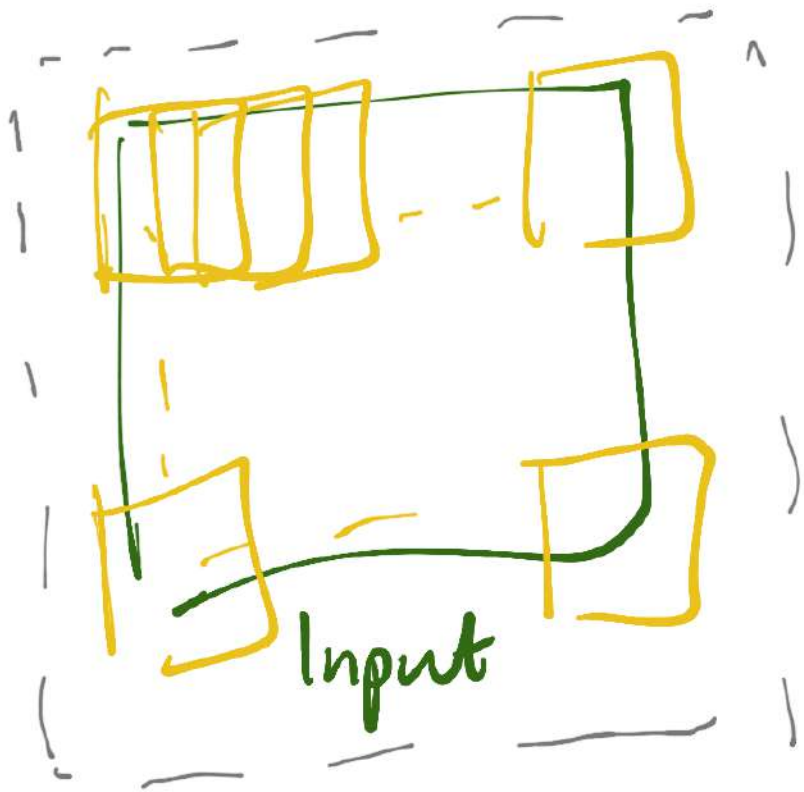
Advanced Machine Learning, 29 Aug 2019

last time Convolutional NN



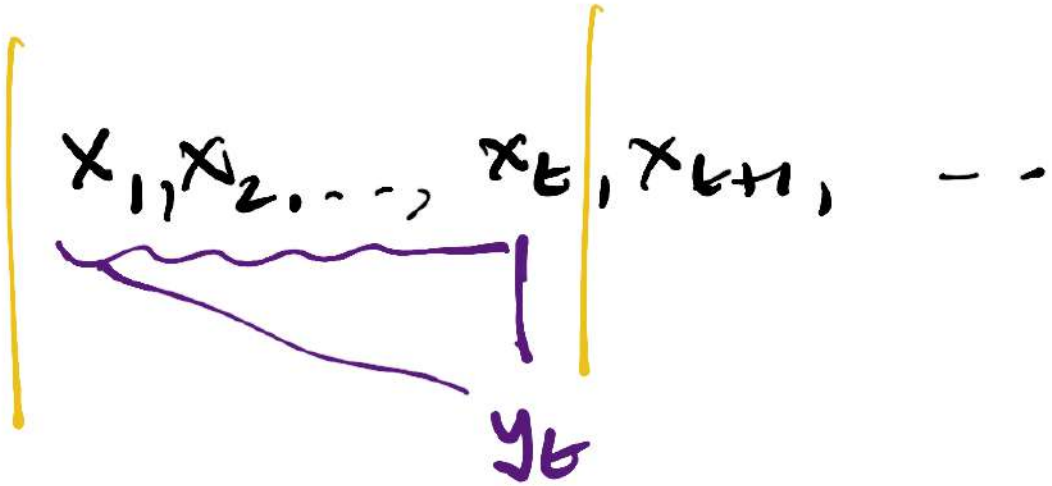
Different size image, we can adapt the kernel to slide over that image

