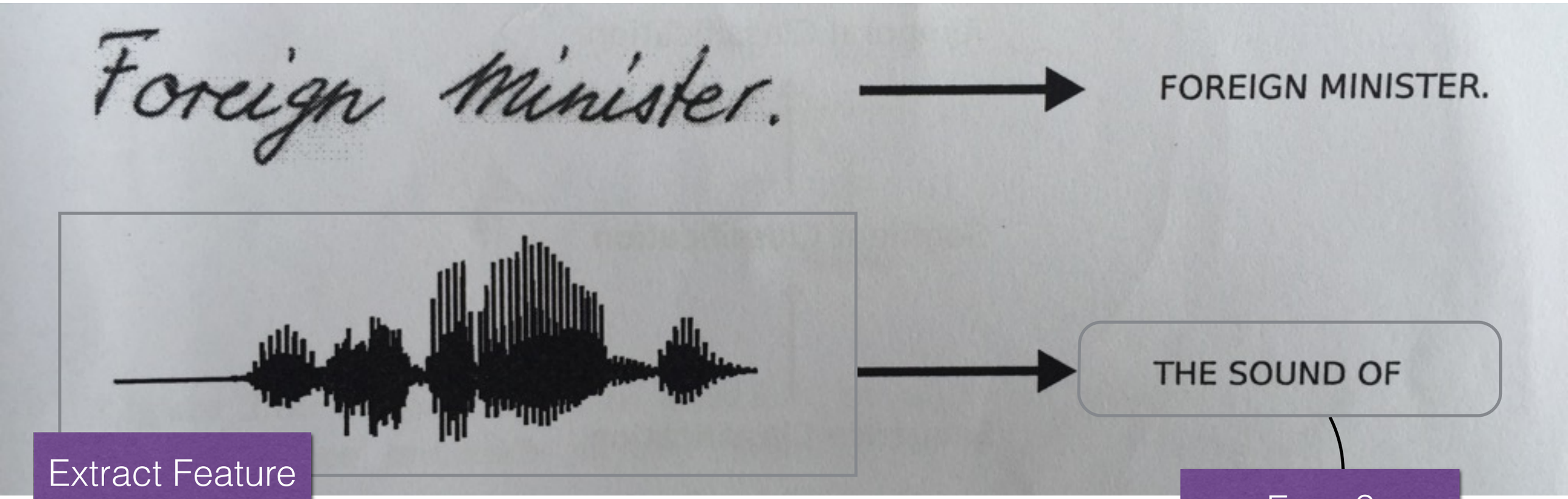
Sequence Learning from character based acoustic model to End-to-End ASR system

替代役男 王俊豪
chuchuhao831@gmail.com

Sequence Learning



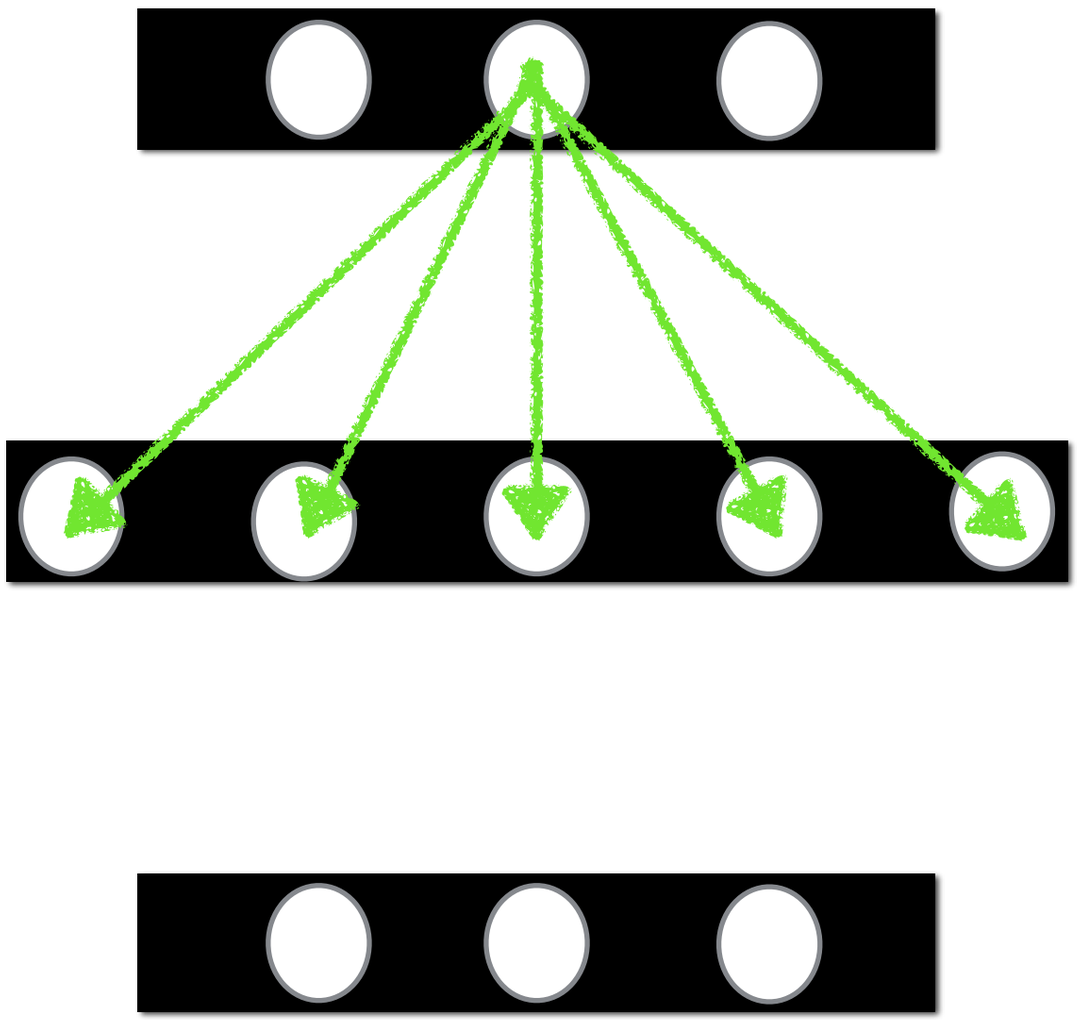
a stream of input data
= a sequence of acoustic feature
= a list of MFCC (Here we use)

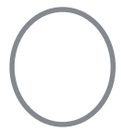
figure from A. Graves, Supervised sequence labelling, ch2 p9

Outline

1. **F**eedforward **N**eural **N**etwork
2. **R**ecurrent **N**eural **N**etwork
3. **B**idirectional **R**ecurrent **N**eural **N**etwork
4. **L**ong **S**hort-**T**erm **M**emory
5. Connectionist Temporal Classification
6. Build a End-to-End System
7. Experiment on Low-resource Language

Feedforward Neural Network



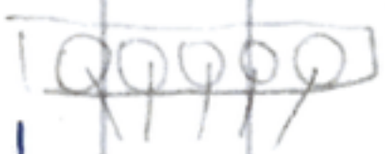


Active Function

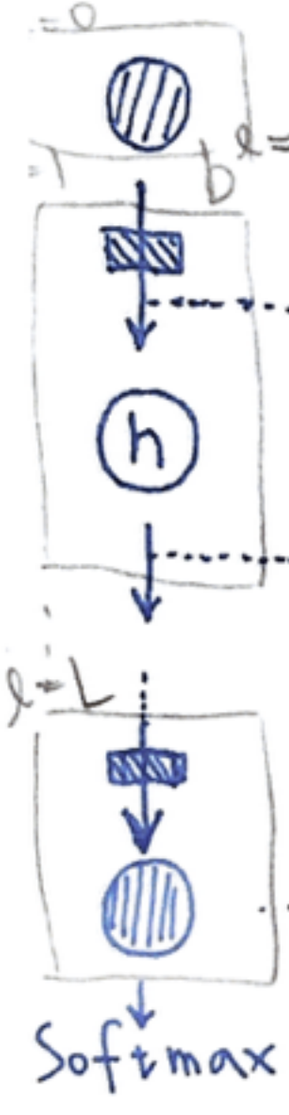
Weighted

FNN

ℓ = layer



[Forward Pass] ↓



$$a_h^\ell = \sum_{h'=1}^{H^{\ell-1}} w_{h'h}^\ell \cdot b_{h'}^{\ell-1}$$

$$b_h^\ell = \sigma_h(a_h)$$

h: hidden unit

H_h^ℓ: # of hidden unit's

$$a_k^L = \sum_{h=1} w_{hk}^\ell \cdot b_h^{\ell-1}$$

k: output units

$$y_k = p(C_k|x) = \frac{e^{a_k}}{\sum_{k'=1}^{k} e^{a_{k'}}}$$

Multiclass

[Backward Pass]

$$\frac{\partial O}{\partial b_h} = \sum_{k=1}^{k} \frac{\partial O}{\partial a_k} \frac{\partial a_k}{\partial b_h}$$

δ_h = $\sum w_{h'h}$

$\sigma'(a_h)$

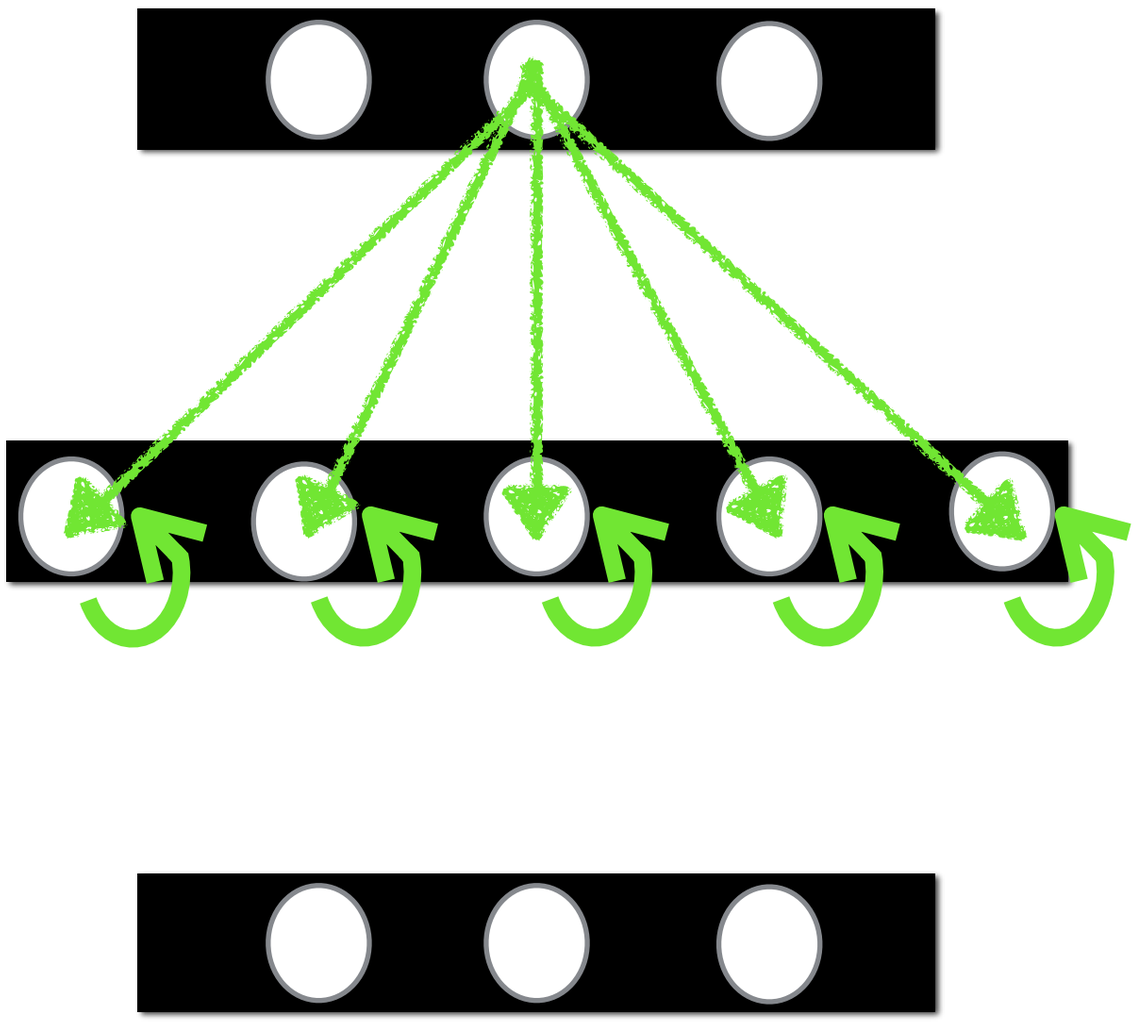
$$\delta_h \overset{\text{def}}{=} \frac{\partial O}{\partial a_h} = \frac{\partial O}{\partial b_h} \frac{\partial b_h}{\partial a_h} = \sigma'(a_h) \sum_{k=1}^{K} \delta_k w_{hk}$$

b_i

$$\frac{\partial O}{\partial w_{ij}} = \frac{\partial O}{\partial a_i} \frac{\partial a_i}{\partial w_{ij}} = \delta_i b_i$$

$$O = -\sum_{(x,z)\in S} \sum_{k=1}^{K} z_k \ln y_k$$

Recurrent Neural Network

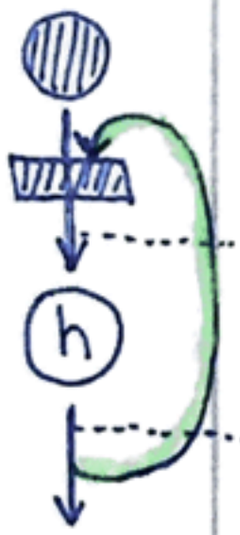


Active Function

Weighted

RNN

[Forward Pass]



$$a_h^{t,l} = \sum_{h'=1}^{H^{l-1}} w_{h'h} \cdot b_{h'}^{t} + \sum_{h''=1}^{H^{l}} w_{h''h} \cdot b_{h''}^{t-1}$$

$$b_h^t = \sigma_h(a_h^t)$$

$$a_k^t = \sum_{h=1}^{H^{l+1}} w_{hk} \cdot b_h^t$$

[Backward Pass]

