Machine Learning

Chapter 7

재귀신경망(RNN)

7. 재귀신경망

7.1 시계열 데이터의 분류

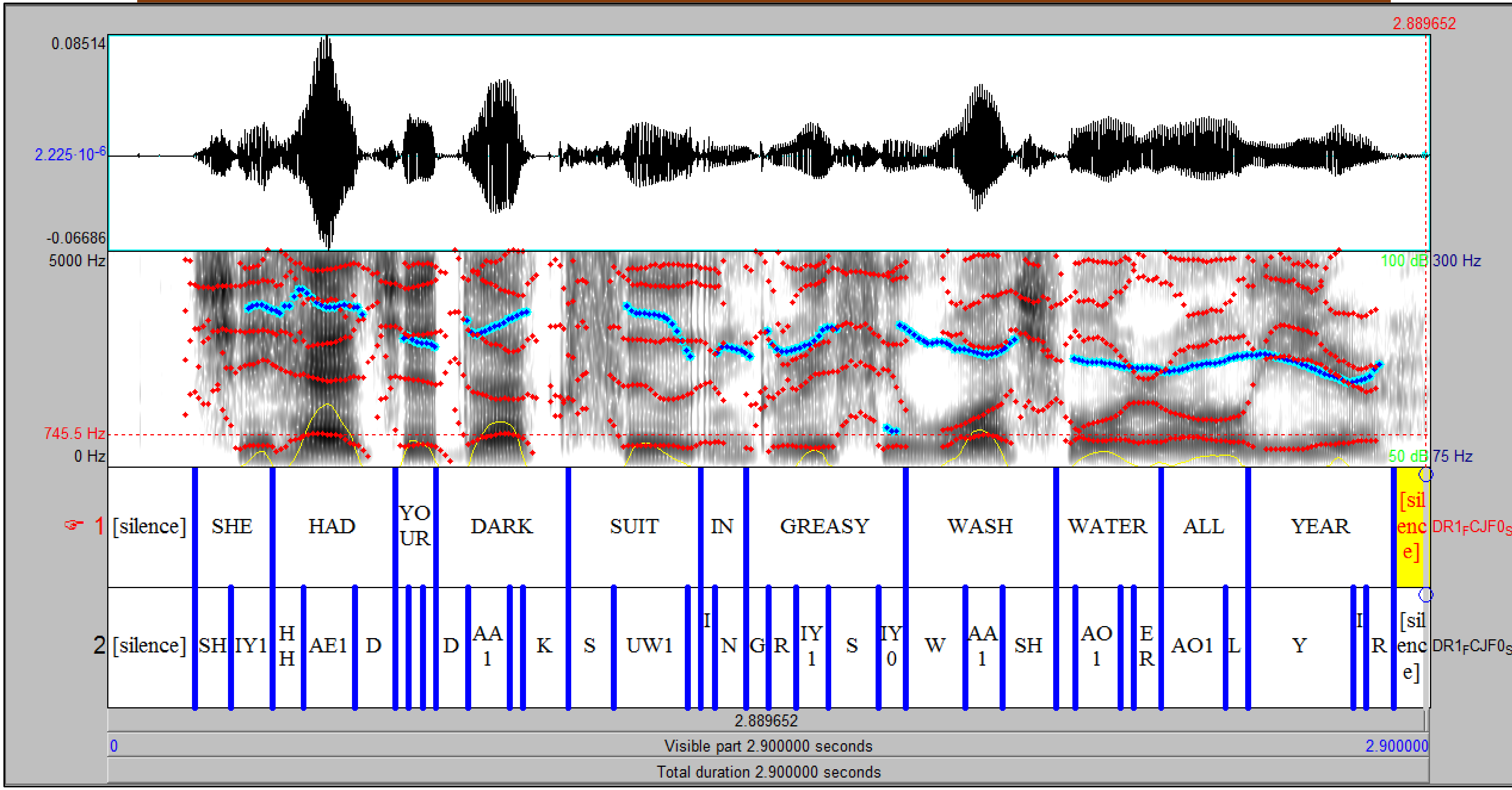
- 시계열 데이터 : 각각의 요소에 순서가 있는 모임으로 주어지는 데이터
  (음성, 동영상, 텍스트 등)

$$x^1, x^2, x^3, \cdots, x^T$$

예) <u>We can get an idea of the quality of the learned</u> feature vectors by displaying them in a 2-D map.

단어	We	can	get	⋯	the	learned	?
입력	x^1	x^2	x^3	⋯	x^{t-1}	x^t	x^{t+1}
출력		y^1	y^2	⋯	y^{t-2}	y^{t-1}	y^t

- 각 단어는 이전 단어의 시계열에 강하게 영향을 받는다.
- **재귀 신경망**은 이러한 단어의 의존관계(문맥)를 잘 학습하여 단어를 예측한다.