Conventional (fully connected) network would behave very differently across input sizes

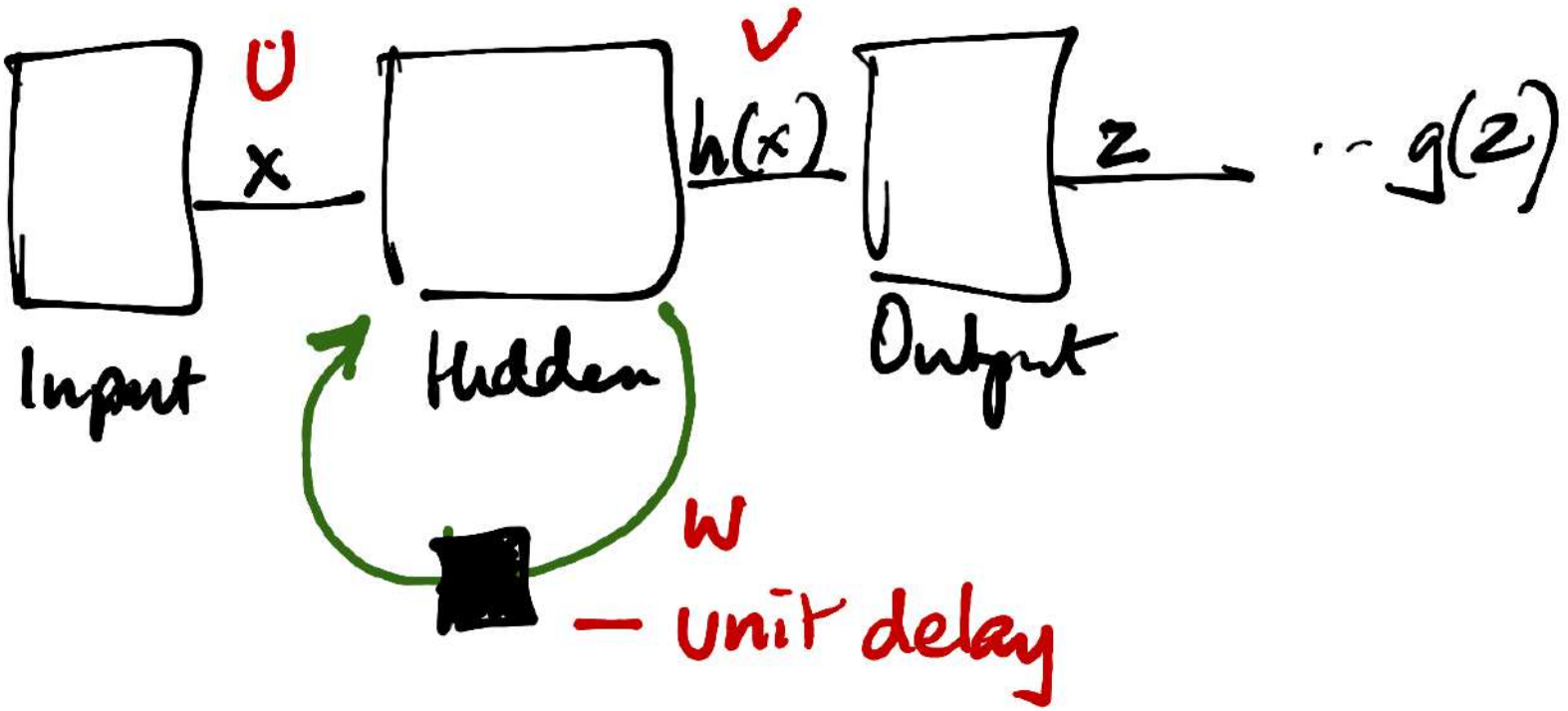
Sequential inputs

