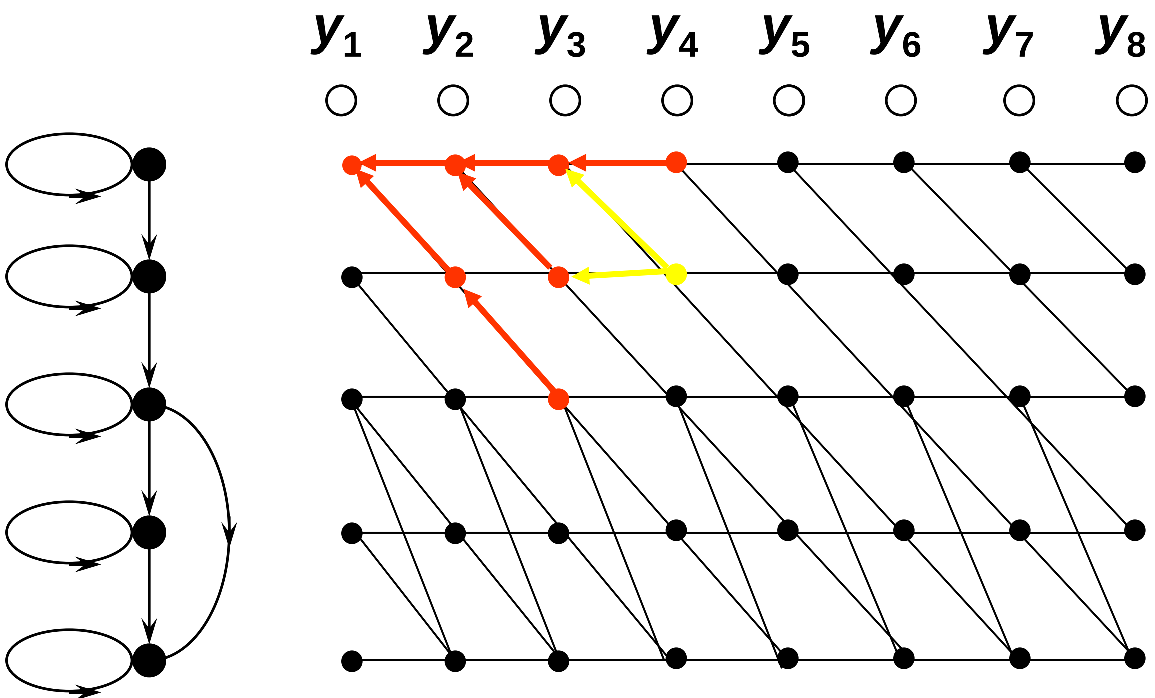


$$\alpha_3(3) = \alpha_2(2) a_{23} b_3(y_3)$$

Viterbi Decoding

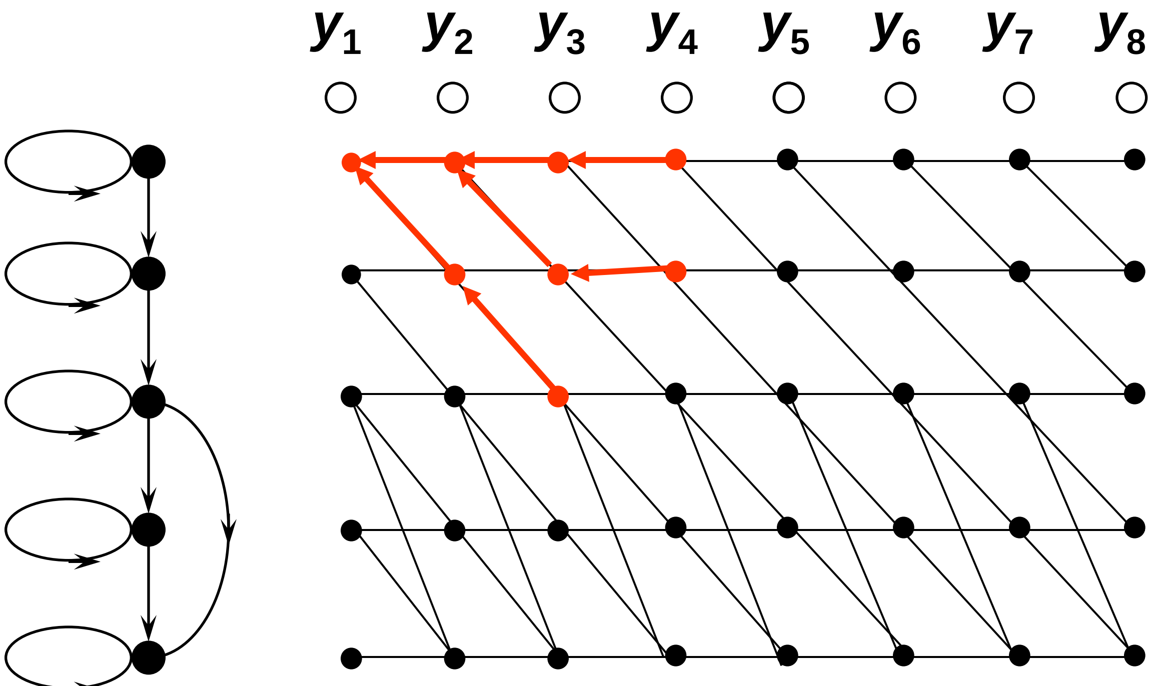