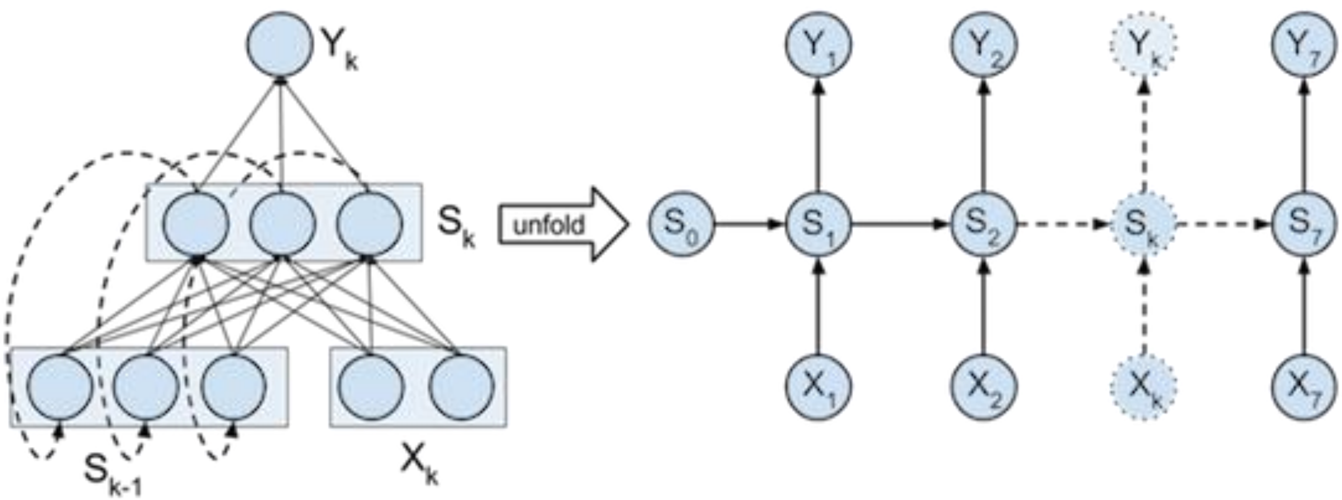


예) She had your dark suit in greasy wash water all year…

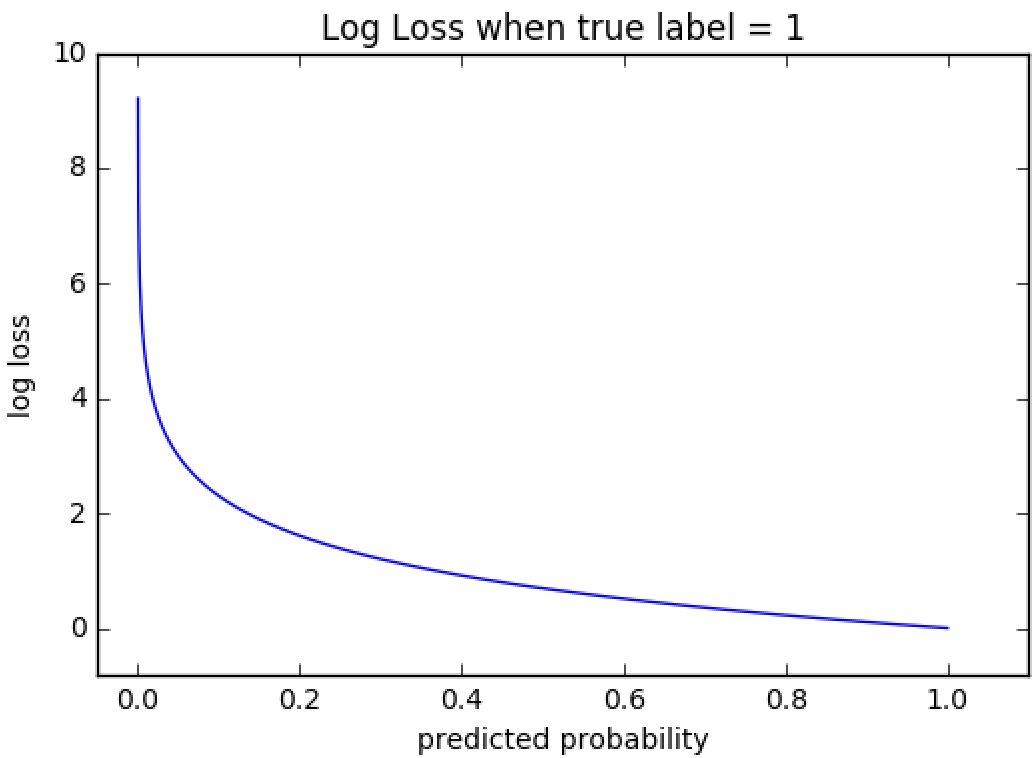
- 음소(phoneme) : 모음과 자음으로 나누어진 발화의 최소 단위
- 영어의 경우 음성신호에 대한 약 60개의 음소레이블이 존재한다.
- $x^t = s(t\Delta t)$, $(\Delta t: \sim 10ms\ interval)$ 연속열로부터 음소의 연속열 y^t을 추정할 수 있다.

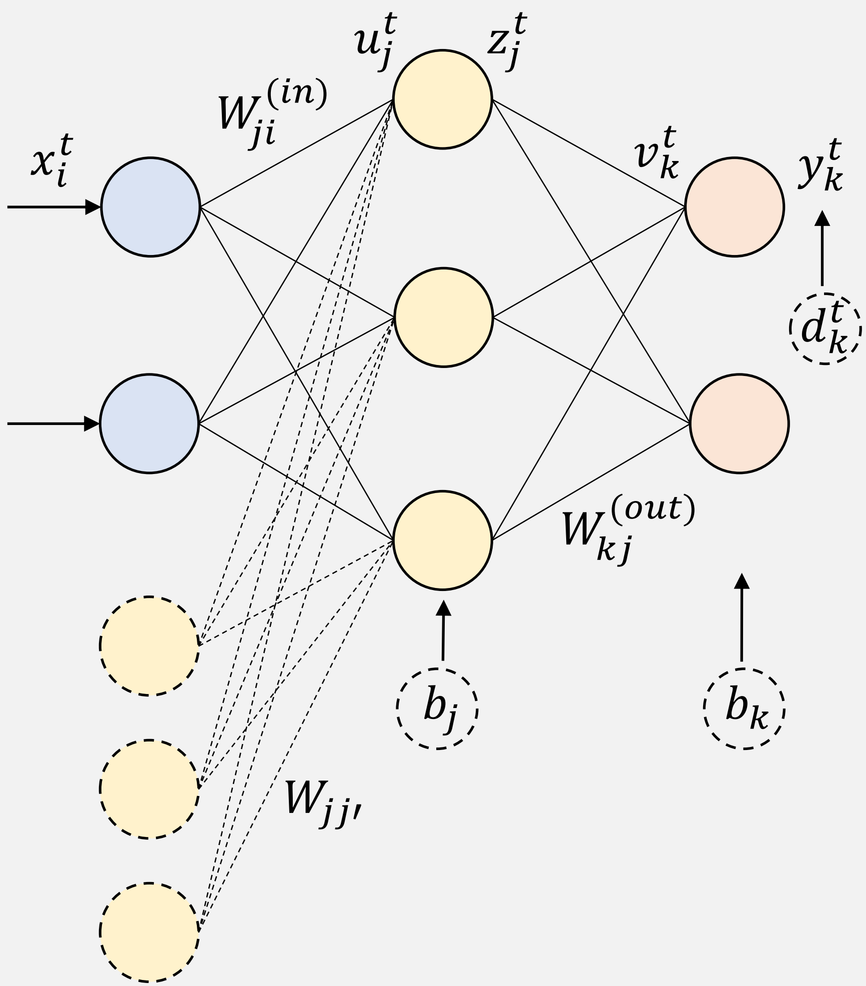
7.2 RNN의 구조

- 재귀 신경망(RNN) : 내부에 순환경로를 가진 신경망, 정보를 일시적으로 기억하고, 그에 따른 반응을 동적으로 변화
  - Elman 신경망, Jordan 신경망, 시간 지연 신경망, 에코 상태 신경망 ….
  - 시각 t 마다 하나의 입력 x^t를 입력받아 동시의 하나의 출력 y^t 를 출력.
  - 중간층은 다음 시각의 중간층으로 결합하는 귀환로를 가진다.