Speech recognition / synthesis

x_i - input at time



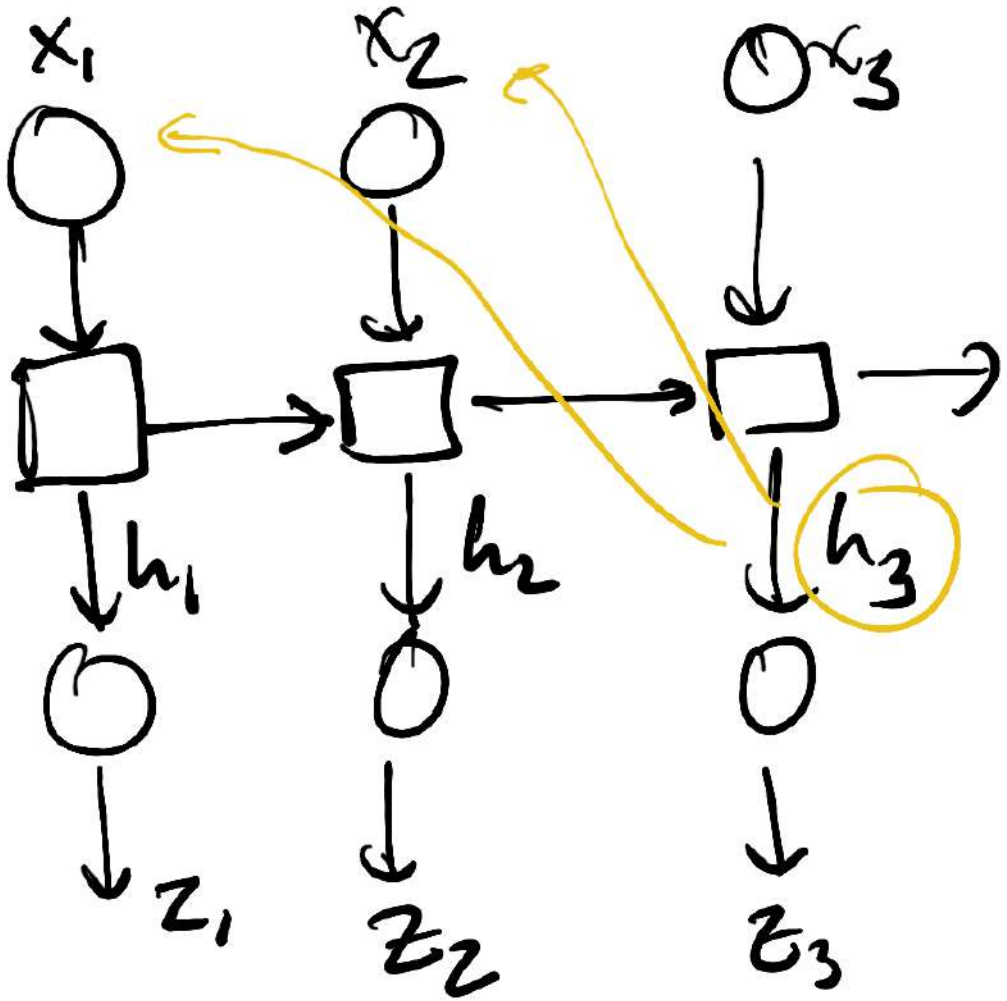
Use hidden layers to "remember" past inputs