Viterbi Decoding



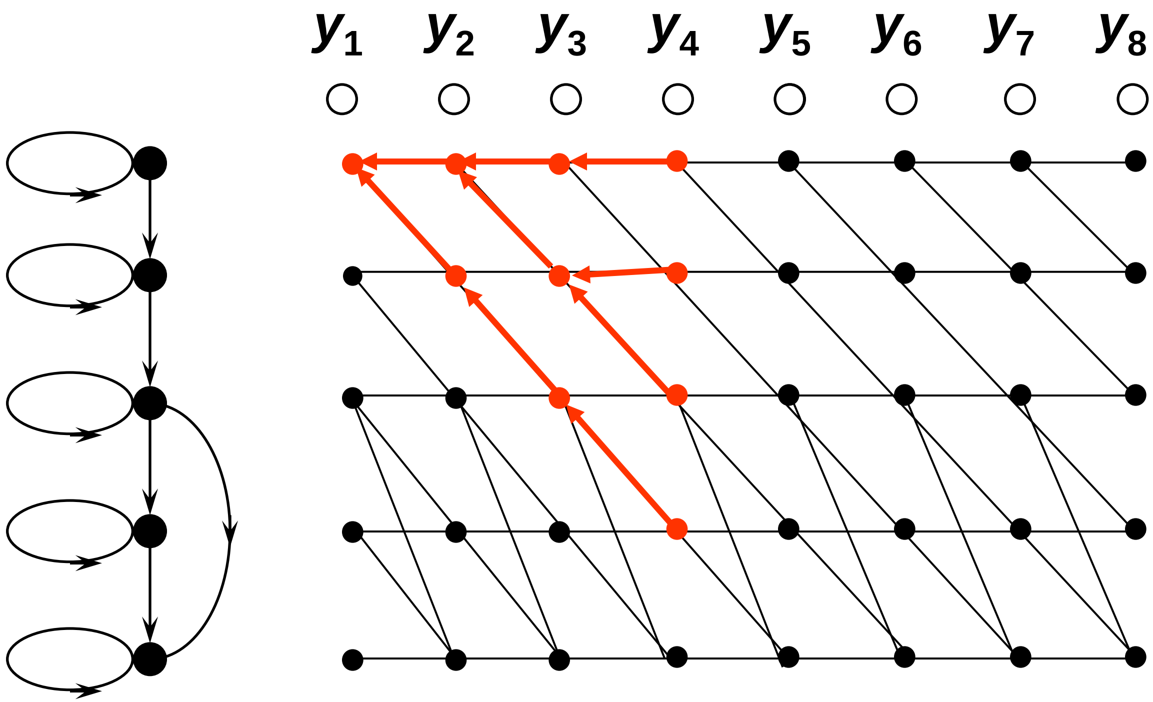
$$\alpha_4(2) = \max \begin{cases} \alpha_3(1) a_{12} b_2(y_4) \\ \alpha_3(2) a_{22} b_2(y_4) \end{cases}$$