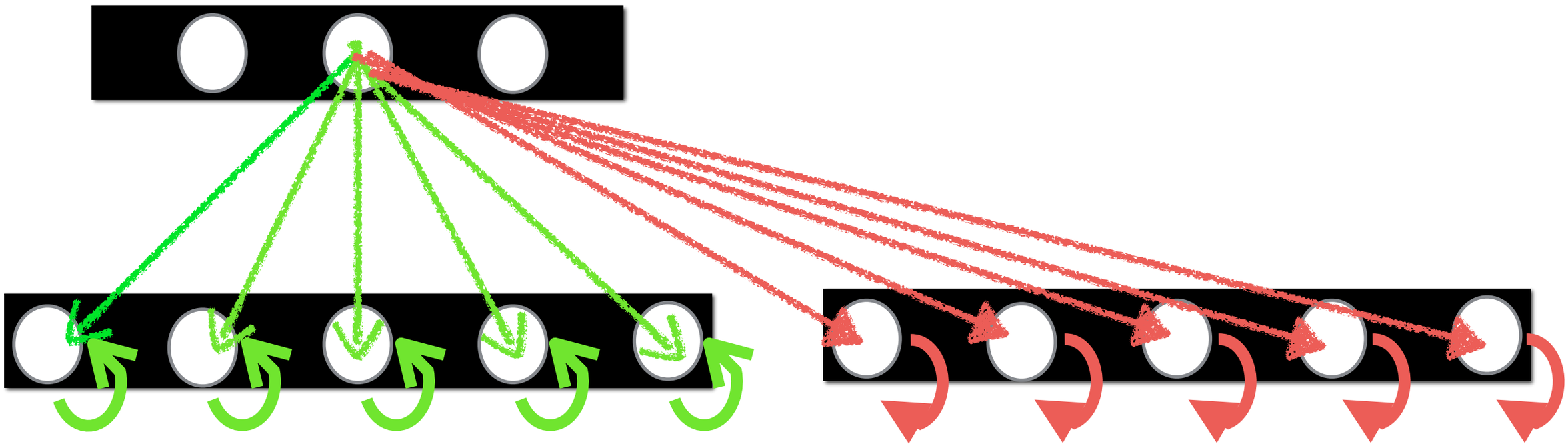
$$\frac{\partial O}{\partial b_h^t} = \sum_{k=1}^{K} \frac{\partial O}{\partial a_k^t} \frac{\partial a_k^t}{\partial b_h^t} + \sum_{h'=1}^{H} \frac{\partial O}{\partial a_{h'}^{t+1}} \frac{\partial a_{h'}^{t+1}}{\partial b_h^t}$$

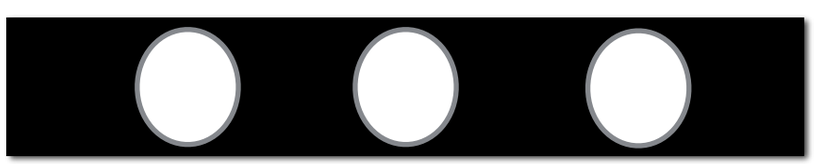
$$= \sum_{k=1}^{K} \delta_k^t w_{hk} + \sum_{h'=1}^{H} \delta_{h'}^{t+1} w_{hh'}$$

$$\delta_h^t = \frac{\partial O}{\partial a_h^t} = \frac{\partial O}{\partial b_h^t} \frac{\partial b_h^t}{\partial a_h^t} \quad \frac{\partial b_h^t}{\partial a_h^t} = \sigma'(a_h^t)$$

$$\frac{\partial O}{\partial w_{ij}} = \sum_{t=1}^{T} \frac{\partial O}{\partial a_j^t} \frac{\partial a_j^t}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_j^t b_i^t$$

Bidirectional RNN





Active Function

Weighted

for $t = 1$ to T **do**
  Do forward pass for the forward hidden layer, storing activations at each timestep
for $t = T$ to 1 **do**
  Do forward pass for the backward hidden layer, storing activations at each timestep
for $t = 1$ to T **do**
  Do forward pass for the output layer, using the stored activations from both hidden layers

Algorithm 3.1: BRNN Forward Pass

Similarly, the backward pass proceeds as for a standard RNN trained with BPTT, except that all the output layer δ terms are calculated first, then fed back to the two hidden layers in opposite directions:

for $t = T$ to 1 **do**
  Do BPTT backward pass for the output layer only, storing δ terms at each timestep
for $t = T$ to 1 **do**
  Do BPTT backward pass for the forward hidden layer, using the stored δ terms from the output layer
for $t = 1$ to T **do**
  Do BPTT backward pass for the backward hidden layer, using the stored δ terms from the output layer