Recurrent neural network

Convert this to an acyclic network –

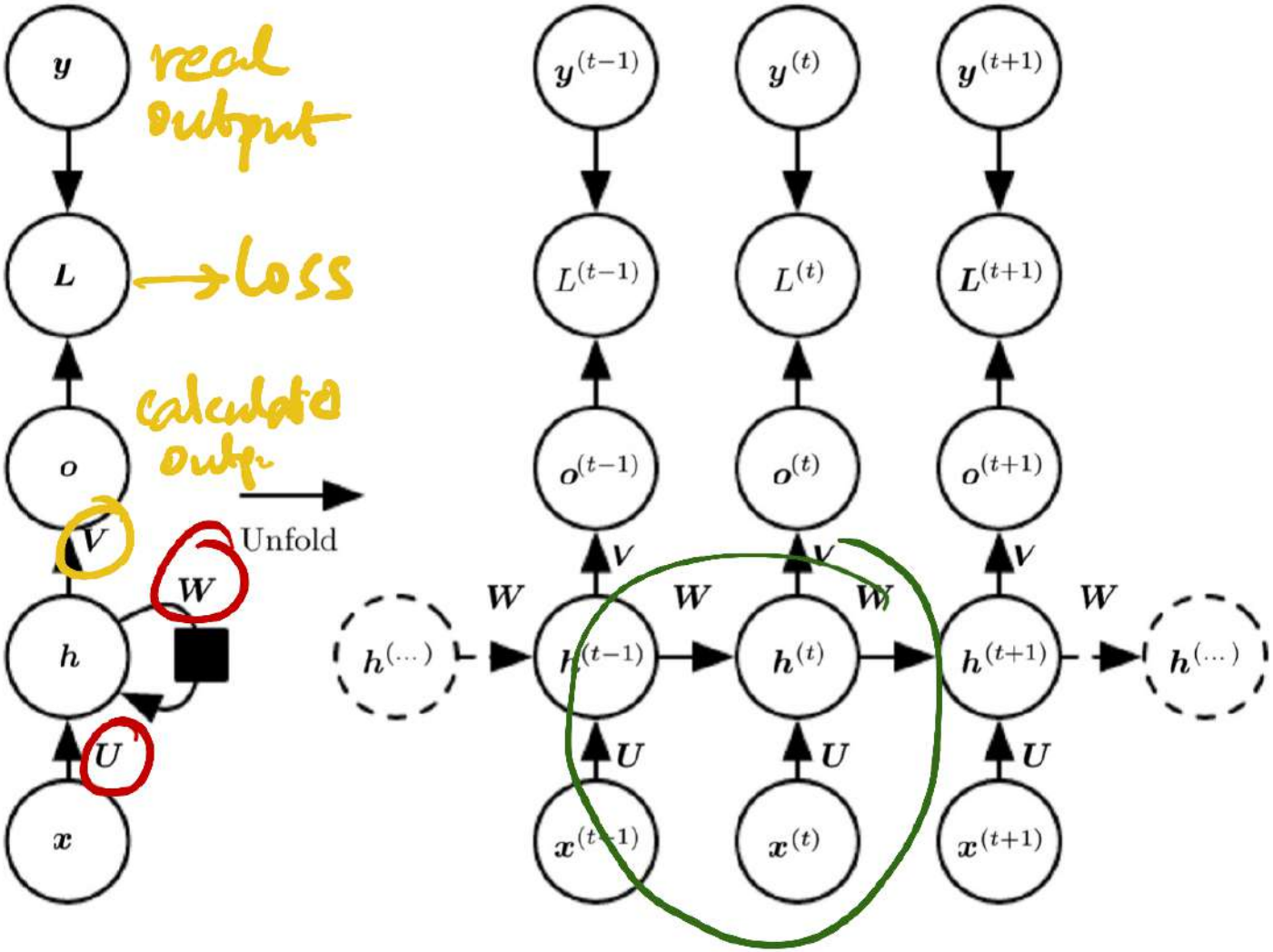
Unrolling / Unfolding



Uniformity in time of how output depends on input

Types of RNN