Viterbi Decoding

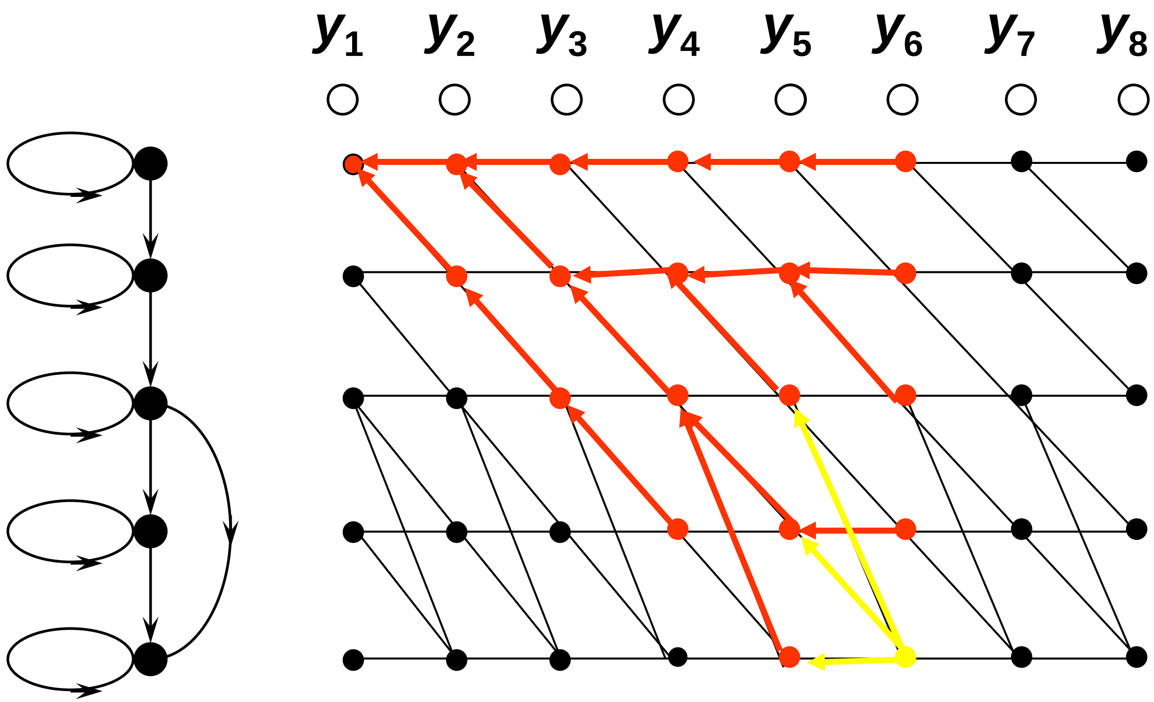


$$\alpha_4(2) = \max \begin{cases} \cancel{\alpha_3(1) a_{12} b_2(y_4)} \\ \alpha_3(2) a_{22} b_2(y_4) \end{cases}$$