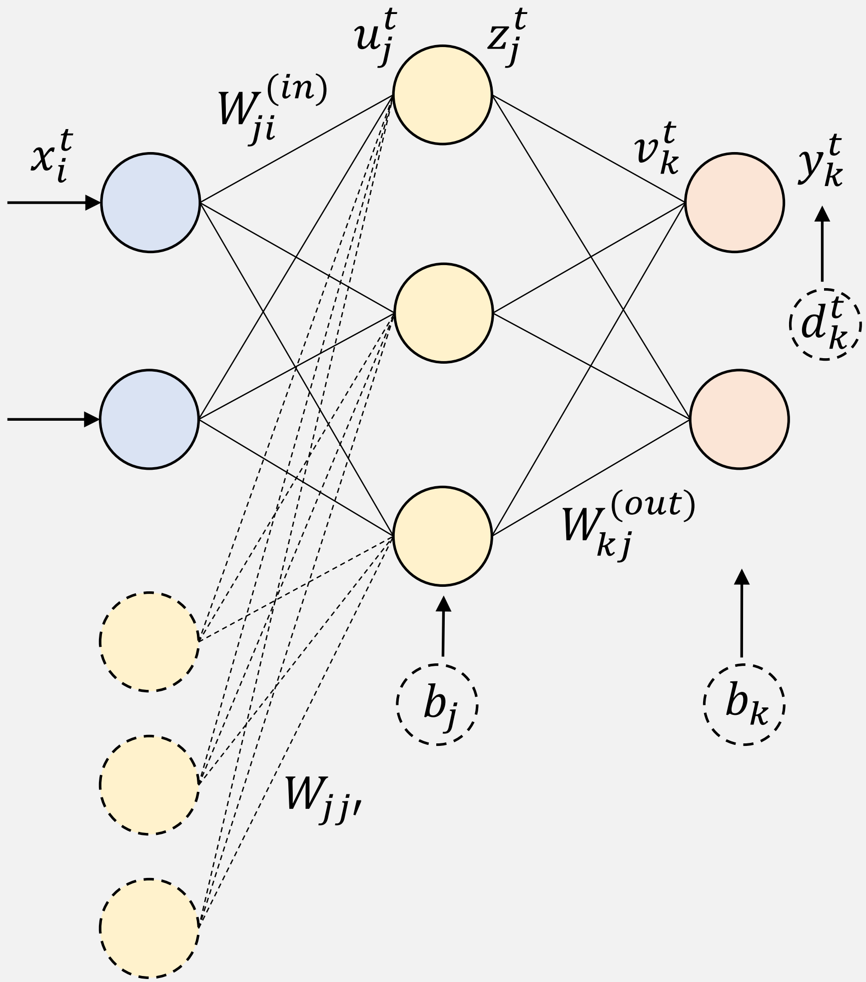
- 오차함수(CE) : E(w) = $-\sum_n \sum_t d_n^t \, log y^t$
    - d_n^t : n번째 샘플의 시각 t에서의 목표 출력
    - y^t : t시점에서 d_n^t와 비교할 실제 RNN의 출력





7.3 순전파 계산

- 입력, 중간, 출력층 유닛은 각 시각 (t=1, 2, ⋯)마다 서로 다른 상태를 가진다.
- 가중치는 학습에 의해 업데이트되기 때문에 시각 t와는 무관하다.
- 시각 t에 대한 중간층의 입력은 입력층으로부터 전해지는 값과 시각 t-1에 중간층에서 나온 출력이 피드백된 값의 합으로 이루어진다.