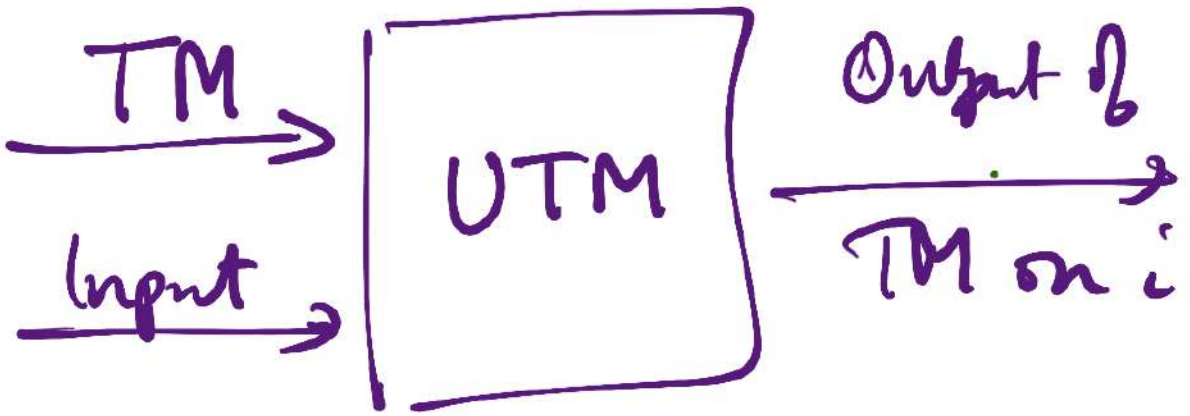
History feedback



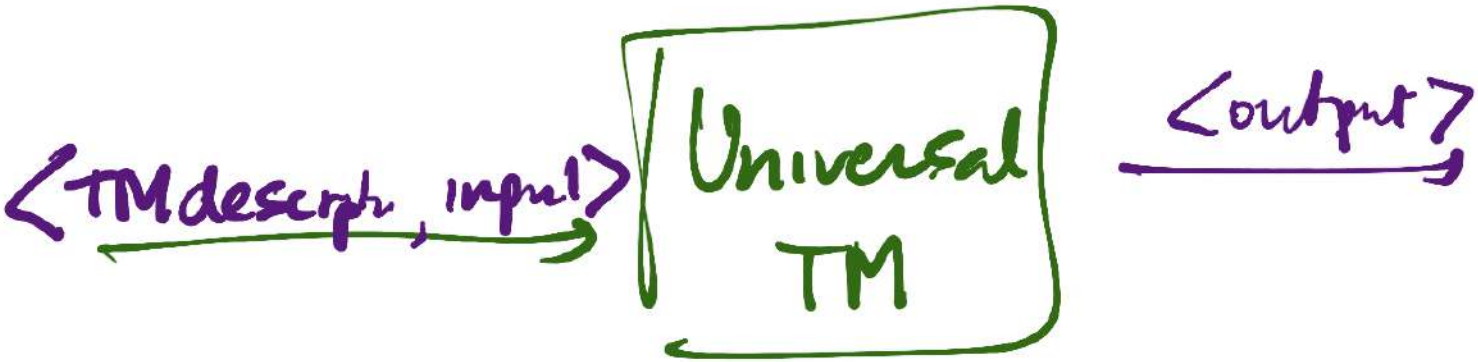
Universal

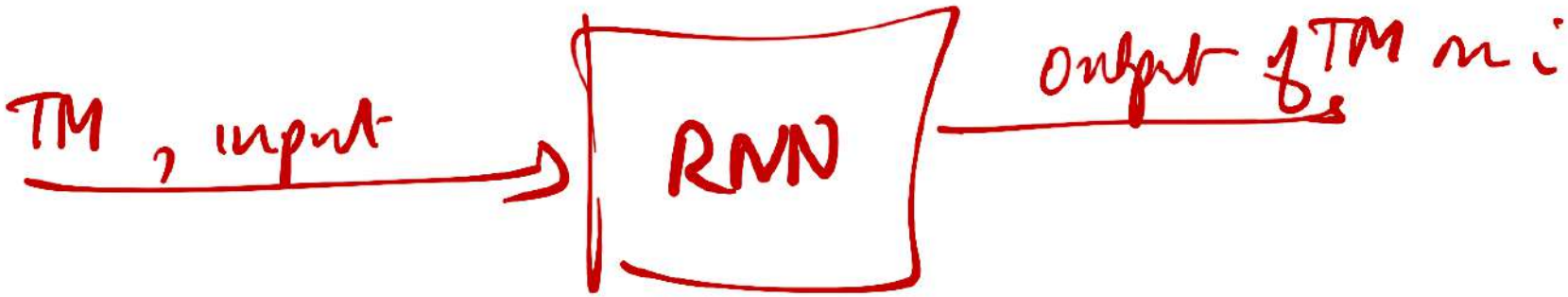
Turing Machine – Universal model of computation