Algorithm 3.2: BRNN Backward Pass

figure from A. Graves, Supervised sequence labelling, ch3 p24

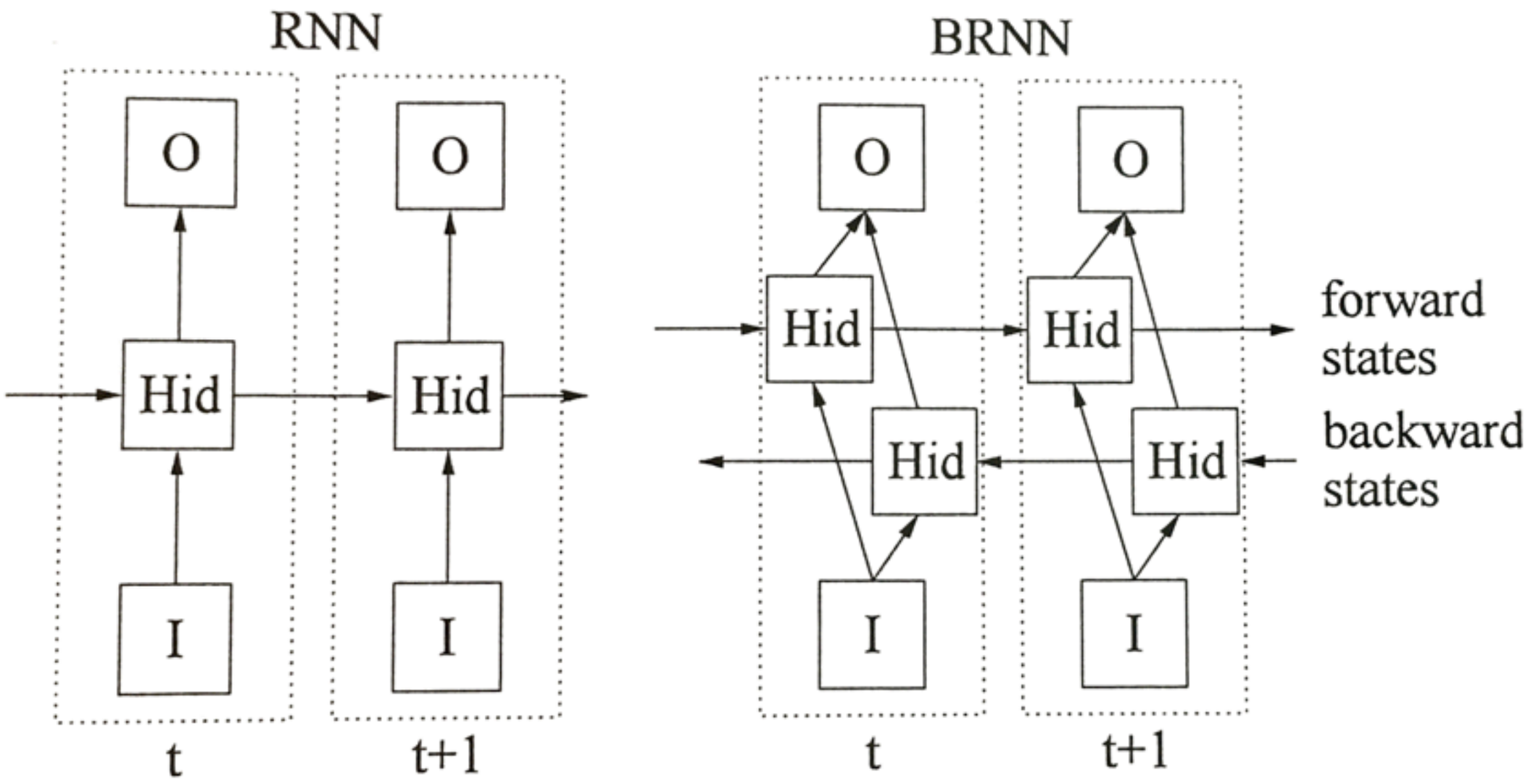


Figure 3.4: **Standard and bidirectional RNNs**

figure from A. Graves, Supervised sequence labelling, ch3 p23

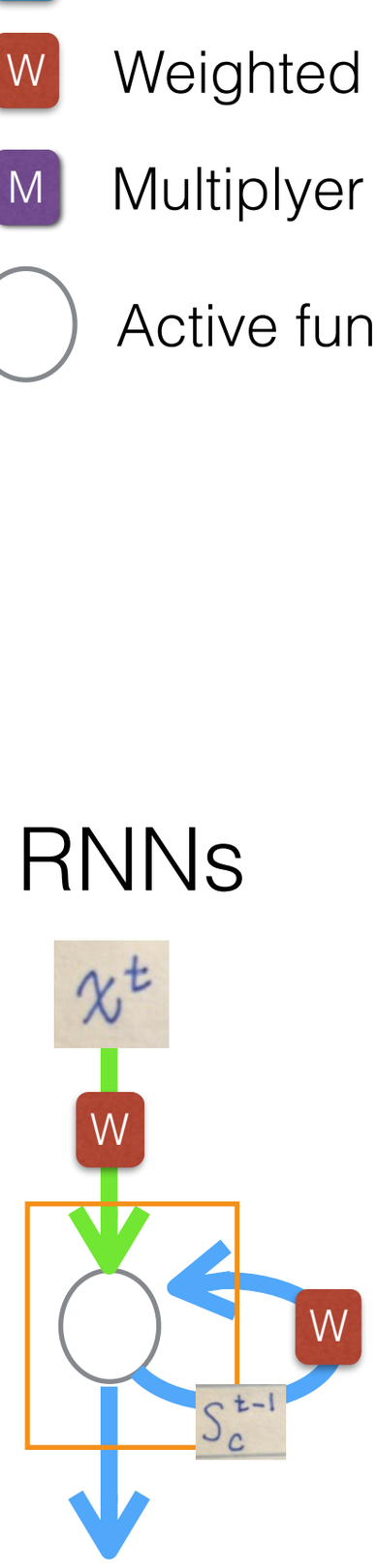
G Gate

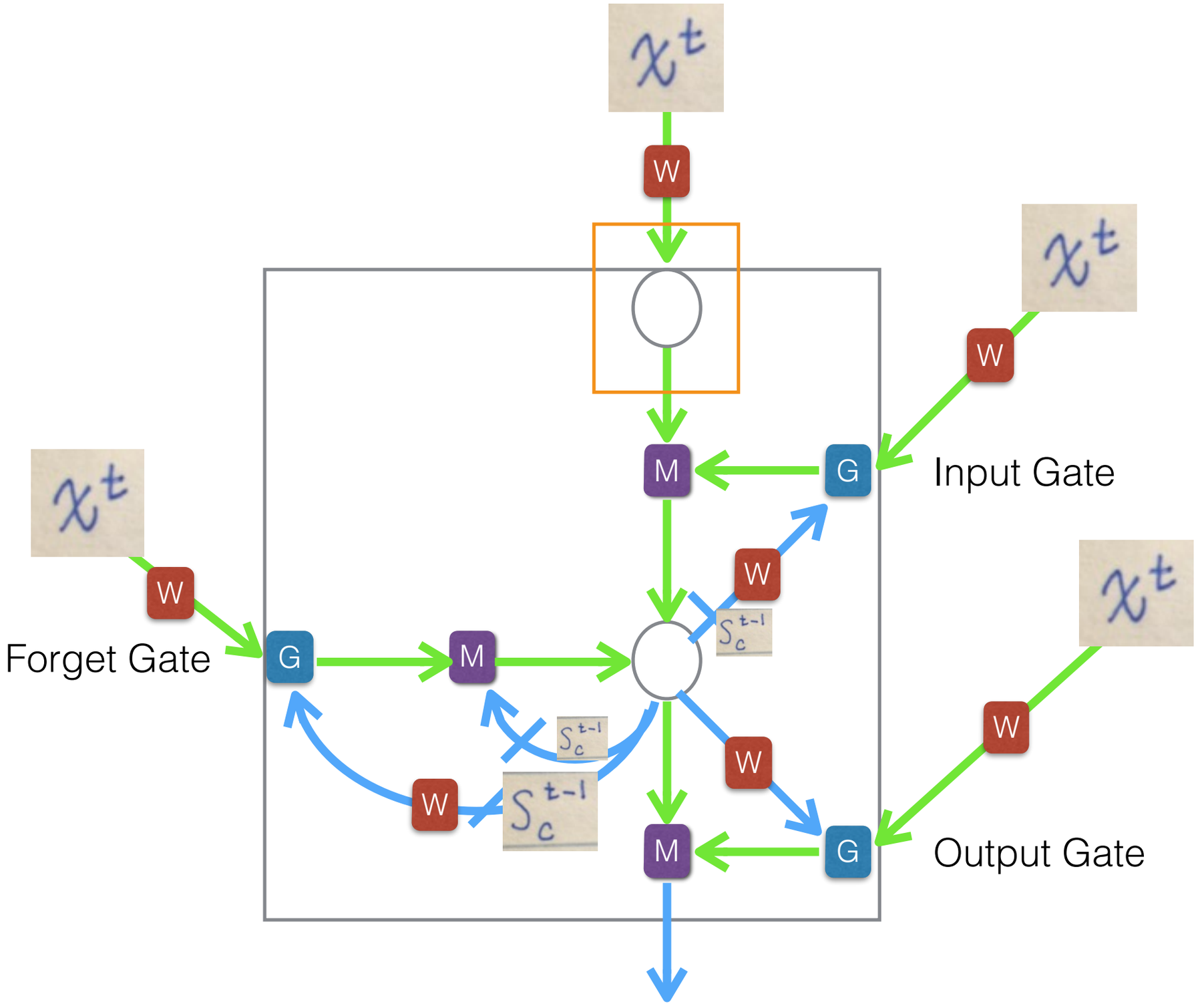
W Weighted

M Multiplyer

Active func

LSTM

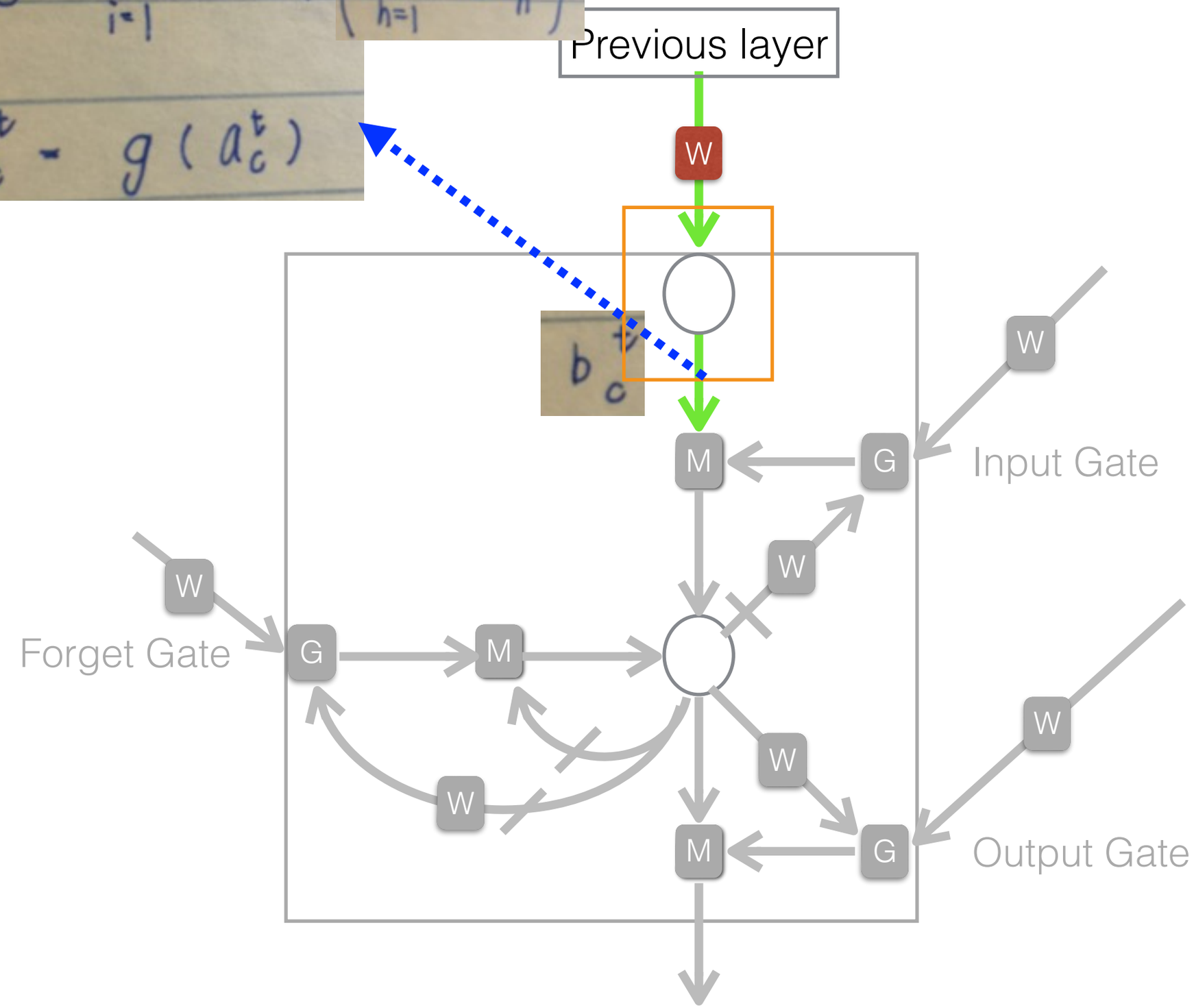




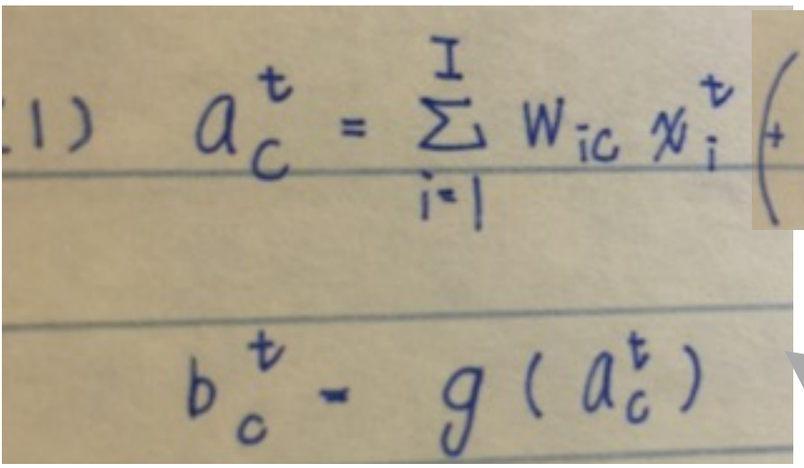
LSTM Forward Pass, Previous Layer Input

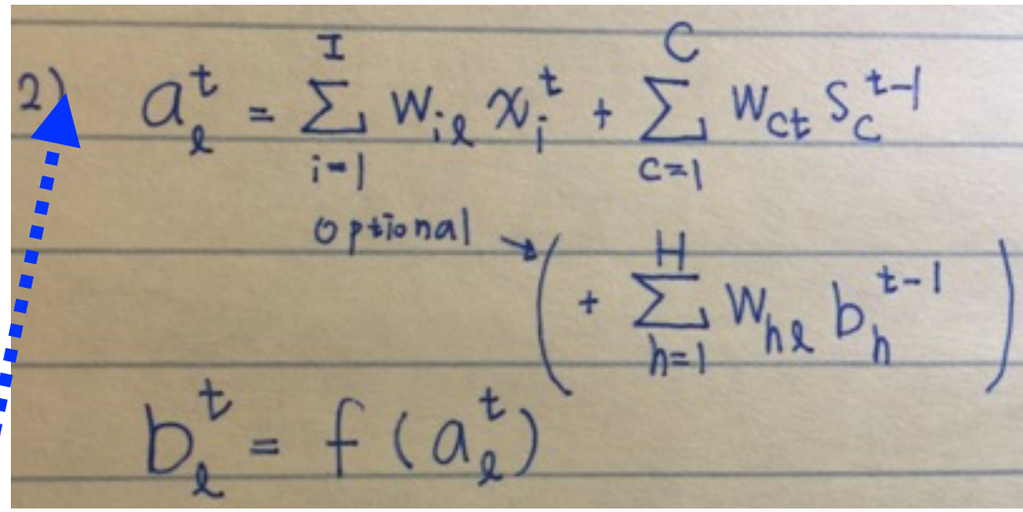
$$(1) \quad a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$$

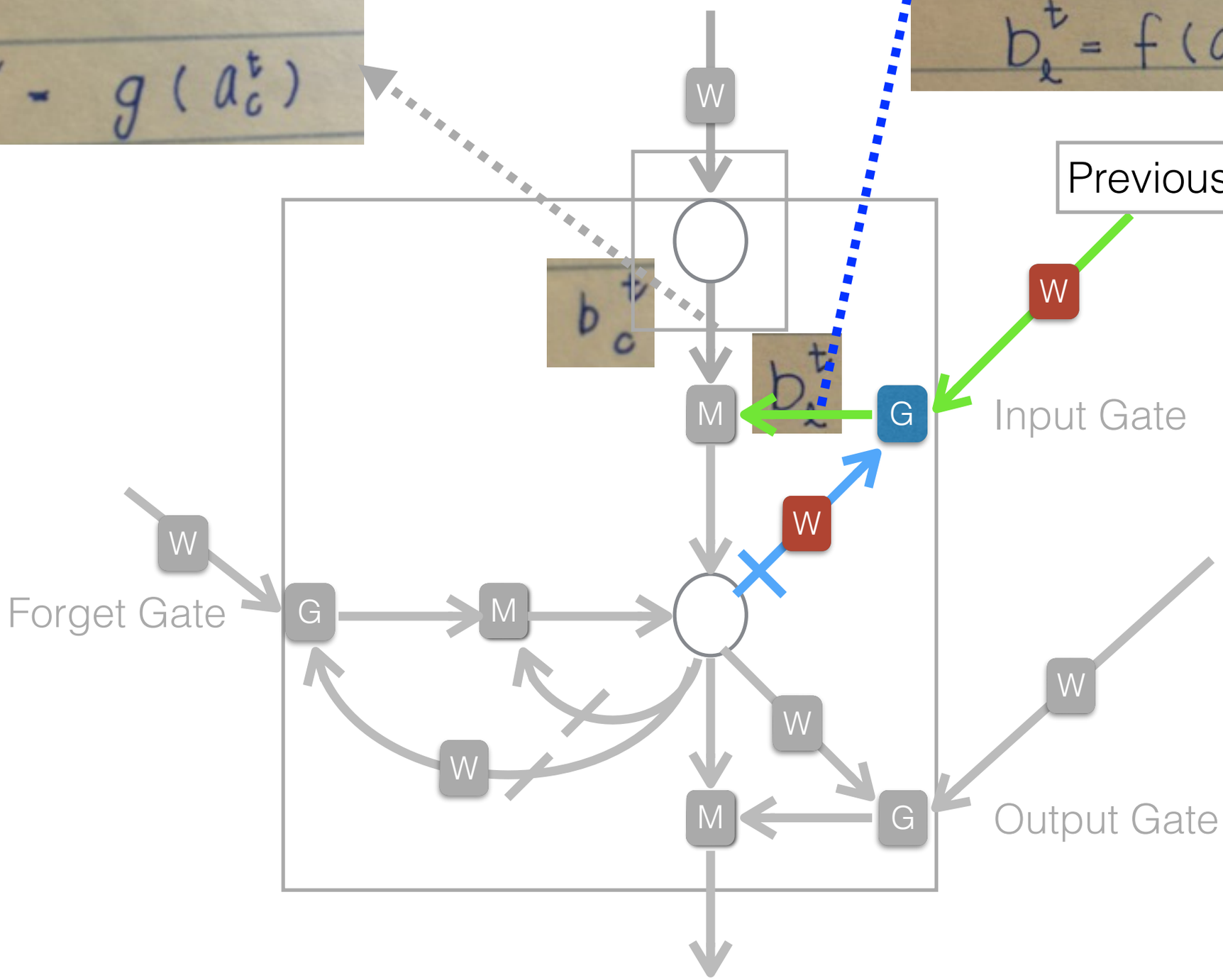
$$b_c^t = g(a_c^t)$$



LSTM Fordward Pass, Input Gate







LSTM Fordward Pass, Foget Gate

(1) $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t \left(+ \sum_{h=1}^{H} w_{hc} b_h^{t-1} \right)$

$b_c^t = g(a_c^t)$

(2) $a_\ell^t = \sum_{i=1}^{I} w_{i\ell} x_i^t + \sum_{c=1}^{C} w_{ct} s_c^{t-1}$

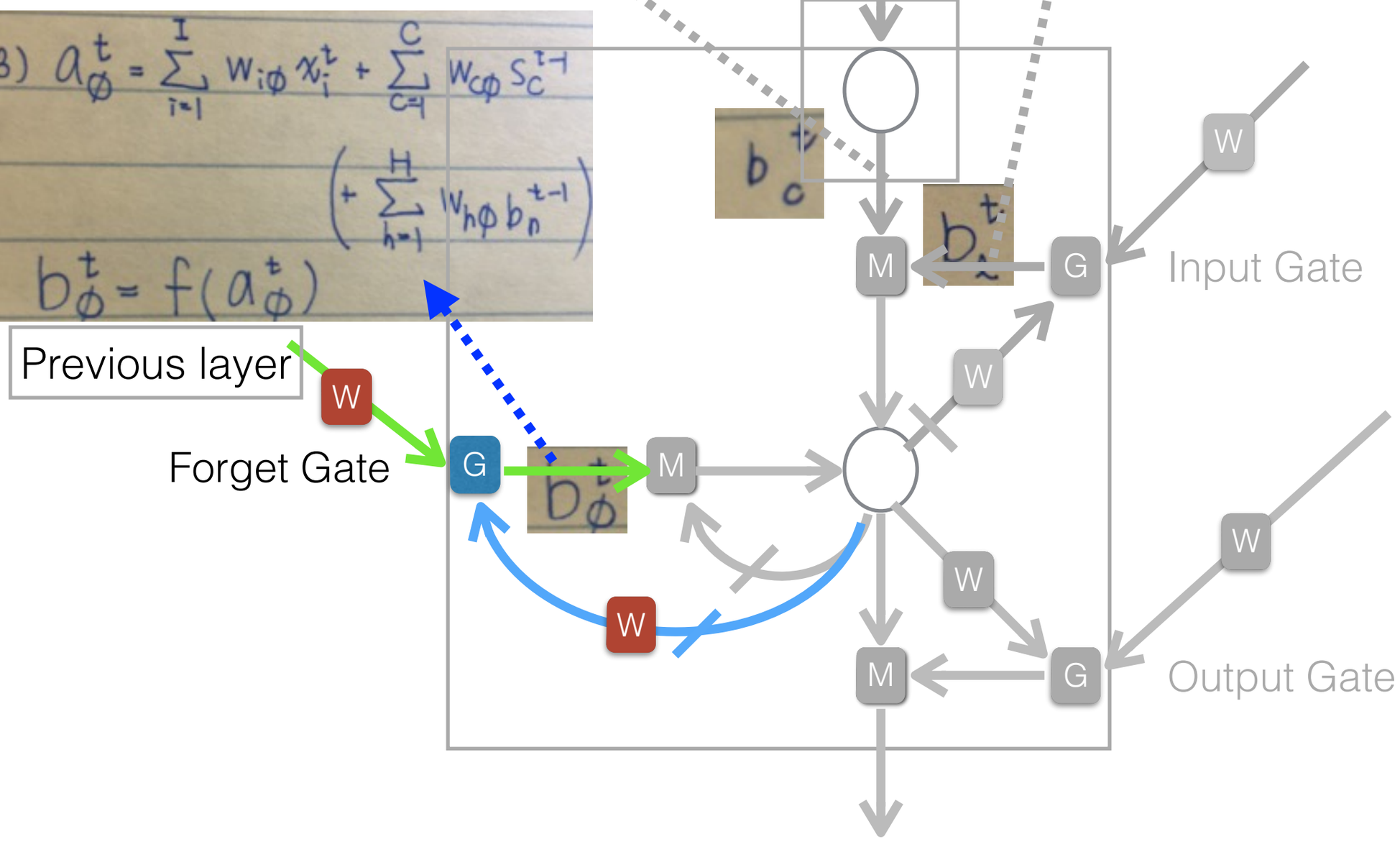
optional $\left(+ \sum_{h=1}^{H} w_{h\ell} b_h^{t-1} \right)$

$b_\ell^t = f(a_\ell^t)$

(3) $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$

$\left(+ \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} \right)$

$b_\phi^t = f(a_\phi^t)$



LSTM Forward Pass, Cell

(1) $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \left(+ \sum_{h=1}^{H} w_{hc} b_h^{t-1} \right)$

$b_c^t = g(a_c^t)$

(2) $a_\ell^t = \sum_{i=1}^{I} w_{i\ell} x_i^t + \sum_{c=1}^{C} w_{c\ell} s_c^{t-1}$