- 중간층 각 유닛의 입력

$$u_j^t = \sum_i w_{ji}^{(in)} x_i^t + \sum_{j'} w_{jj'} z_{j'}^{t-1}$$

- 중간층 각 유닛의 출력

$$z_j^t = f(u_j^t)$$

- 중간층의 출력

$$z^t = f(W^{(in)} x^t + W z^{t-1})$$

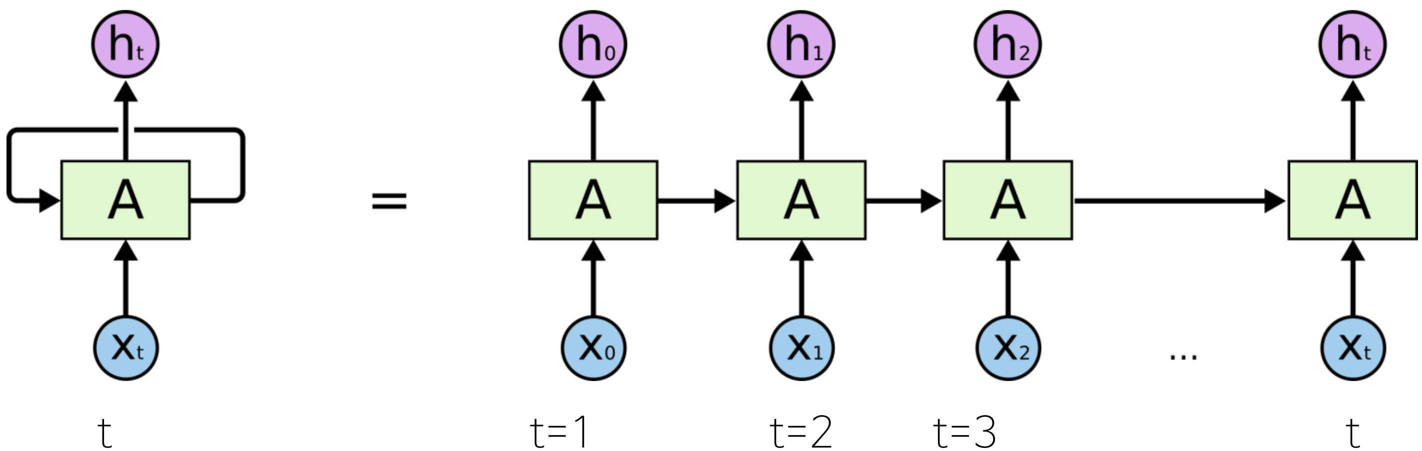
- 출력층 각 유닛의 출력

$$v_k^t = \sum_j w_{kj}^{(out)} z_j^t$$

- RNN의 출력

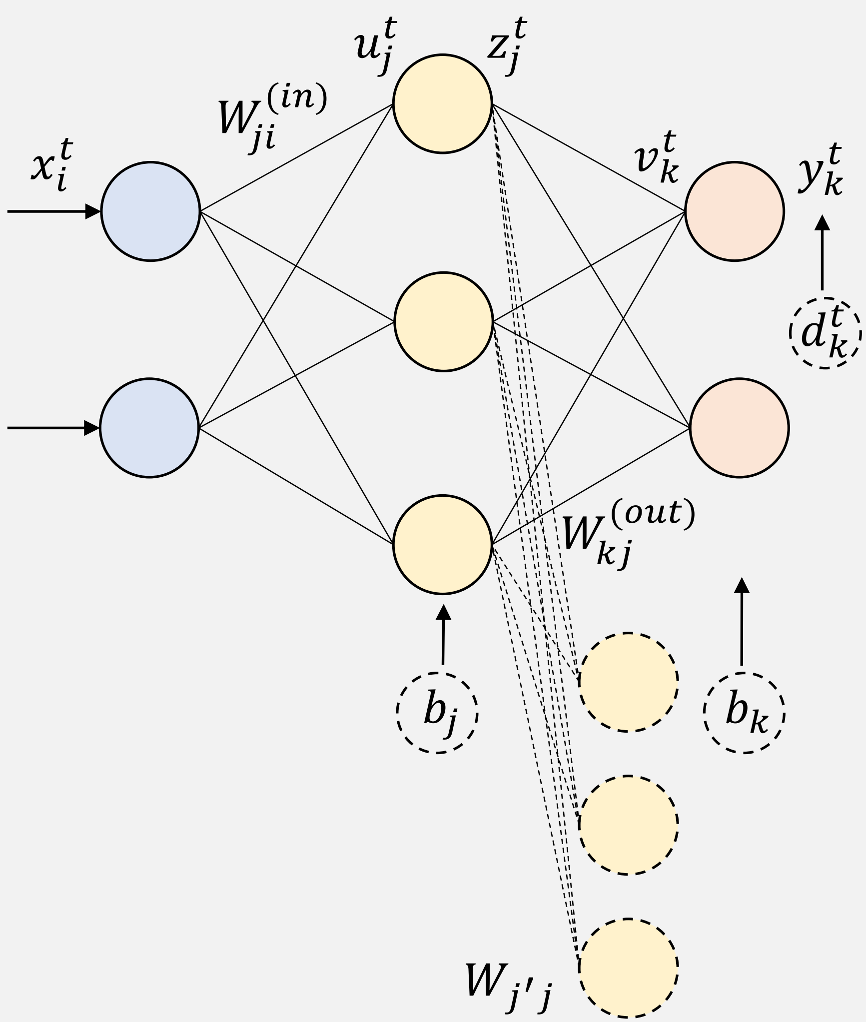
ㄱ $y^t = f^{(out)}(v^t) = f^{(out)}(W^{(out)} z^t)$

7.4 역전파 계산



- 역전파 학습 알고리즘으로 확률적 경사 하강법(SGD)을 사용.

1. RTRL(RealTime Recurrent Learning)
   - 시간 t마다 오차를 계산하여 가중치를 수정.
   - 메모리 효율이 좋음.
2. BPTT(BackPropagation Through Time)
   - RNN을 시간방향으로 전개하여 feed forward신경망과 같이 역전파 학습을 수행.
   - 메모리 효율이 좋지 않으나, 계산속도가 빠르고 좀 더 간단하다.



- 입력(j, k)에 대한 오차미분(δ)을 구한 뒤 각 가중치(w)에 대한 오차미분을 구하려 함.

- 시간 t, 출력층 k 유닛의 오차기울기

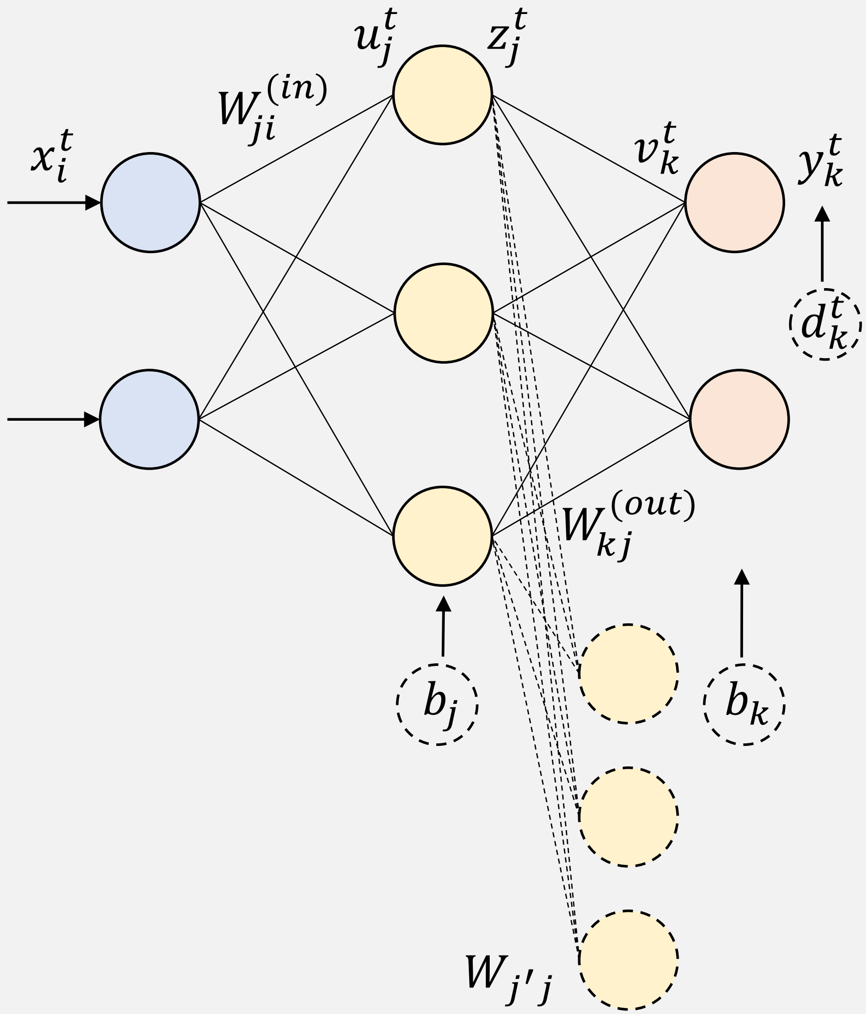
$$\delta_k^{out,t} = \frac{\partial E}{\partial v_k^t}$$

- 시간 t, 중간층 j 유닛의 오차기울기

$$\delta_j^t = \frac{\partial E}{\partial u_j^t}$$

- δ_j^t 는 t +1의 중간층 유닛과 연결되어 다시 계산됨.

$$\delta_j^t = \left(\sum_k w_{kj}^{out} \delta_k^{out,t} + \sum_{j'} w_{j'j} \delta_{j'}^{t+1}\right) f'(u_j^t)$$



- 각 층의 가중치에 대한 미분을 구할 수 있다.

$$\frac{\partial E}{\partial w}$$

- w_{ji}^{in}에 대한 오차(E)미분

$$\frac{\partial E}{\partial w_{ji}^{in}} = \sum_{t=1}^{T} \frac{\partial E}{\partial u_j^t} \frac{\partial u_j^t}{\partial w_{ji}^{in}} = \sum_{t=1}^{T} \delta_j^t x_i^t$$

- $w_{jj'}$에 대한 오차(E)미분

$$\frac{\partial E}{\partial w_{jj'}} = \sum_{t=1}^{T} \frac{\partial E}{\partial u_j^t} \frac{\partial u_j^t}{\partial w_{jj'}} = \sum_{t=1}^{T} \delta_j^t z_j^{t-1}$$