Viterbi Decoding

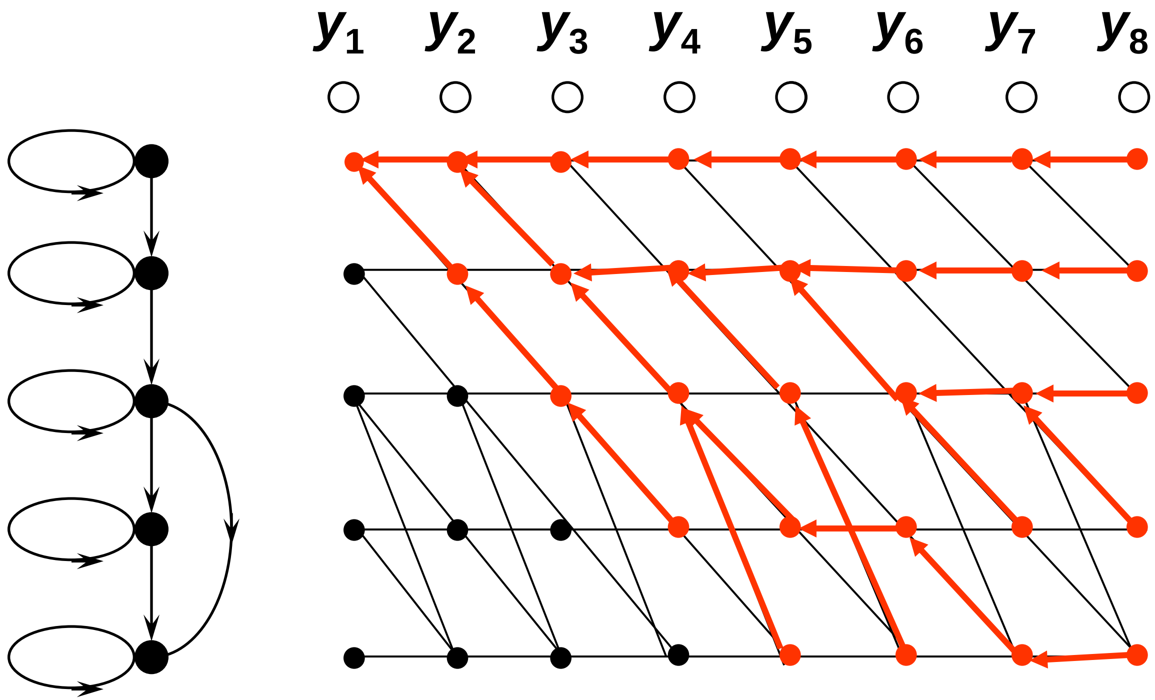


Viterbi Decoding

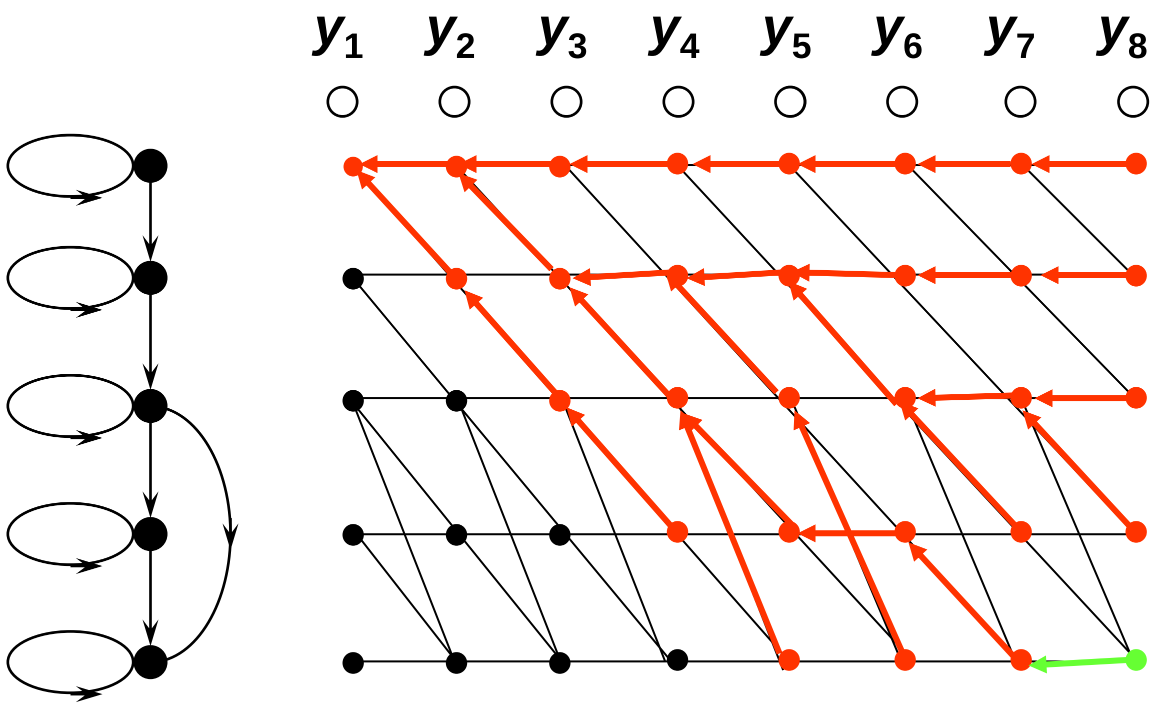


$$\alpha_6(5) = \max \begin{cases} \alpha_5(5) a_{55} b_6(y_6) \\ \alpha_5(4) a_{45} b_6(y_6) \\ \alpha_5(3) a_{35} b_6(y_6) \end{cases}$$

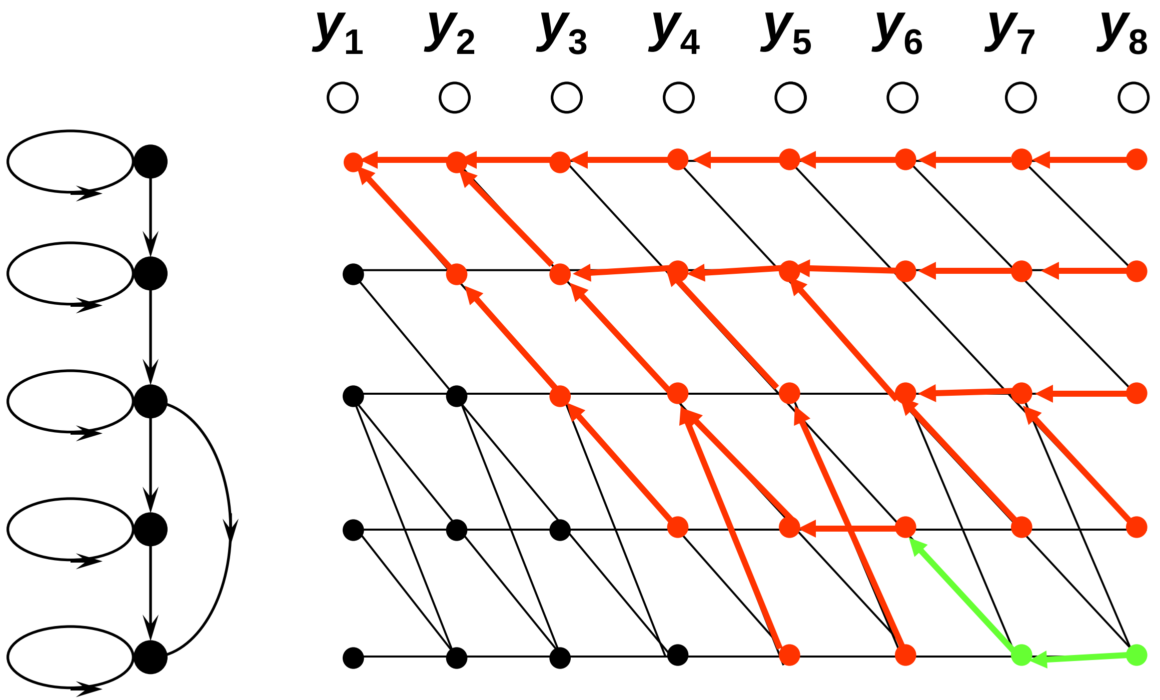
Viterbi Decoding



Trace-back



Trace-back



Trace-back