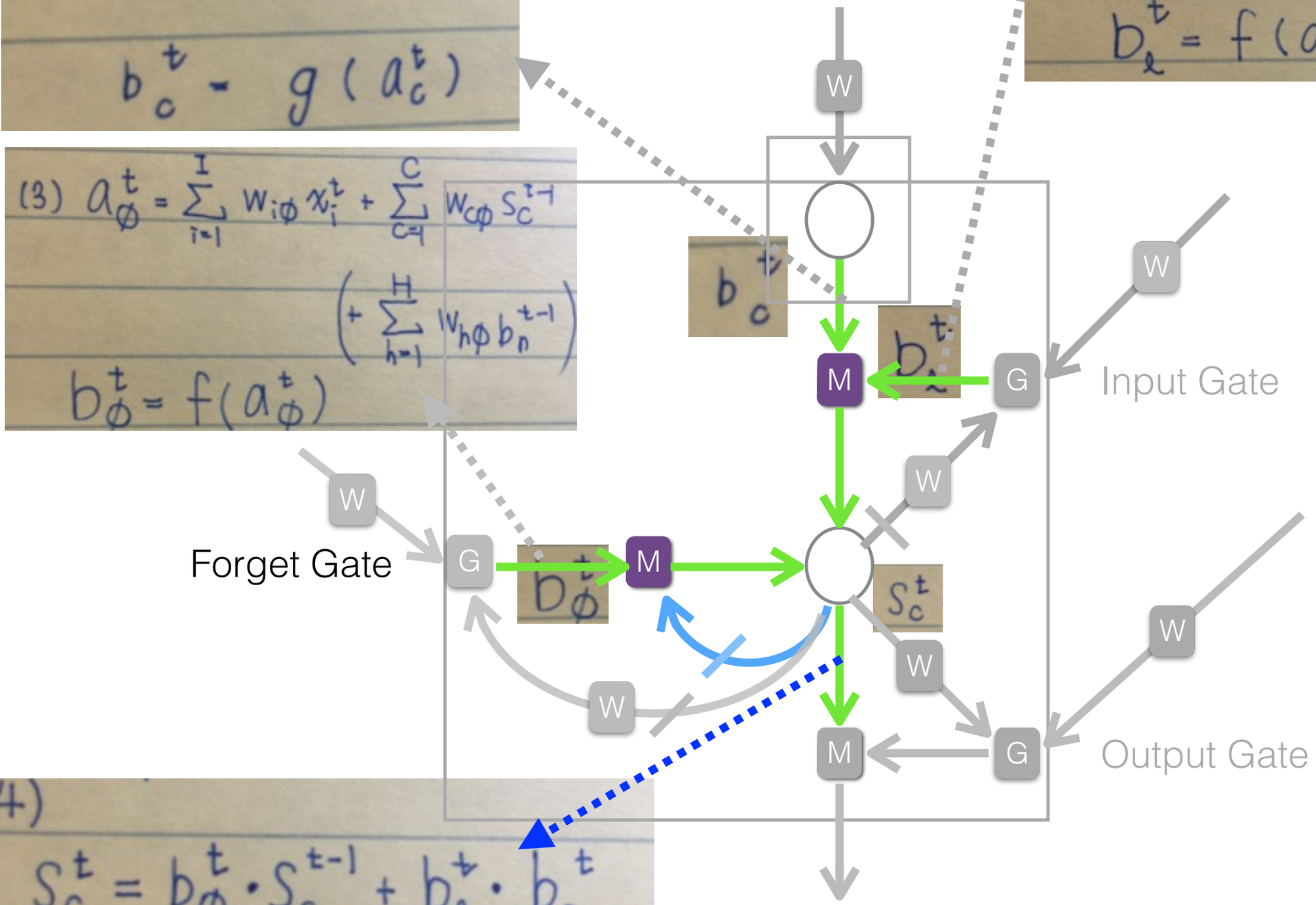
optional → $\left(+ \sum_{h=1}^{H} w_{h\ell} b_h^{t-1} \right)$

$b_\ell^t = f(a_\ell^t)$

(3) $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$

$\left(+ \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} \right)$

$b_\phi^t = f(a_\phi^t)$



(4) $s_c^t = \underbrace{b_\phi^t \cdot s_c^{t-1}}_{\text{(3) forget Gate}} + \underbrace{b_\ell^t \cdot b_c^t}_{\text{(2) Input Gate}}$

LSTM Forward Pass, Output Gate

(1) $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \left(+ \sum_{h=1}^{H} w_{hc} b_h^{t-1} \right)$

$b_c^t = g(a_c^t)$

2) $a_\ell^t = \sum_{i=1}^{I} w_{i\ell} x_i^t + \sum_{c=1}^{C} w_{c\ell} s_c^{t-1}$

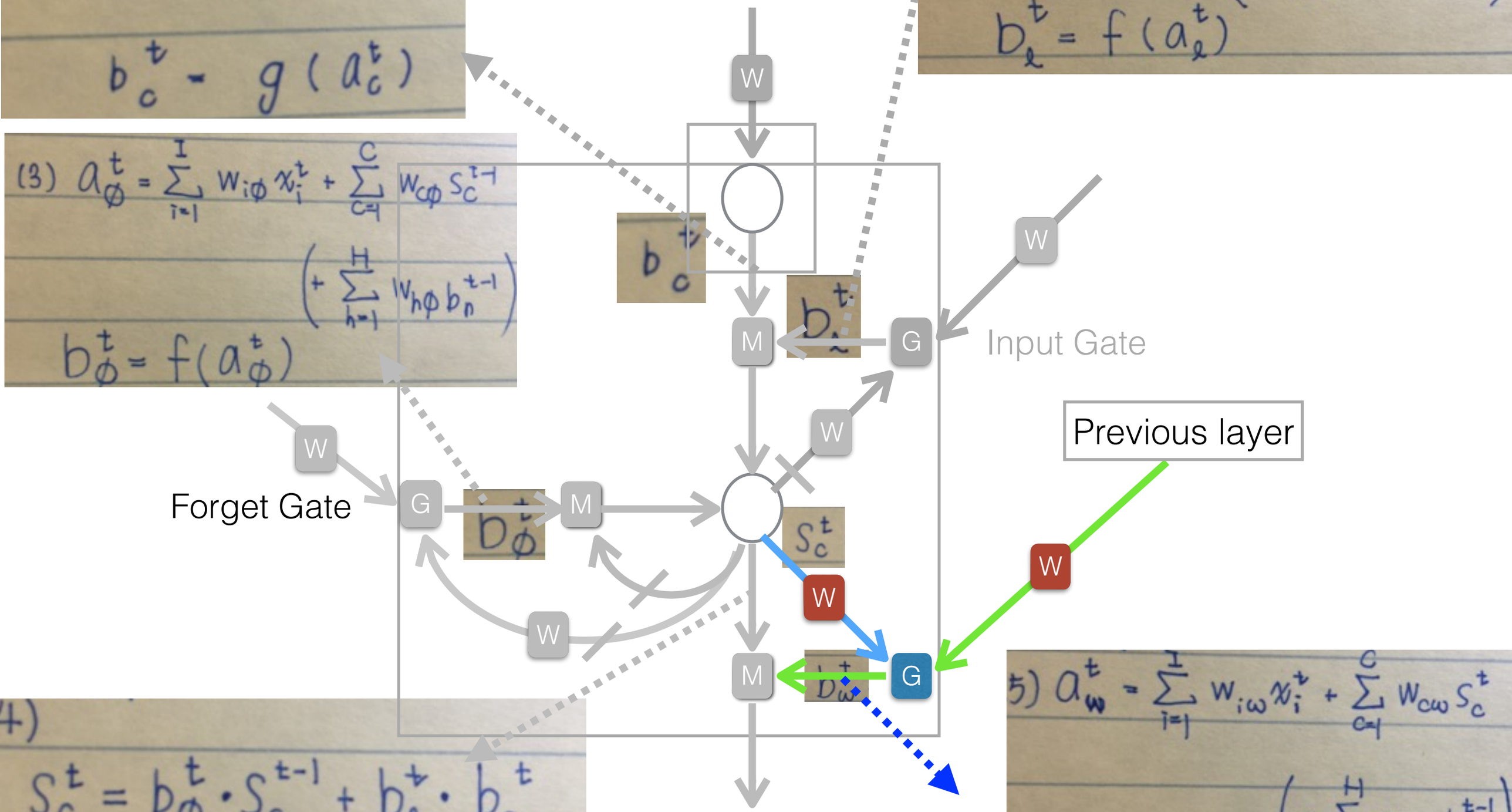
optional → $\left(+ \sum_{h=1}^{H} w_{h\ell} b_h^{t-1} \right)$

$b_\ell^t = f(a_\ell^t)$

(3) $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$

$\left(+ \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} \right)$

$b_\phi^t = f(a_\phi^t)$



4) $s_c^t = \underbrace{b_\phi^t \cdot s_c^{t-1}}_{\text{(3) forget Gate}} + \underbrace{b_\ell^t \cdot b_c^t}_{\text{(2) Input Gate}}$

5) $a_\omega^t = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{c=1}^{C} w_{c\omega} s_c^t$

$\left(+ \sum_{h=1}^{H} w_{h\omega} b_h^{t-1} \right)$

$b_\omega^t = f(a_\omega^t)$

LSTM Forward Pass, Output

(1) $a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \left(+ \sum_{h=1}^{H} w_{hc} b_h^{t-1} \right)$

$b_c^t = g(a_c^t)$

2) $a_\ell^t = \sum_{i=1}^{I} w_{i\ell} x_i^t + \sum_{c=1}^{C} w_{c\ell} s_c^{t-1}$

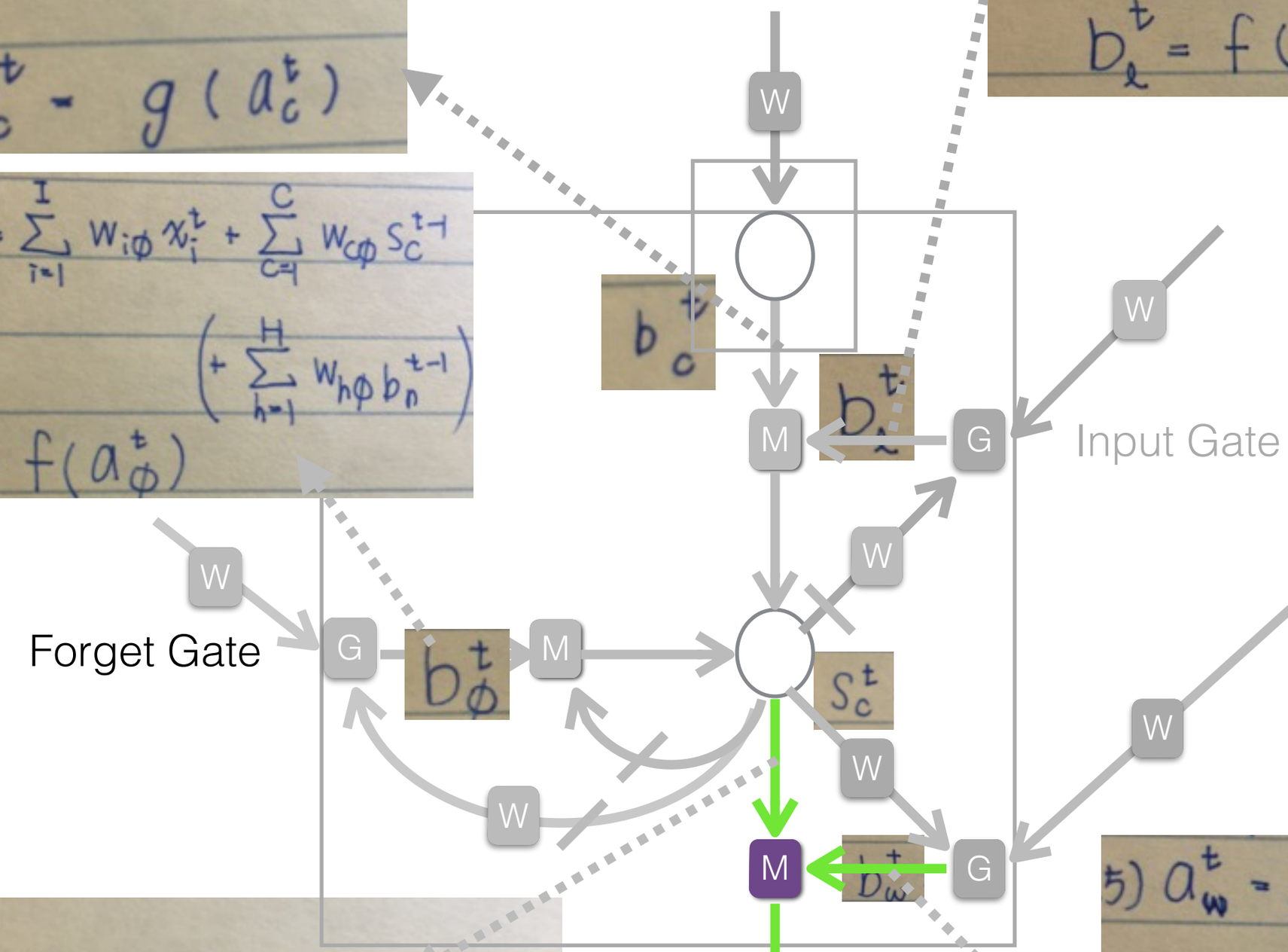
optional $\rightarrow \left(+ \sum_{h=1}^{H} w_{h\ell} b_h^{t-1} \right)$

$b_\ell^t = f(a_\ell^t)$

(3) $a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$

$\left(+ \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} \right)$

$b_\phi^t = f(a_\phi^t)$



4) $s_c^t = \underbrace{b_\phi^t \cdot s_c^{t-1}}_{\text{(3) forget Gate}} + \underbrace{b_\ell^t \cdot b_c^t}_{\text{(2) Input Gate}}$

(6) $b_c^t = b_\omega^t \cdot \underbrace{h(s_c^t)}_{}$

5) $a_\omega^t = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{c=1}^{C} w_{c\omega} s_c^t$

$\left(+ \sum_{h=1}^{H} w_{h\omega} b_h^{t-1} \right)$

$b_\omega^t = f(a_\omega^t)$

LSTM Backward Pass, Output Gate

Cells

$$\delta_c^t = b_t^t \, g'(a_c^t) \, \epsilon_s^t$$

Input Gate

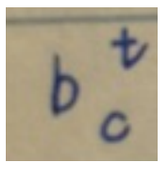
$$\delta_\iota^t = f'(a_\iota^t) \sum_{c=1}^{C} g(a_c^t) \, \epsilon_s^t$$

States

$$\epsilon_s^t = b_w^t \, h'(s_c^t) \, \epsilon_c^t + b_\phi^{t+1} \, \epsilon_s^{t+1} + w_{o\iota} \, b_\iota^{t+1} + w_{c\phi} \, \delta_\phi^{t+1} + w_{cw} \, \delta_w^t$$

Forget Gate

$$\delta_\phi^t = f'(a_\phi^t) \sum_{c=1}^{C} s_c^{t-1} \, \epsilon_s^t$$



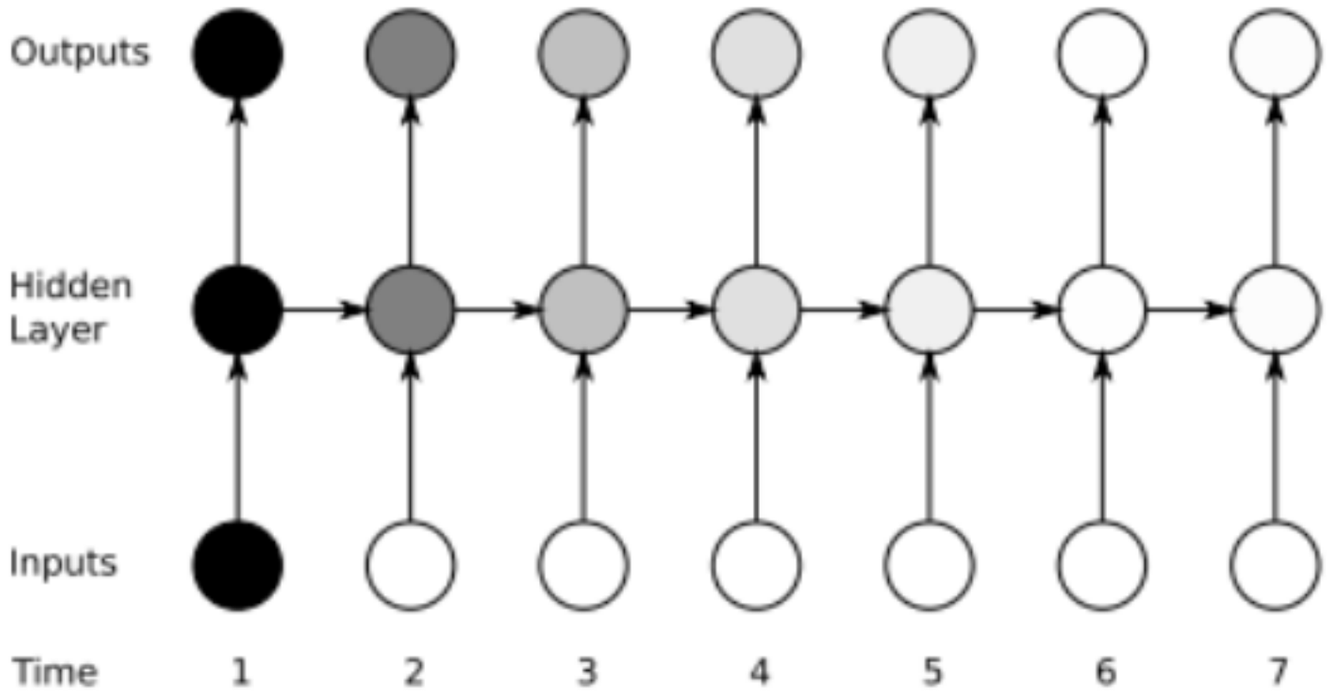
Cell Output

$$\epsilon_c^t \equiv \frac{\partial O}{\partial b_c^t} = \sum_{k=1}^{K} w_{ck} \delta_k^t + \sum_{h=1}^{H} w_{ch} \delta_h^{t+1}$$