- w_{kj}^{out}에 대한 오차(E)미분

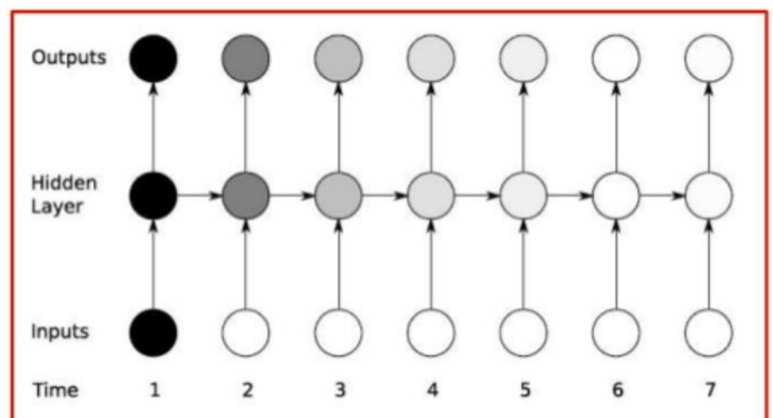
$$\frac{\partial E}{\partial w_{kj}^{out}} = \sum_{t=1}^{T} \frac{\partial E}{\partial v_k^t} \frac{\partial v_k^t}{\partial w_{kj}^{out}} = \sum_{t=1}^{T} \delta_k^{out} z_j^t$$

7.5 장.단기기억

7.5.1 RNN의 기울기 소실 문제(Vanishing Gradient)

- RNN은 연속열 데이터의 문맥을 포착하여 추청하므로 포착할 수 있는 문맥의 길이가 중요하다.
  - 이론상 과거의 모든 입력 이력이 고려되어야 한다.
  - 실제 RNN은 과거 10시각(t)정도를 출력에 반영시킬 수 있다.

- **Conventional RNN with sigmoid**
  - The sensitivity of the input values decays over time
  - The network forgets the previous input



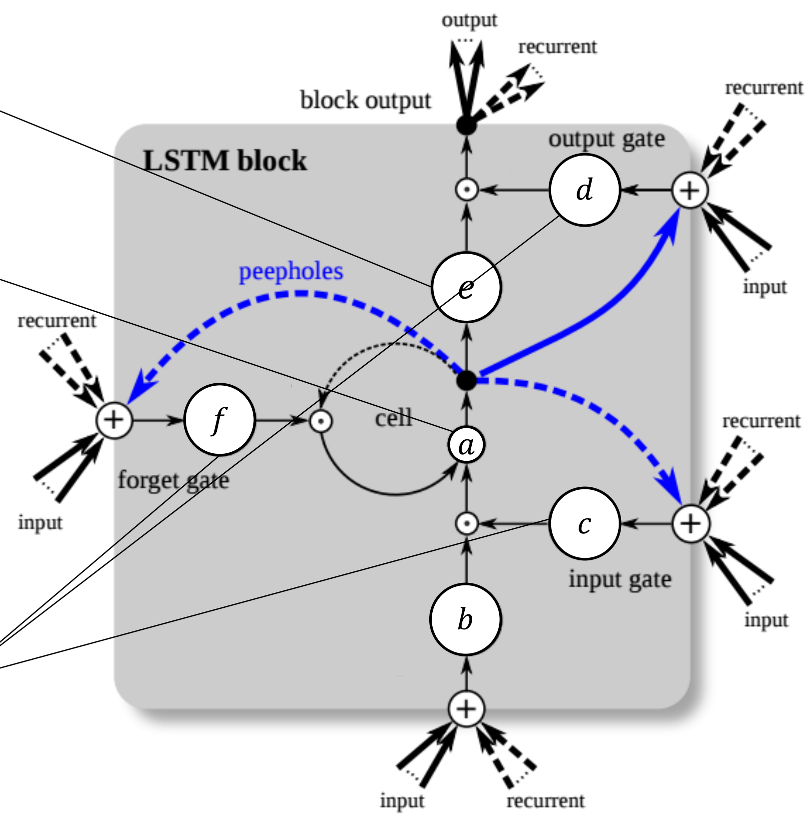
7.5.2 LSTM의 개요

- LSTM : 장단기기억(Long Short-Term Memory)
  - 기울기 소실을 해결하기 위해 장기적인 기억을 실현하기 위한 방법 중 하나
  - 중간층의 각 유닛이 메모리유닛이라 불리는 요소로 구성됨.
  - 과거와 현재의 정보를 선택적으로 기억하기 위해 메모리셀과 게이트유닛을 사용.

e : 메모리셀의 출력에 활성화 함수를 적용 [0,1]

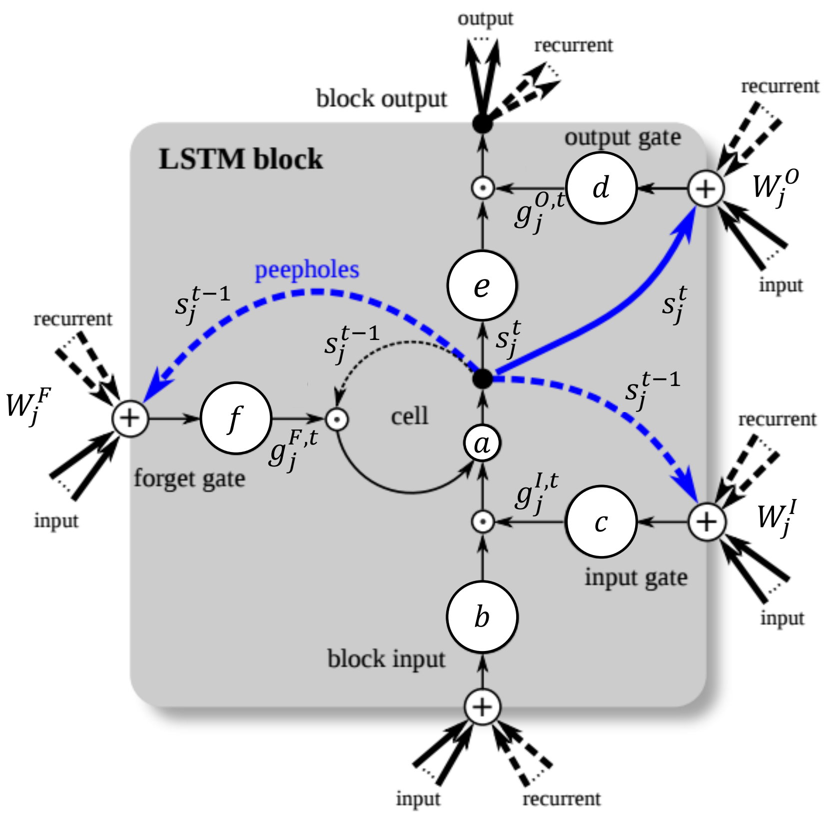
a : 상태를 기억하는 메모리셀

c, d, f $gate$: 입력과 재귀값에 활성화 함수를 적용 [0,1]



e, b : tanh
c, d, f : $sigmoid$

7.5.3 순전파 계산



$$input + recurrent =$$

$$u_j^t = \sum_i w_{ji}^{(in)} x_i^t + \sum_{j'} w_{jj'} \, z_{j'}^{t-1}$$

- j번째 유닛의 메모리셀 a는 상태 s_j^t를 유지하며, 상태를 한 시각 후로 전달한다.