Universal Turing Machine

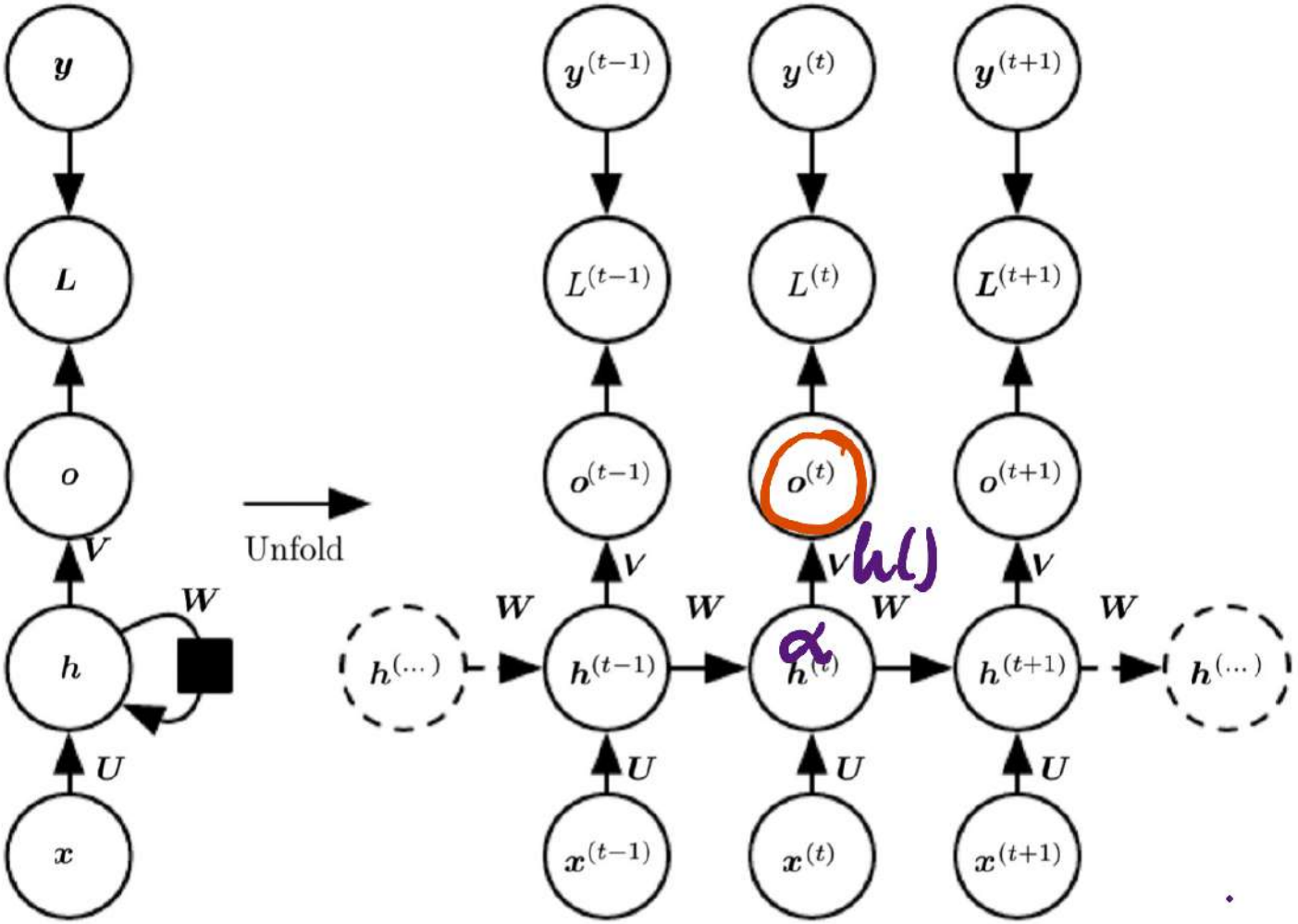


RNNs can simulate universal TMs





∃ a universal RNN
with 886 nodes



$$a^t = W h^{t-1} + U x^t + b$$

$$h^t = \tanh(a^t) = 2\sigma(2z) - 1$$