Output gate

$$\delta_w^t = f'(a_w^t) \sum_{c=1}^{C} h(s_c^t) \, \epsilon_c^t$$

RNN



LSTM

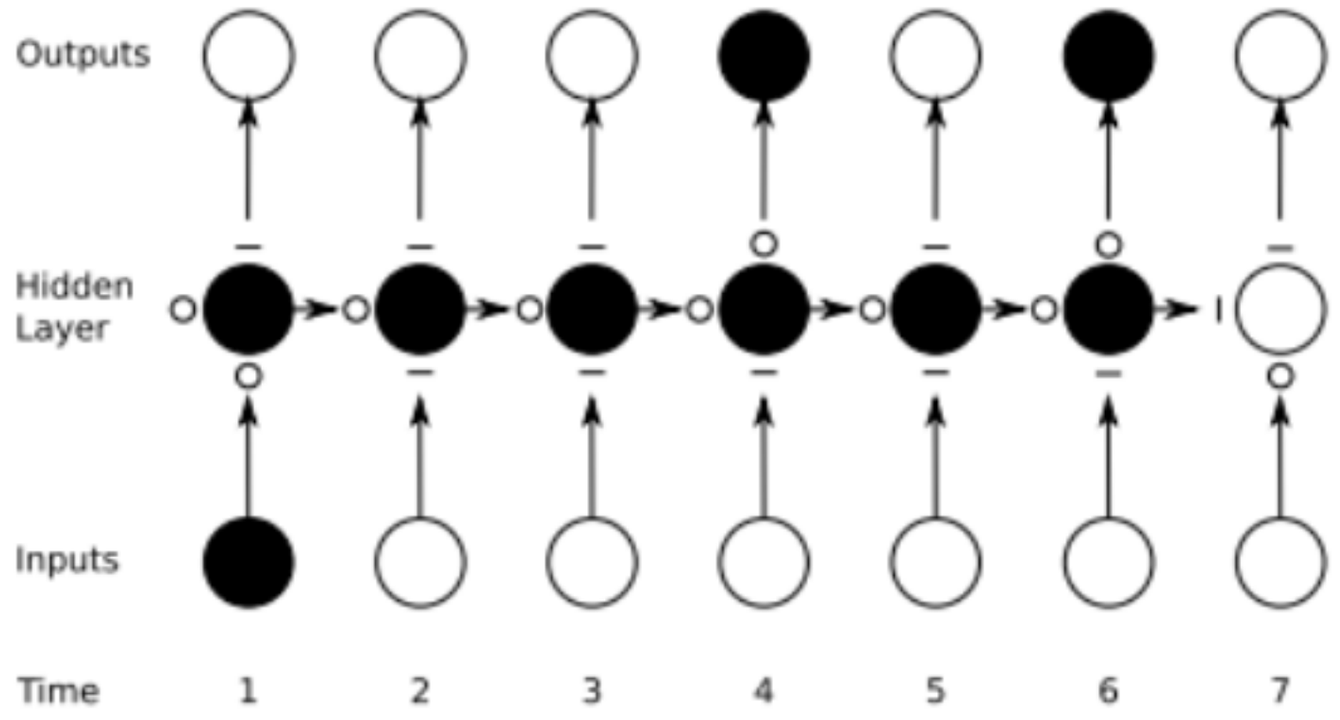
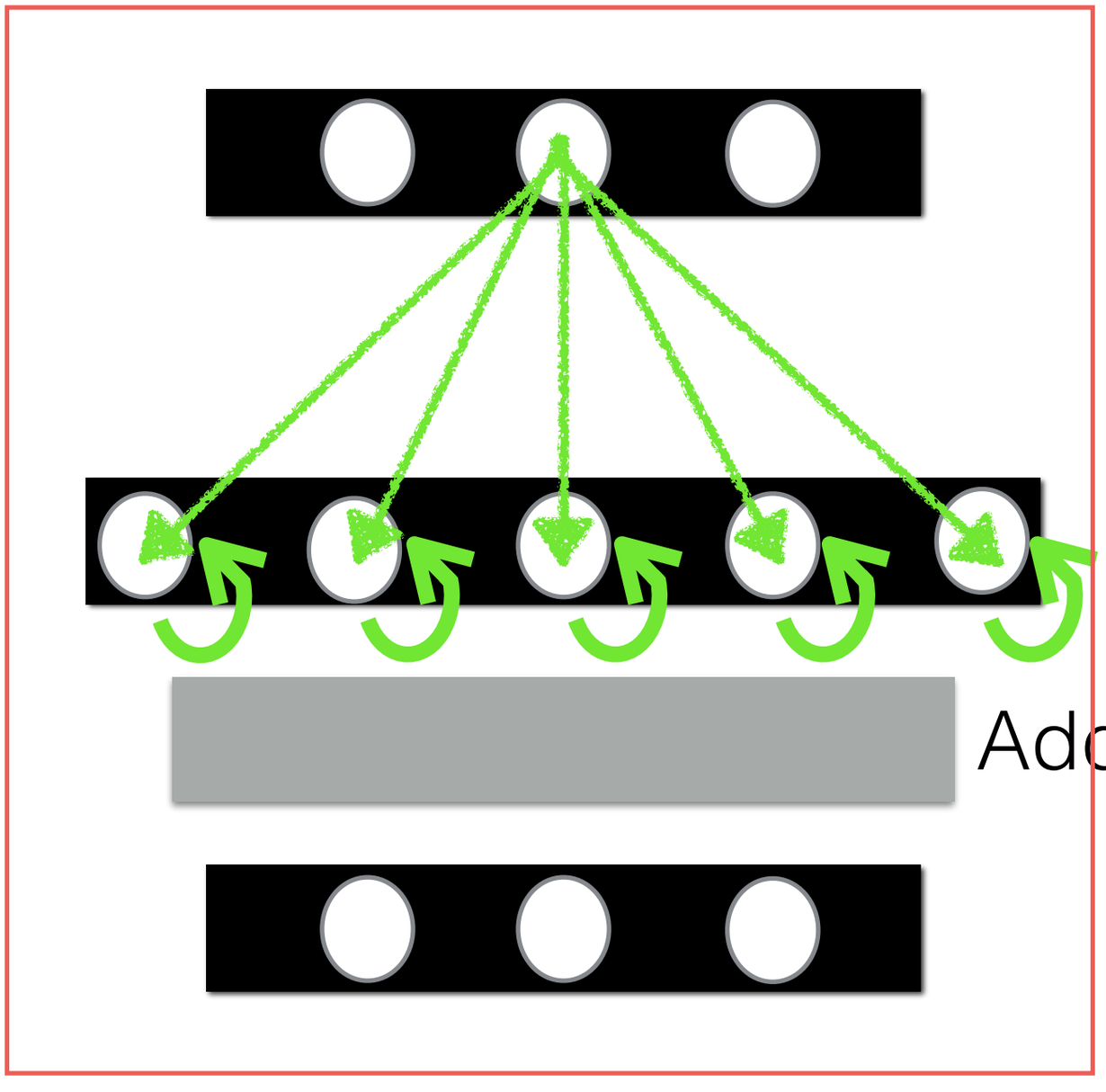
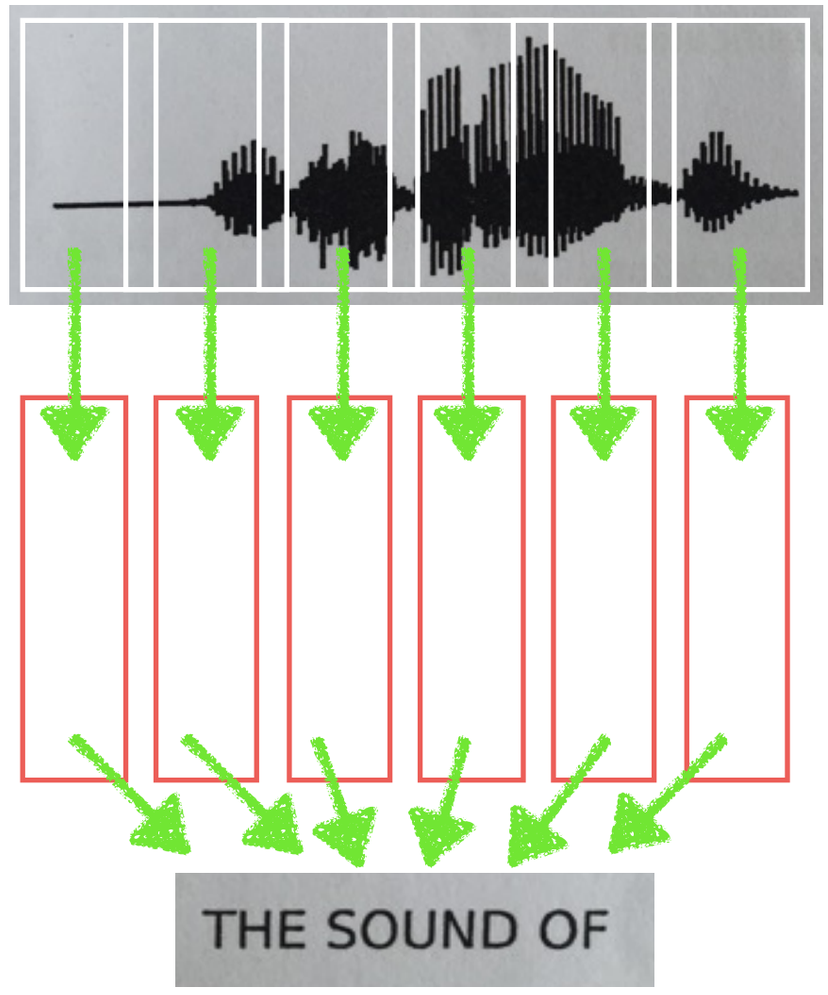


figure from A. Graves, Supervised sequence labelling, p33,35

Connectionist
Temporal
Classification



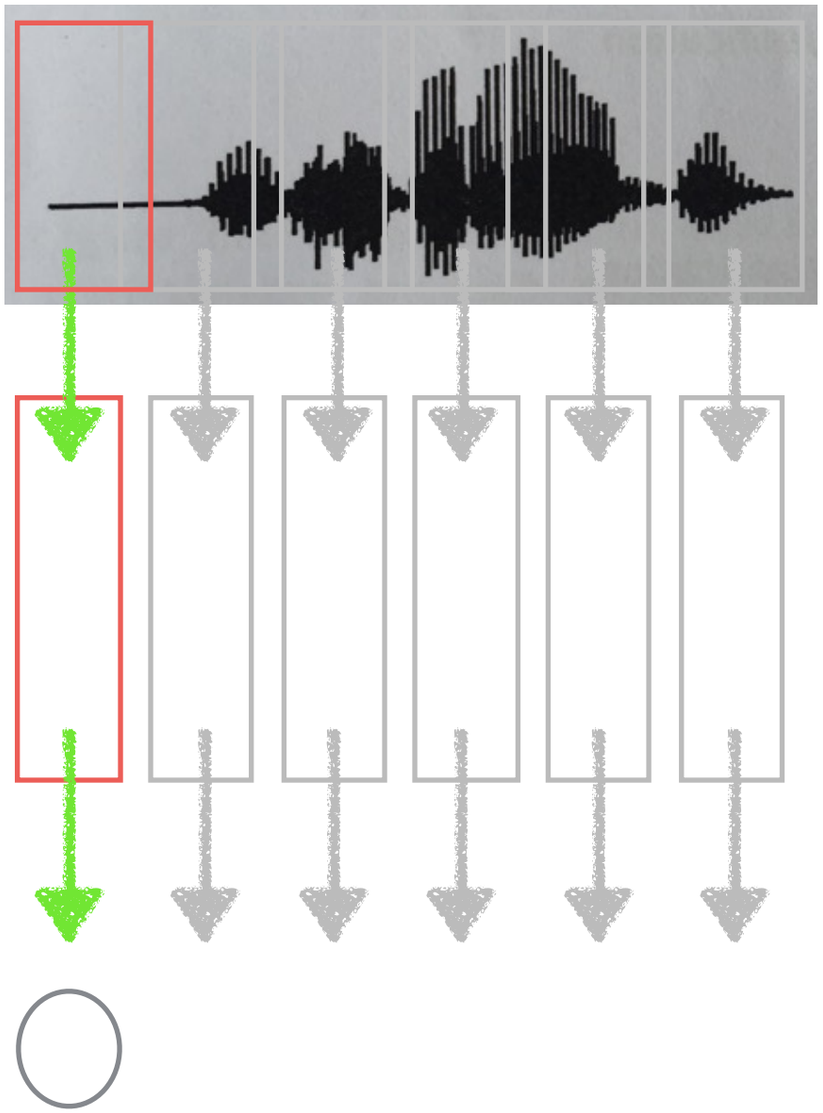


Additional CTC Output Layer

(1) Predict label at any timestep

(2) Probability of Sentence

Predict label
at any tilmestep



Framewise Acoustic Feature

Label

A B C D E F G I J
K L M N O P Q R
S T U V W X Y Z
{space}
{blank}
others like ‘ “ .

$$\mathbf{y} = \mathcal{N}_w(\mathbf{x}) \quad \mathcal{N}_w : (\mathbb{R}^{\bar{m}})^T \mapsto (\mathbb{R}^n)^{\bar{T}}$$

$$p(\pi|\mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t, \ \forall \pi \in L'^T$$