$$s_j^t = g_j^{F,t} s_j^{t-1} + g_j^{I,t} f(u_j^t)$$

  - s_j^{t-1} : 한 시각 전으로부터 전달된 상태
  - $f(u_j^t)$: 이 메모리유닛 j가 받아들인 입력

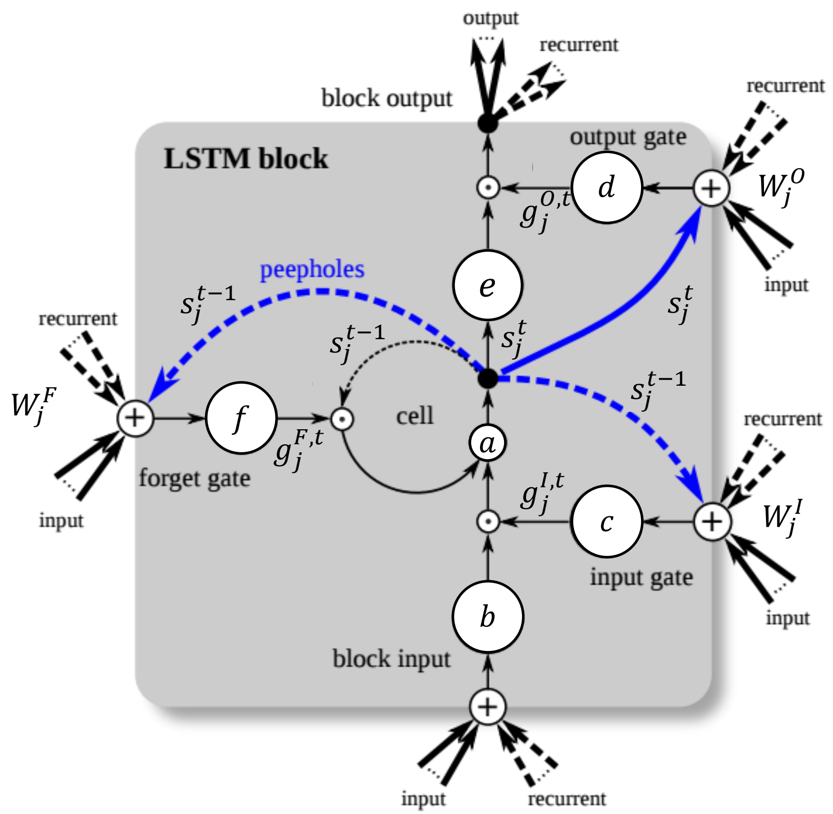
- 게이트 (g_j^t) : 각 게이트는 **sigmoid**에 입력을 받아 1(기억), 0(망각) 사이의 상태를 전달한다.

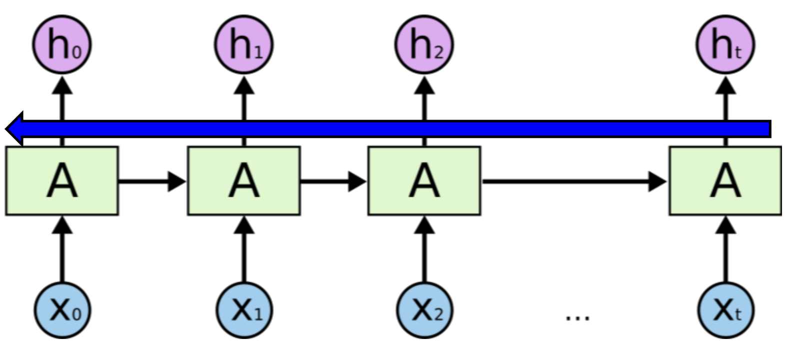
$$g_j^t = f\left(\sum_i w_{ji}^{in} x_i^t + \sum_{j'} w_{jj'} z_j^{t-1} + w_j s_j^{t-1}\right)$$

  - 출력 게이트는 peephole결합 값이 $w_j^O s_j^t$을 사용함에 주의

- 메모리 유닛의 출력(s_j^t를 사용함에 주의)

$$z_j^t = g_j^{O,t} f(s_j^t)$$





7.5.4 역전파 계산

- 메모리셀, 입력유닛, 각 게이트의 입력에 대해 역전파 계산을 수행하여 기울기를 계산.