$$o^t = V h^t + c$$

$$z^t = \text{softmax}(o^t)$$

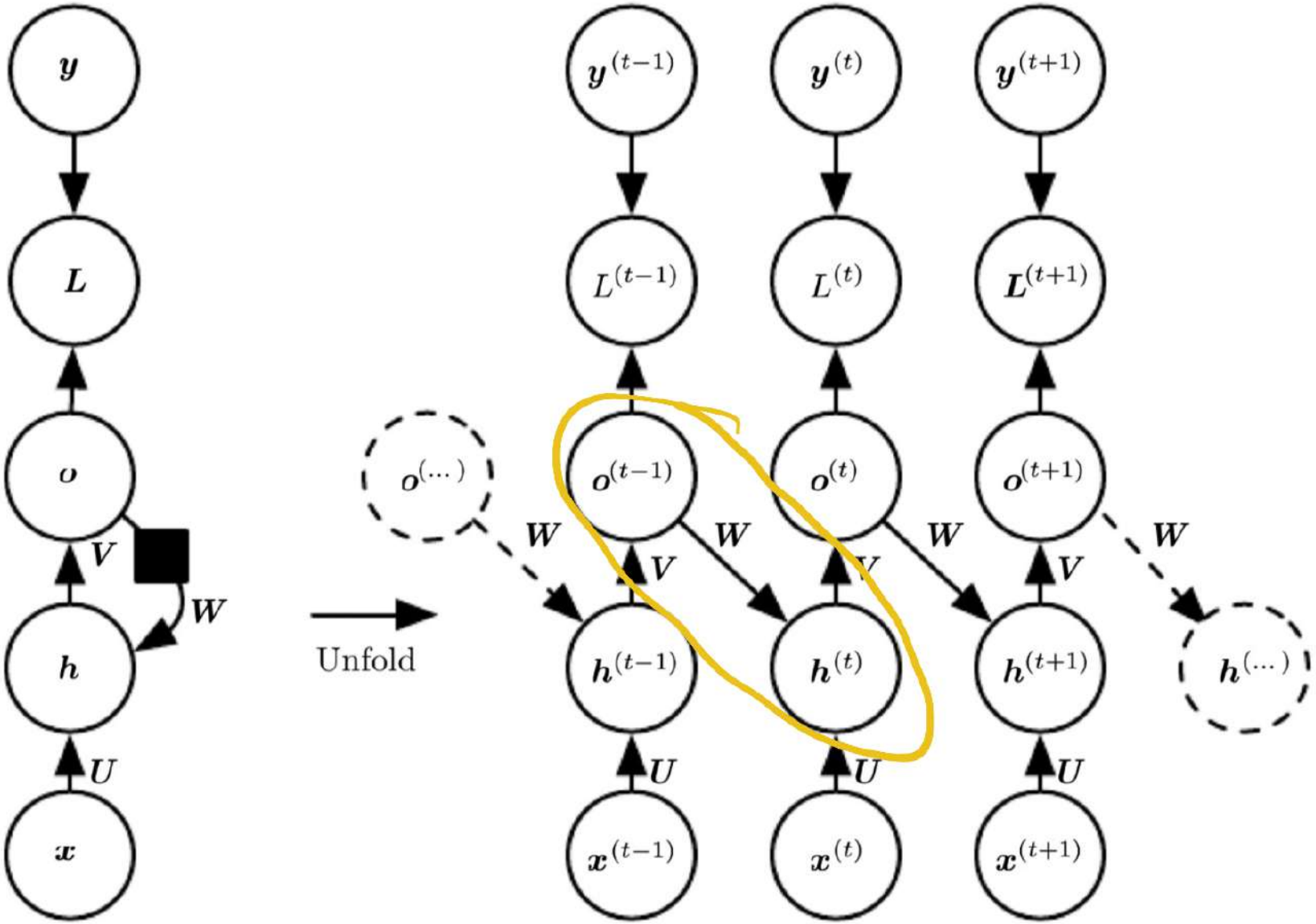
Forward & backpropagation through all time layers

Back propagation through time

$$\frac{\partial z}{\partial W_i}$$

Is there a better way?