THE SOUND OF

Probability of Sentence

$$\mathcal{B} : L'^{T} \mapsto L^{\leq T}$$

paths: $\pi \in L'^{T}$

Thh..e S..outtd ooof

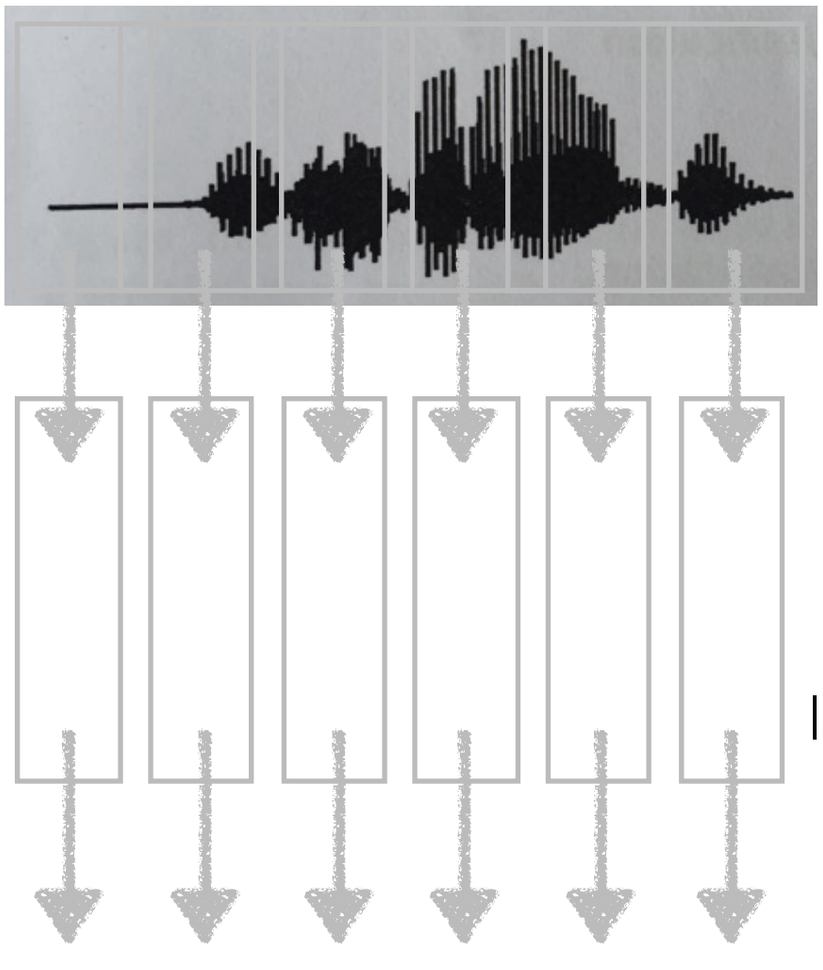
Step 1, remove repeated labels

Th.e S.outd of

Step 2, remove all blanks

Predict

The Soutd of



labelling: $\mathbf{l} \in L^{\leq T}$

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x})$$

$\mathcal{B}$

Target THE SOUND OF

Now, we can do the maximum likelihood training

Output Decoding

Choose the labelling : $h(\mathbf{x}) = \arg\max_{\mathbf{l} \in L^{\leq T}} p(\mathbf{l}|\mathbf{x})$

Best Path, the most probable path will correspond to most probable labelling

$$h(\mathbf{x}) \approx \mathcal{B}(\pi^*)$$
$$\text{where } \pi^* = \arg\max_{\pi \in N^t} p(\pi|\mathbf{x})$$

Something wrong here. Prefix Search would be better

Maximum likelihood

target

$$O^{ML}(S, N_W) = -\sum_{(x,z)\in S} \ln\left(p(z|x)\right)$$

O ← objective function

S ← training set

N_W ← weight

x ← acoustic

objective function : O

$$\frac{\partial O}{\partial y_k^t} = -\frac{\partial \ln\left(p(z|x)\right)}{\partial y_k^t}$$

y_k^t ← network output unit k at time t

Probability of observing label k at time t

How to connect target label and RNNs Output

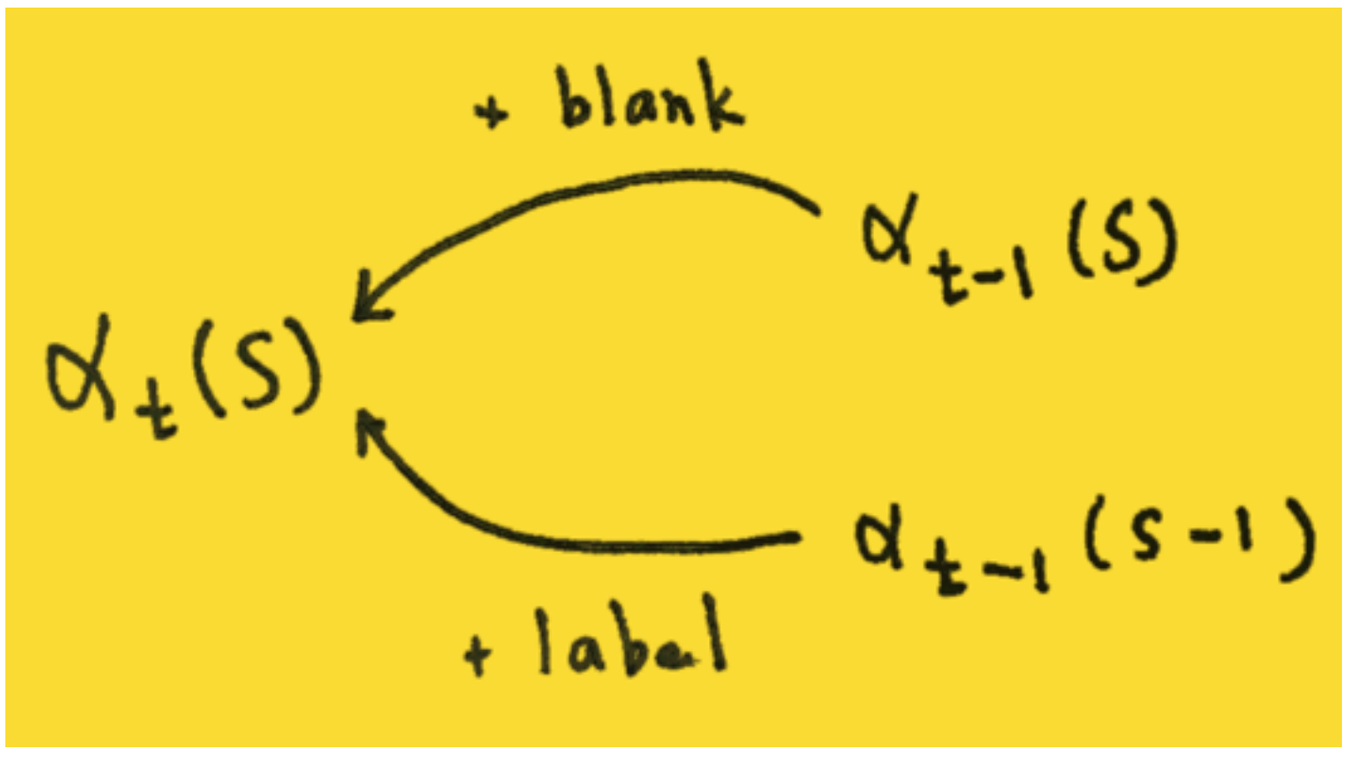
Sum over paths probability, forward

first s symbol

$$\alpha_t(s) \equiv \sum_{\pi \in N^T :} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}}$$

timestep

$$B(\pi_{1:t}) = l_{1:s}$$



$$\alpha_1(1) = y^1_b$$ ← blank symbol

$$\alpha_1(2) = y^1_{l_1}$$ ← first symbol of target

$$\alpha_1(s) = 0 \quad , \forall s > 2$$

only allow transition
(1) between blank and non-blank label
(2) distinct non-blank label

$$\bar{\alpha}_t(s) \equiv \alpha_{t-1}(s) + \alpha_{t-1}(s-1)$$

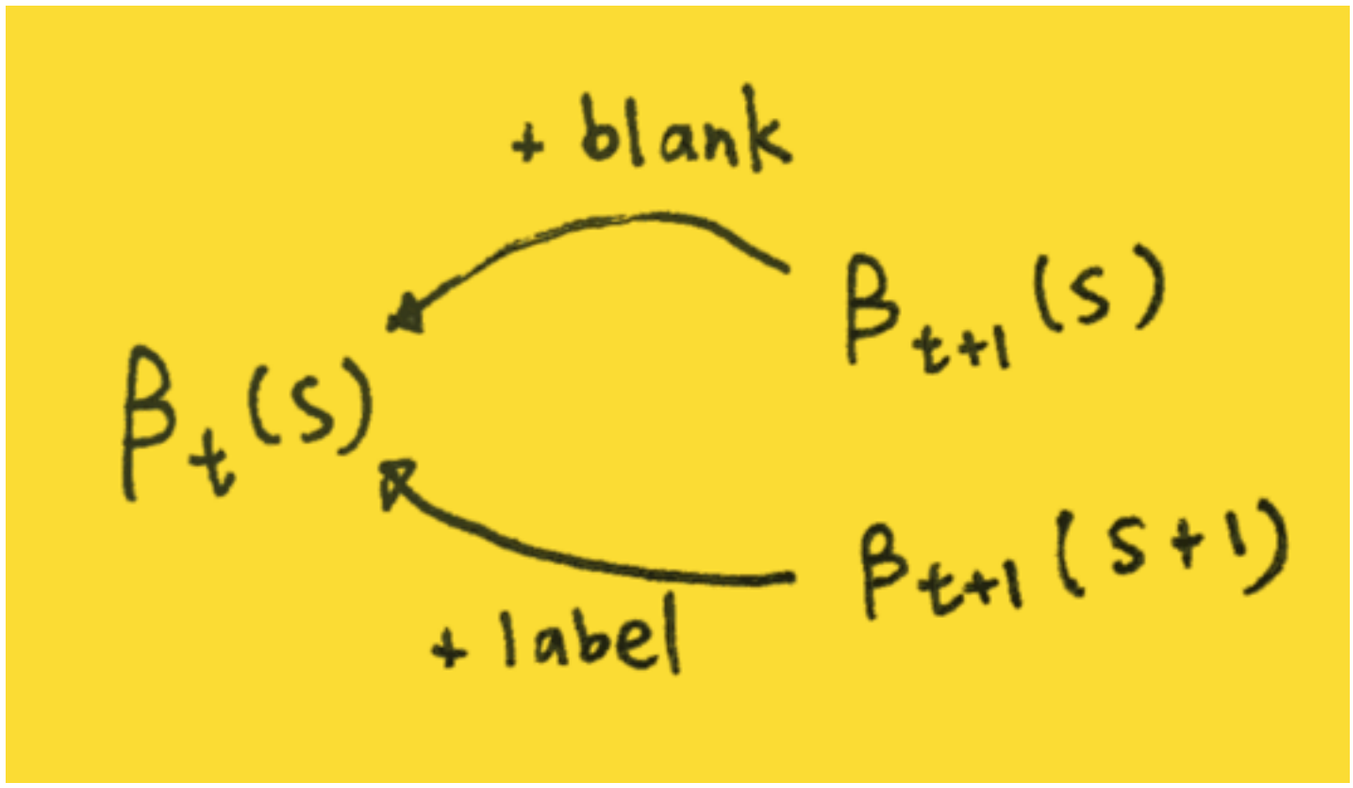
$$\alpha_t(s) = \begin{cases} \bar{\alpha}_t(s) \cdot y^t_{l'_s} & \text{if } l'_s = b \text{ or } l'_{s-2} = l'_s \\ (\bar{\alpha}_t(s) + \alpha_{t-1}(s-2)) y^t_{l'_s} \end{cases}$$

l_s = .
s-0 : .h.e.l.
s-1 : .h.e.l
s-2 : .h.e.
(s-2 blank to blank)

l_s = l
s-0 : .h.e.l.l
s-1 : .h.e.l.
s-2 : .h.e.l
(s-2 same non-blank label)

Sum over paths probability, backward

$$\beta_t(s) \equiv \sum_{\pi \in N^T : B(\pi_{t,T}) = l_{s:|l|}} \prod_{t'=t}^{T} y^{t'}_{\pi_{t'}}$$



init

$$\beta_T(|l'|) = y_b^T$$

$$\beta_T(|l'|-1) = y_{l_{|l|}}^T$$

$$\beta_T(s) = 0, \quad \forall s < |l'|-1$$

only allow transition
(1) between blank and non-blank label
(2) distinct non-blank label

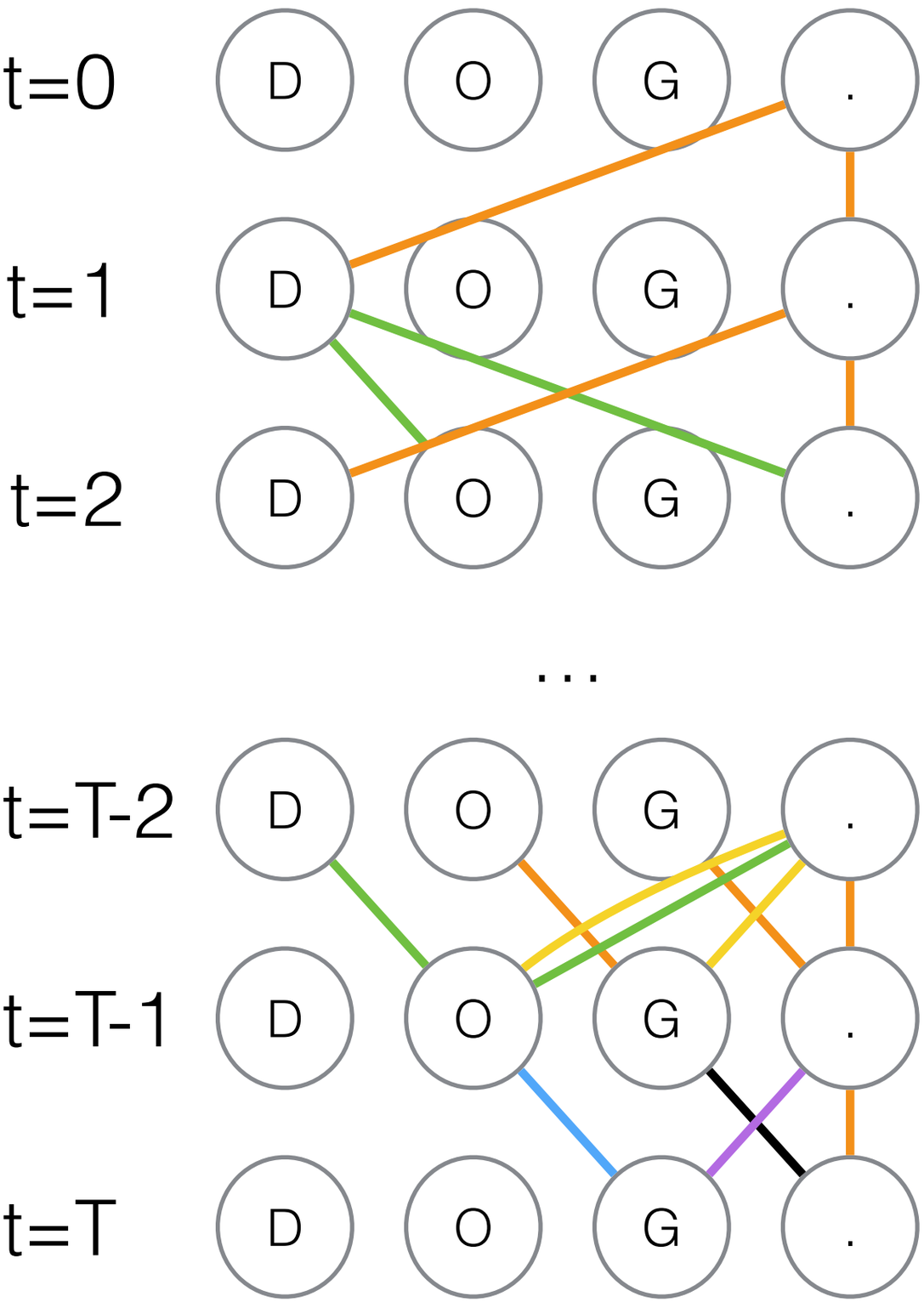
$$\bar{\beta}_t(s) \equiv \beta_{t+1}(s) + \beta_{t+1}(s+1)$$

$$\beta_t(s) = \begin{cases} \bar{\beta}_t(s) \cdot y_{l'_s}^t & \text{if } l'_s = b \text{ or } l'_{s+2} = l'_s \\ (\bar{\beta}_t(s) + \beta_{t+1}(s+2)) \cdot y_{l'_s}^t & \end{cases}$$

l_s = .
s+0 : .l.l.o.
s+1 : l.l.o.
s+2 : .l.o.
(s+2 blank to blank)

l_s = l
s+0 : l.l.o.
s+1 : .l.o.
s+2 : l.o.
(s+2 same non-blank label)

Sum over paths probability



Target : dog -> .d.o.g.

s=1 .

s=2 .d

s=3 .d.