- t시각의 기울기

$$\delta_j^t = \sum_k \delta_k^{t+1} \frac{\partial u_k^{t+1}}{\partial u_j^t}$$

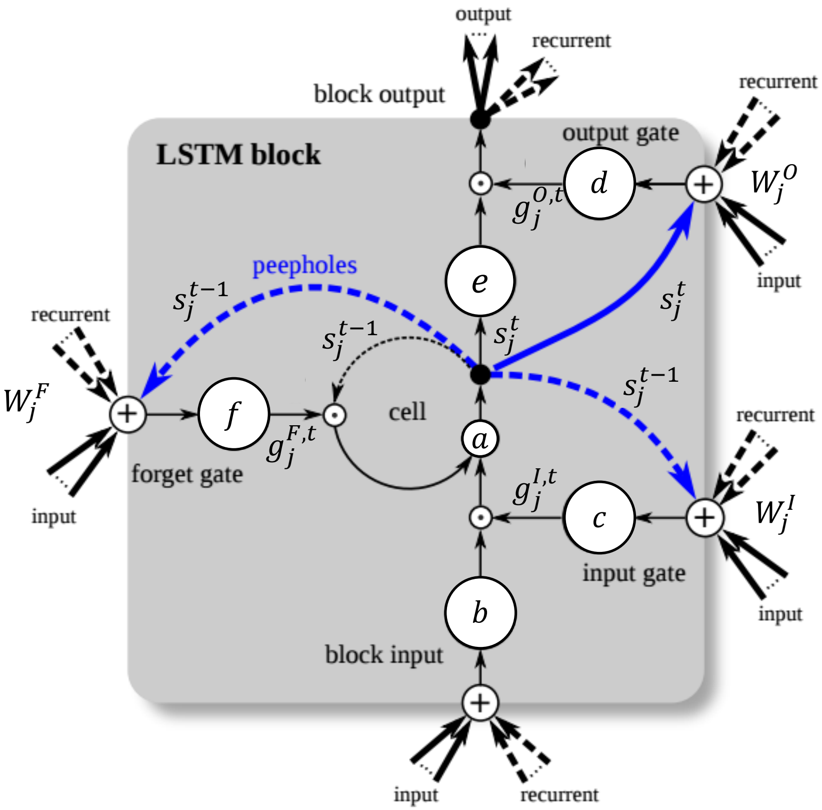
- 출력층에 들어가는 입력의 기울기

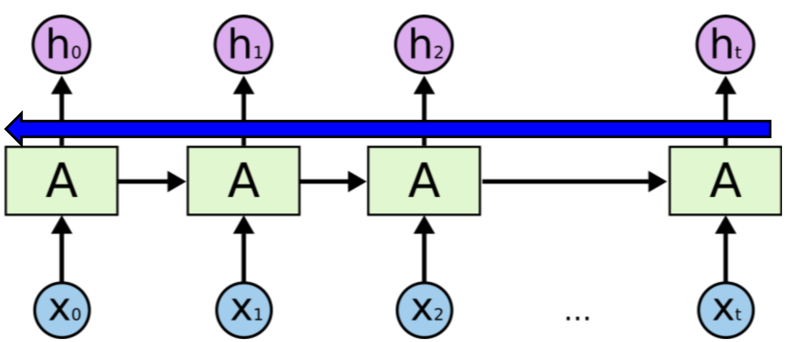
$$\frac{\partial v_k^t}{\partial u_j^{O,t}} = w_{kj}^{out} f'\left(u_{kj}^{out}\right) f\left(s_j^t\right)$$

- 출력게이트의 기울기

$$\delta_j^{O,t} = f'\left(u_j^{O,t}\right) f\left(s_j^t\right) \epsilon_j^t$$

$$\epsilon_j^t = \sum_k w_{kj}^{out} \delta_k^{out,t} + \sum_{j'} w_{j'j}\, \delta_{j'}^{t+1}$$





- 메모리셀의 기울기

$$\delta_j^{cell,t} = \tilde{\delta}_j^t + g_j^{F,t+1}\delta_j^{cell,t+1} + w_j^F\delta_j^{F,t+1} + w_j^O\delta_j^{O,t}$$

- 입력유닛의 기울기

$$\delta_j^t = g_j^{I,y} f'(u_j^t)\delta_j^{cell,t}$$

- 망각게이트의 기울기

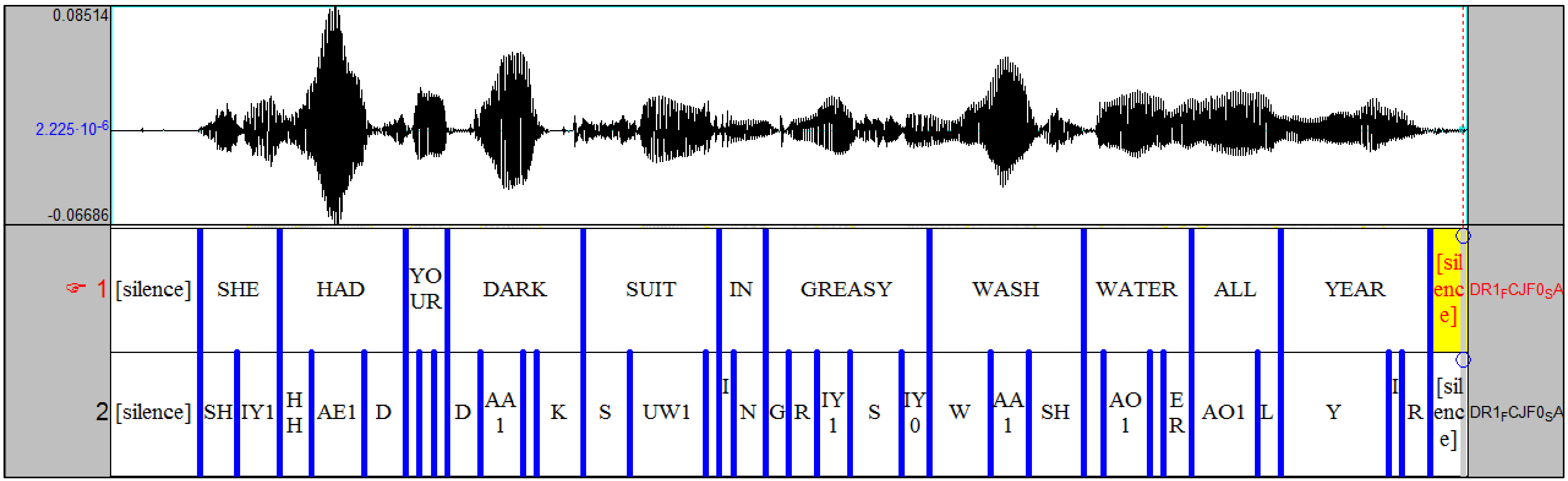
$$\delta^{F,t}j = f'(u_j^{F,t})s_j^{t-1}\delta_j^{cell,t}$$

- 입력게이트의 기울기

$$\delta_j^{I,t} = f'(u_j^{I,t})f(u_j^t)\delta_j^{cell,t}$$

7.6 입력과 출력의 연속열 길이가 다른 경우

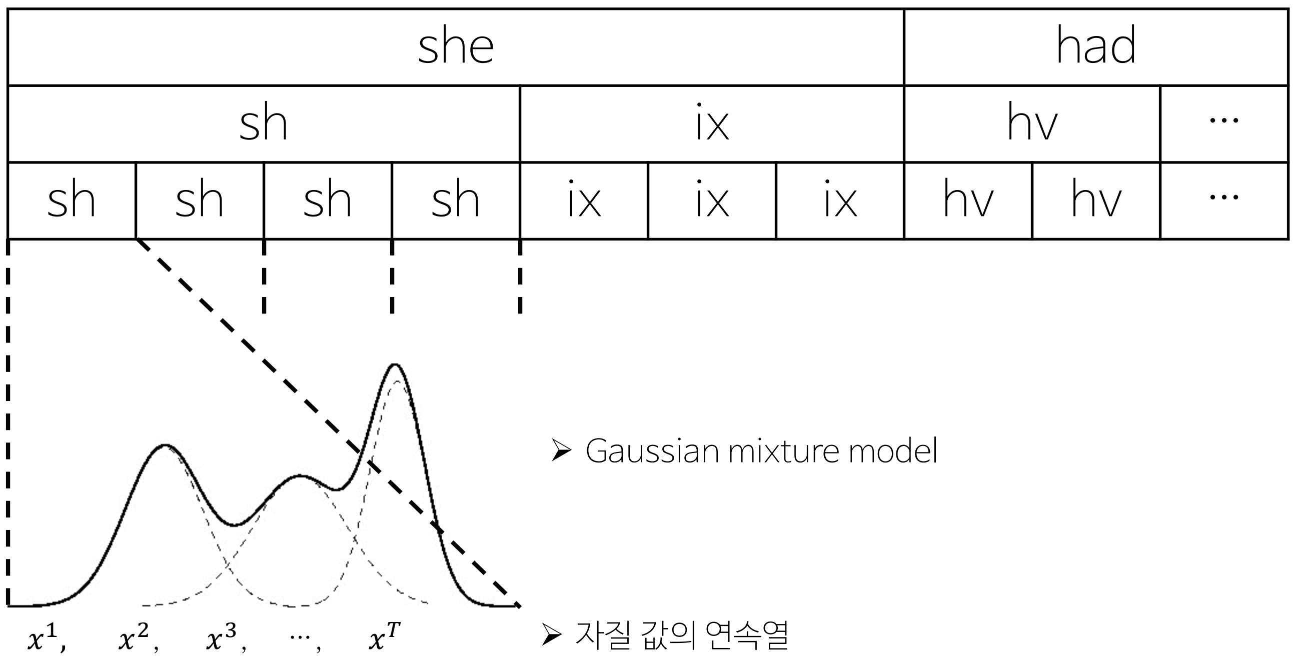
7.6.1 은닉 마르코프 모델(Hiddern Markov Model)