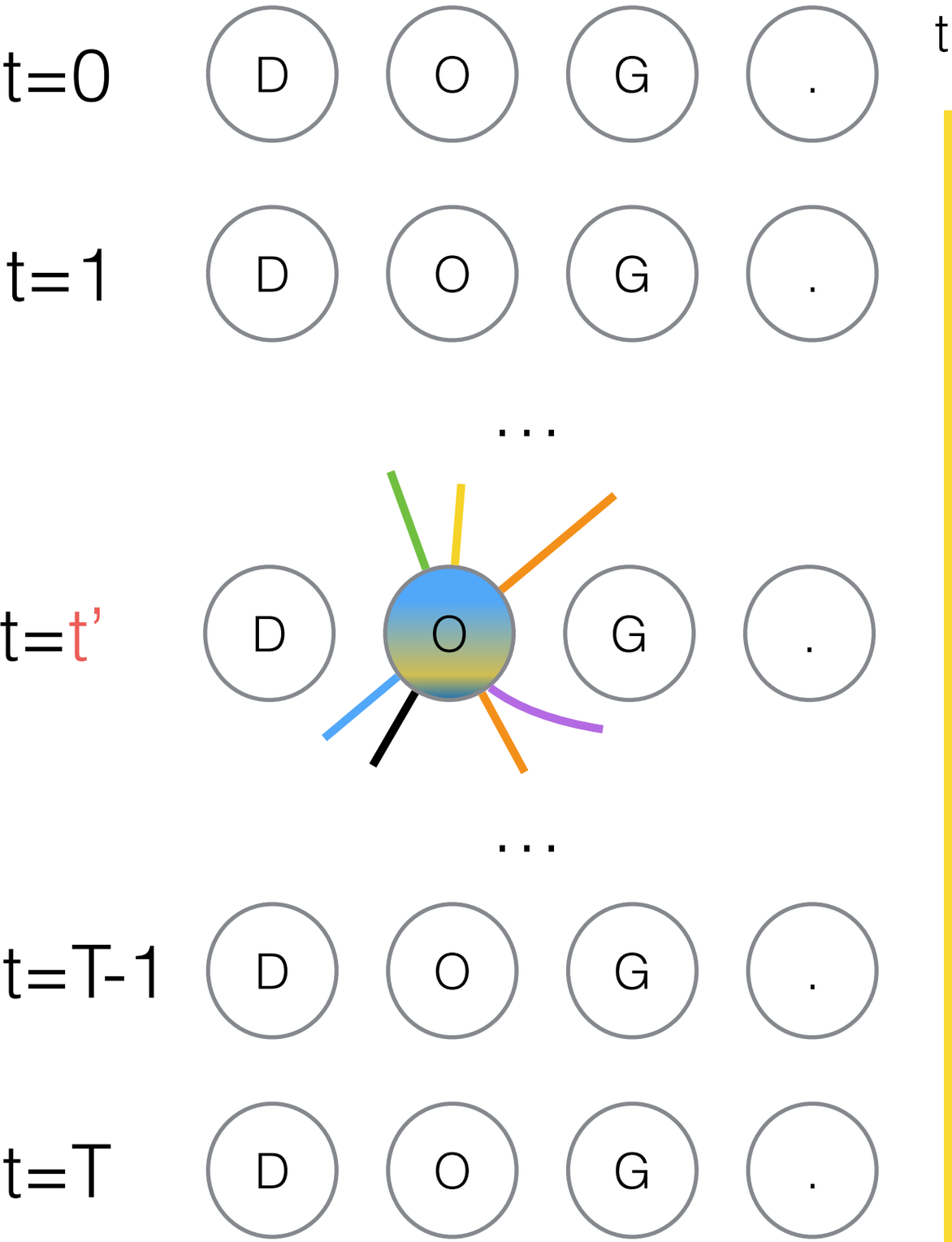
s=4 .d.o

s=5 .d.o.

s=6 .d.o.g

s=7 .d.o.g.

Sum over paths probability



backward-forward :
the probability of all the paths corresponding to l
that go through the symbol at time t

$$\alpha_t(s)\cdot\beta_t(s) = \sum_{\substack{\pi\in B^{-1}(l):\\ \pi_t = l'_s}} y^t_{l'_s} \prod_{t=1}^{T} y^t_{\pi_t}$$

$$\Rightarrow \frac{\alpha_t(s)\cdot\beta_t(s)}{y^t_{l'_s}} = \sum_{\substack{\pi\in B^{-1}(l):\\ \pi_t = l'_s}} p(\pi\mid x)$$

$$p(l\mid x) = \sum_{s=1}^{|l'|} \frac{\alpha_t(s)\cdot\beta_t(s)}{y^t_{l'_s}}$$

$$\frac{\partial O}{\partial y_k^t} = -\frac{\partial \ln(p(z|x))}{\partial y_k^t}$$

$$= \frac{-1}{p(z|x)} \cdot \frac{\partial p(z|x)}{\partial y_k^t}$$

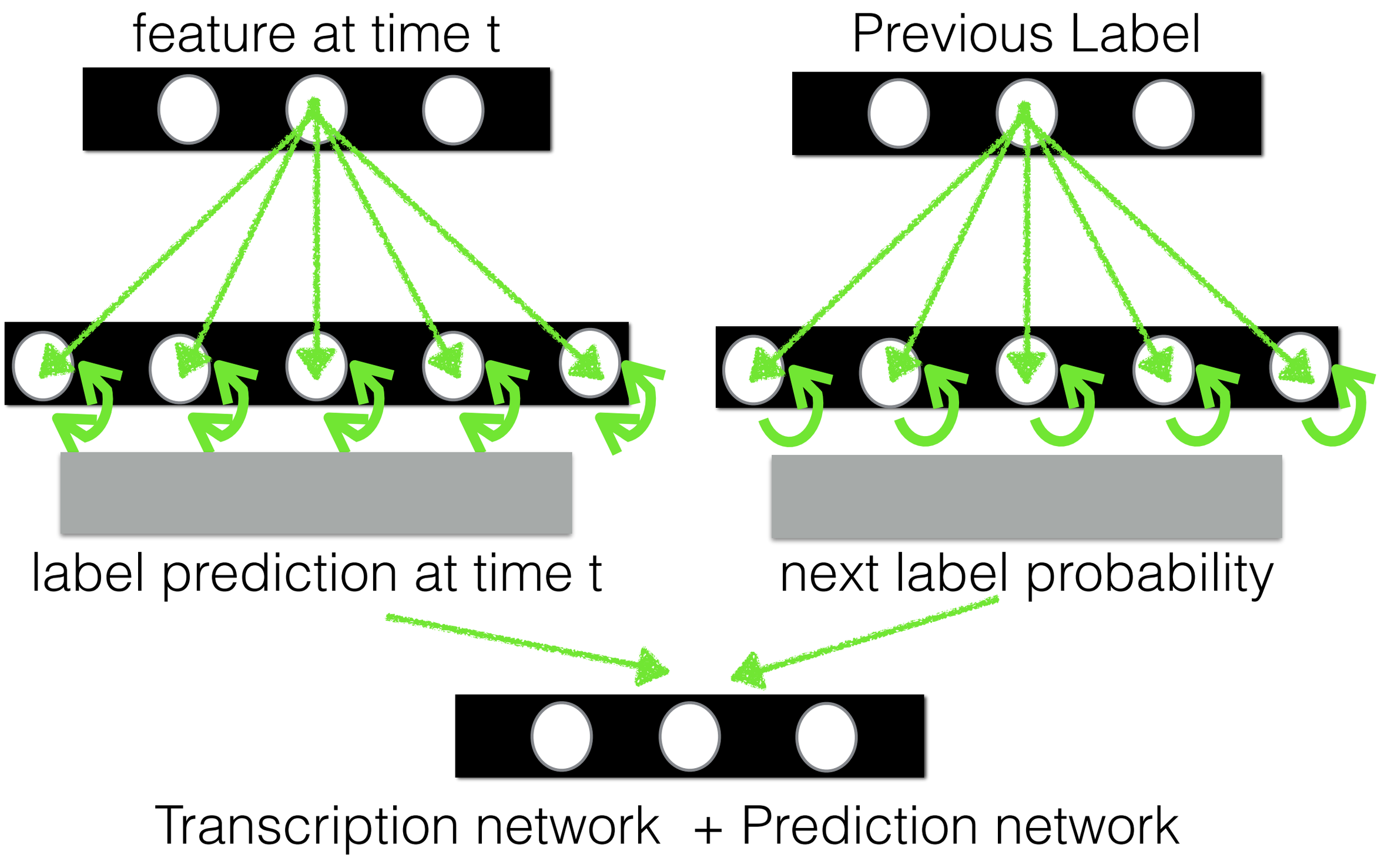
$$= \frac{-1}{\sum_{s=1}^{|z'|} \frac{\alpha_t(s)\cdot\beta_t(s)}{y^t_{z'_s}}} \cdot \frac{1}{y_k^{t\,2}} \sum_{s \in lab(z,k)} \alpha_t(s)\cdot\beta_t(s)$$

$$Z_t \stackrel{\text{def}}{=} \sum_{s=1}^{|\mathbf{l}'|} \frac{\hat{\alpha}_t(s)\hat{\beta}_t(s)}{y^t_{\mathbf{l}'_s}}$$

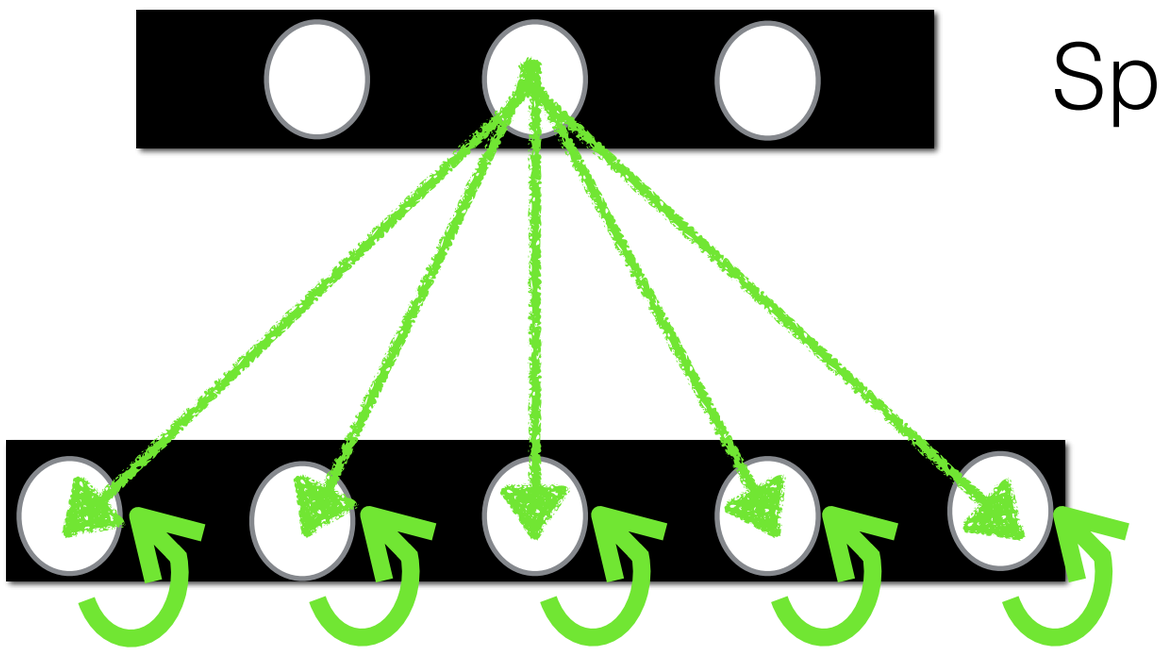
last we derivates w.r.t. unnormalised outputs, get error signal

$$\frac{\partial O^{ML}(\{(\mathbf{x},\mathbf{z})\},\mathcal{N}_w)}{\partial u_k^t} = y_k^t - \frac{1}{y_k^t Z_t} \sum_{s \in lab(\mathbf{z},k)} \hat{\alpha}_t(s)\hat{\beta}_t(s)$$

A. Graves. Sequence Transduction with Recurrent Neural Networks



A. Graves. Towards End-to-end Speech Recognition with Recurrent Neural Networks (**Google Deepmind**)



Spectrogram feature

$$CTC(\boldsymbol{x}) = -\log \Pr(\boldsymbol{y}^*|\boldsymbol{x})$$

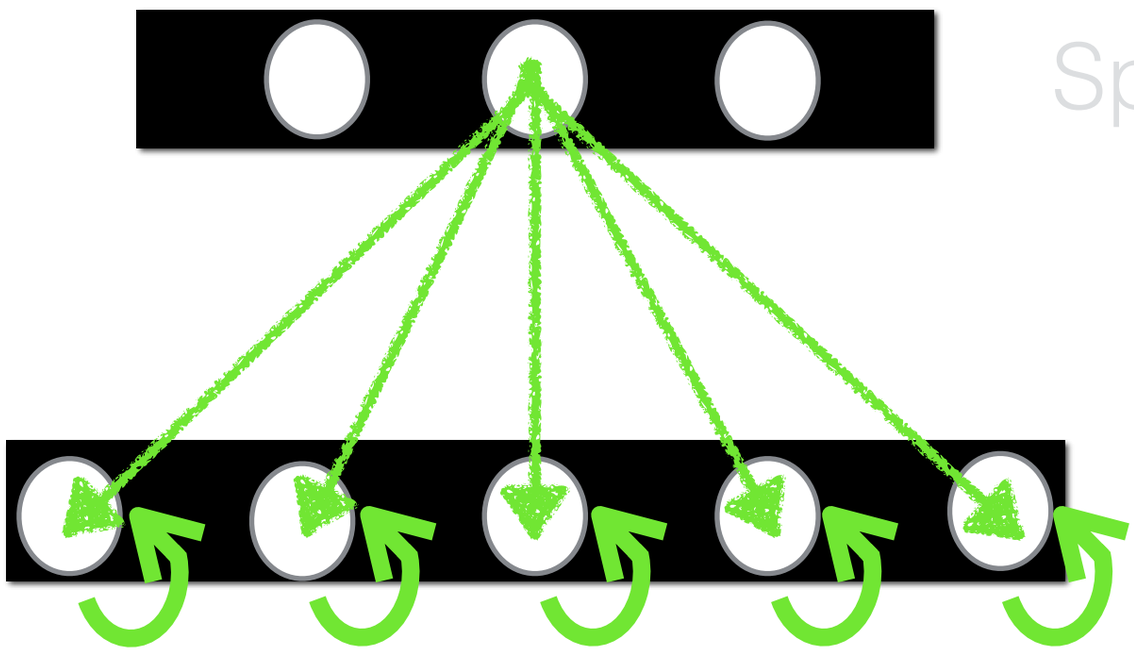
Additional CTC Output Layer

Additional WER Output Layer

$$\mathcal{L}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x})\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$$

Minimise Objective Function -> Minimise Loss Function

Andrew Y. Ng. Deep Speech: Scaling up end-to-end speech recognition (**Baidu Research**)



Spectrogram feature

Additional CTC Output Layer

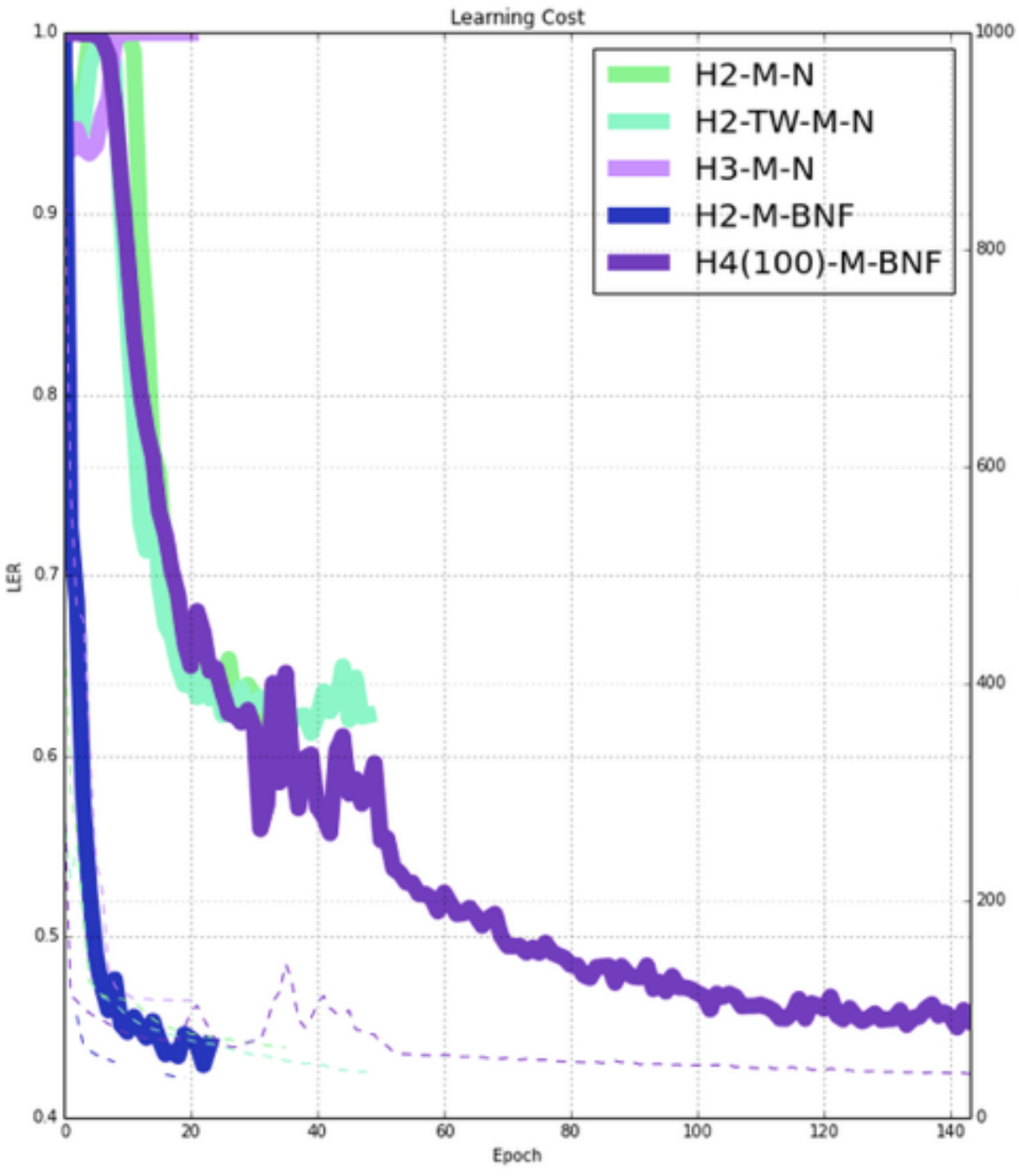
Additional WER Output Layer

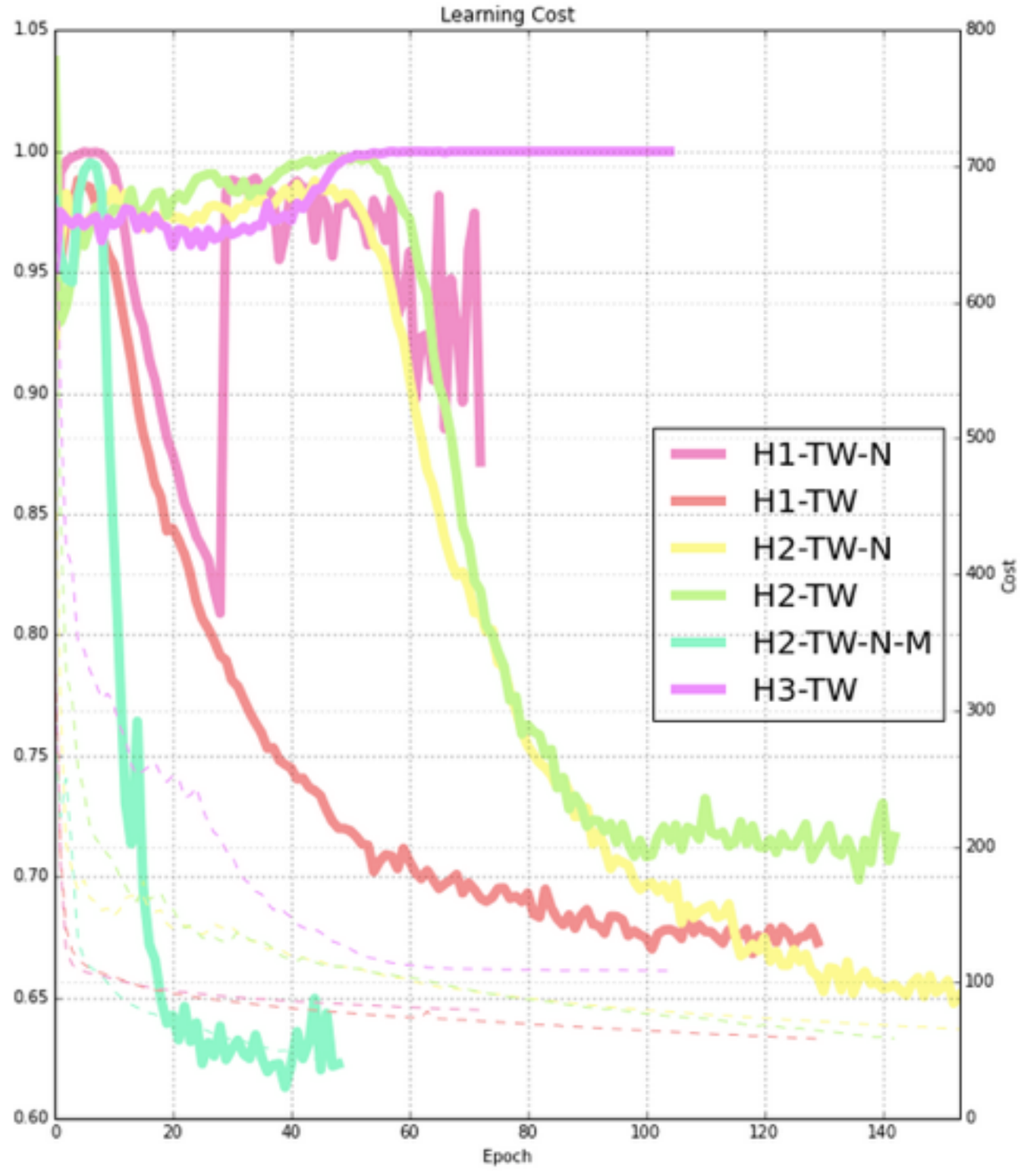
RNN Output : arther n tickets for the game
Decode Target: are there any tickets for the gam

Language Model Transcription

$$Q(L) = \log(P(L \mid x)) + a * \log(P_{LM}(L)) + b * \text{word\_count}(L)$$

Low-resources language experiment





Target: ? ka didto sigi ra og tindog

Output: h bai to giy ngtndog

test 0.44,
test 0.71

Reference

1. Boulard and Morgan (1994) Connectionist speech recognition A hybrid approach.
2. A. Grave. Supervised Sequence Labelling with Recurrent Neural Networks
3. A. Grave. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Network.
4. A. Graves. Sequence Transduction with Recurrent Neural Networks
5. A. Graves. Towards End-to-end Speech Recognition with Recurrent Neural Networks (Google Deepmind)
6. Andrew Y. Ng. Deep Speech: Scaling up end-to-end speech recognition (Baidu Research)
7. standford CTC: https://github.com/amaas/stanford-ctc
8. R. Pascanu , On the difficulty of training recurrent neural networks
9. L. Besacier. Automatic Speech Recognition for Under-Resourced Languages: A Survey
10. A. Gibiansky. http://andrew.gibiansky.com/blog/machine-learning/speech-recognition-neural-networks/
11. LSTM Mathematic formalism (mandarin)

http://blog.csdn.net/u010754290/article/details/47167979

11. Assumption of Markov Model

http://jedlik.phy.bme.hu/~gerjanos/HMM/node5.html

12. Story about ANN-HMM Hybrid (mandarin)

HMM : http://www.taodocs.com/p-5260781.html

13. Generative model v.s. discriminative model

http://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-discriminative-algorithm

Memory Network :
A. Grave. Neural Turing Machine

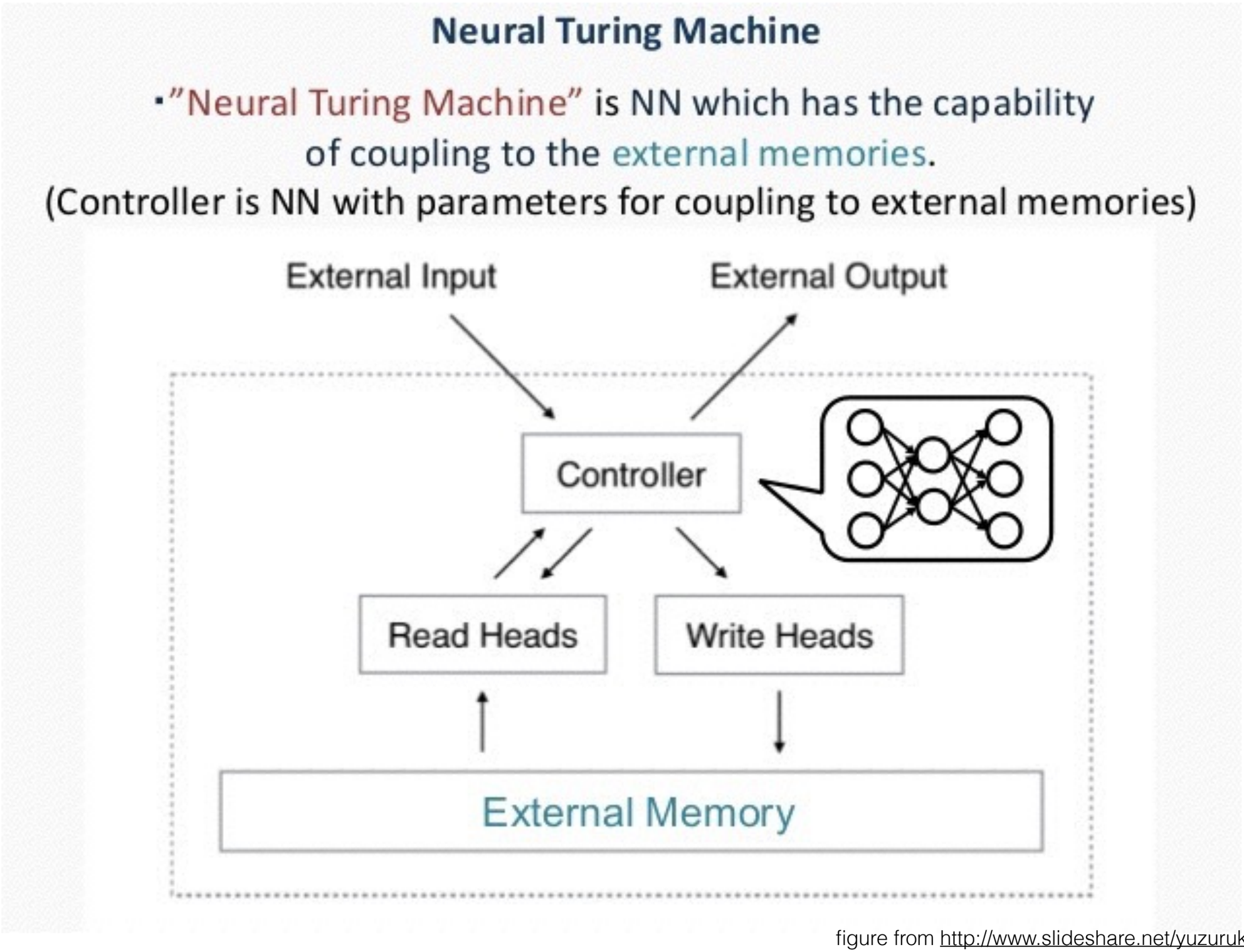


figure from http://www.slideshare.net/yuzurukato/neural-turing-machines-43179669

Appendix A. Generative v.s. discriminative

A generative model learns the **joint** probability p(x,y).
A discriminative model learns the **conditional** probability p(y|x).

data input : (1,0), (1,0), (2,0), (2,1)

Generative

	y=0	y=1
x=1	1/2	0
x=2	1/4	1/4

Discriminative

	y=0	y=1
x=1	1	0
x=2	1/2	1/2

According to Andrew Y. Ng. On Discriminative vs. Generative classifier: A comparison of logistic regression and naive Bayes

The overall gist is that **discriminative models generally outperform generative models in classification tasks**.