예) She had your dark suit in greasy wash water all year…

- 목표는 sh, ix, hv, …와 같은 연속열을 얻는것임.
- 실제 RNN의 출력은 sh, sh, sh, sh, ix, ix, ix, hv, hv, 같은 불필요한 음소의 반복을 얻음

- HMM은 이러한 문제를 다루는 가장 일반적인 방법.
  - 내부 상태를 숨겨진 변수로 가지고, 이 변수가 시각과 함께 확률적으로 변화함.
  - 현재의 내부 상태에 기초한 확률적 관측을 생성함.



- 각 자질마다 음소값을 추정하는 확률을 가진다(가우시안 혼합 모델 등에 의해).
- 음소를 생성할 확률이 높은 HMM을 특정하여 대표음소를 추정할 수 있음.

7.6.2 커넥셔니스트 시계열 분류(Connectionist Temporal Classification)

- 시계열 입출력의 길이가 다른 문제에서 HMM을 사용하지 않고 신경망만을 사용해서 해결하는 방법
  - 입력시계열과 다른 길이의 출력 연속열을 다루기 위해 RNN의 출력에 대한 해석을 바꾼다.
  - 출력층 활성화 함수로 soft-max를 사용.

Ex) 입력 연속열의 길이가 T=8일 때 정답레이블 길이가 4인 l=‘cbab’를 추정하는 문제

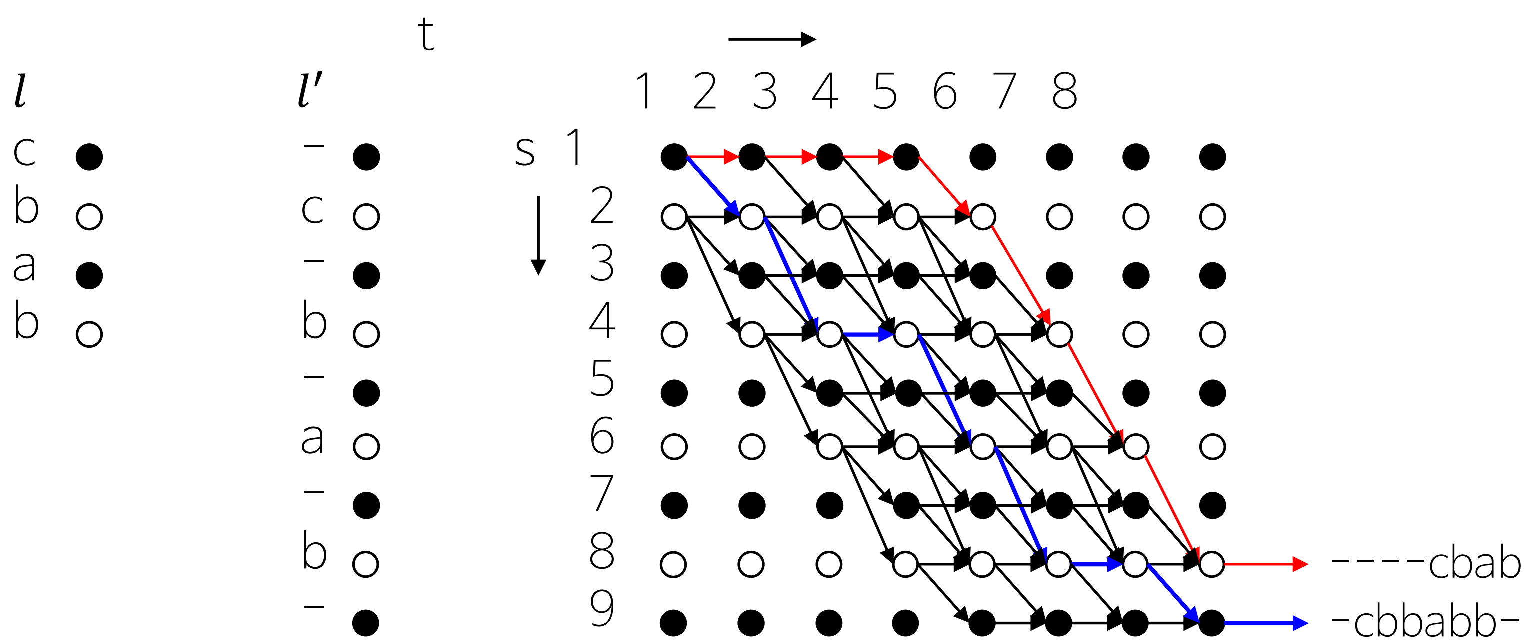
- 정답레이블에 공백레이블 ‘-’을 추가하여 연속열에 잉여정보를 포함시킴.

<table>
<tr><td>l(정답)</td><td colspan="2">c</td><td colspan="2">b</td><td colspan="3">a</td><td>b</td></tr>
<tr><td>π (parsed)</td><td>c</td><td>-</td><td>b</td><td>-</td><td>-</td><td>a</td><td>a</td><td>b</td></tr>
</table>

.
.
.

- 잉여정보를 포함하는 연속열 $\boldsymbol{\pi}$와 원래 연속열 l의 관계

$$l = \beta(\boldsymbol{\pi}) \rightarrow \beta^{-1}(l) = \{\pi \mid \beta(\pi) = l\}$$



- 그림의 화살표를 따라가면 l = ‘cbab’에 대해 길이 T=8인 $\boldsymbol{\pi}$가 모두 조합된다.
- 시각 t에 대한 출력 y_k^t은 시각 t의 정답 레이블이 k일 확률로 해석됨(soft-max).

- 입력 시계열 $X = (x^1, \dots, x^T)$에 대해 하나의 파스 $\pi = [\pi_q, \dots, \pi_T]$가 정답일 확률

$$p(\pi \mid X) = \prod_{t=1}^{T} y_{\pi_t}^{t}$$

- 하나의 정답레이블 l을 실현하는 모든 파스 $\pi \in \beta^{-1}(l)$에 대해 $p(\pi \mid X)$를 더하여 l의 확률을 계산

$$p(l \mid X) = \sum_{\pi \in \beta^{-1}(l)} p(\pi \mid X)$$