Type 2 Output feeds back to hidden layer



These networks are not universal

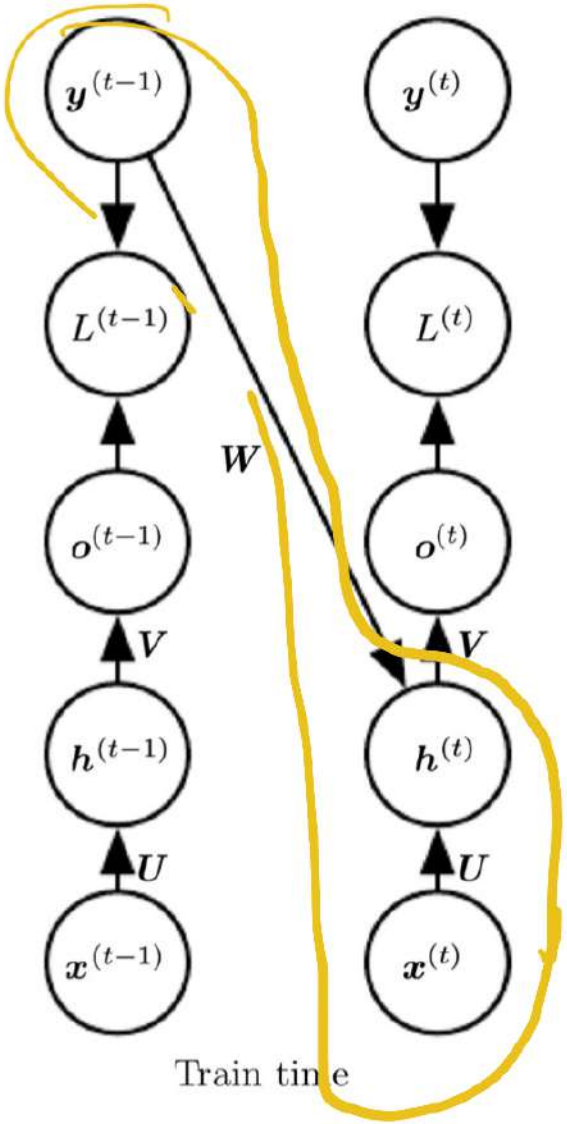
Training

h^t depends on o^{t-1}, x^t

We know y^{t-1}

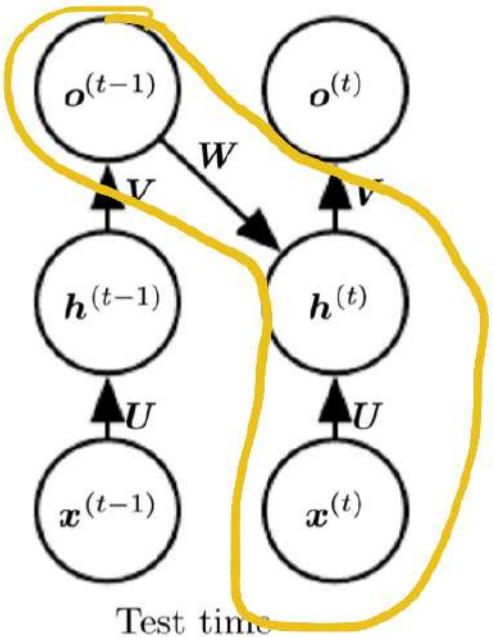
Ideally $o^{t-1} = y^{t-1}$

Why not use y^{t-1} for backpropagation?



Train time

Real Picture



Test time

"Teacher Forcing" – Use $y^{(t-1)}$ instead of $o^{(t-1)}$ for training

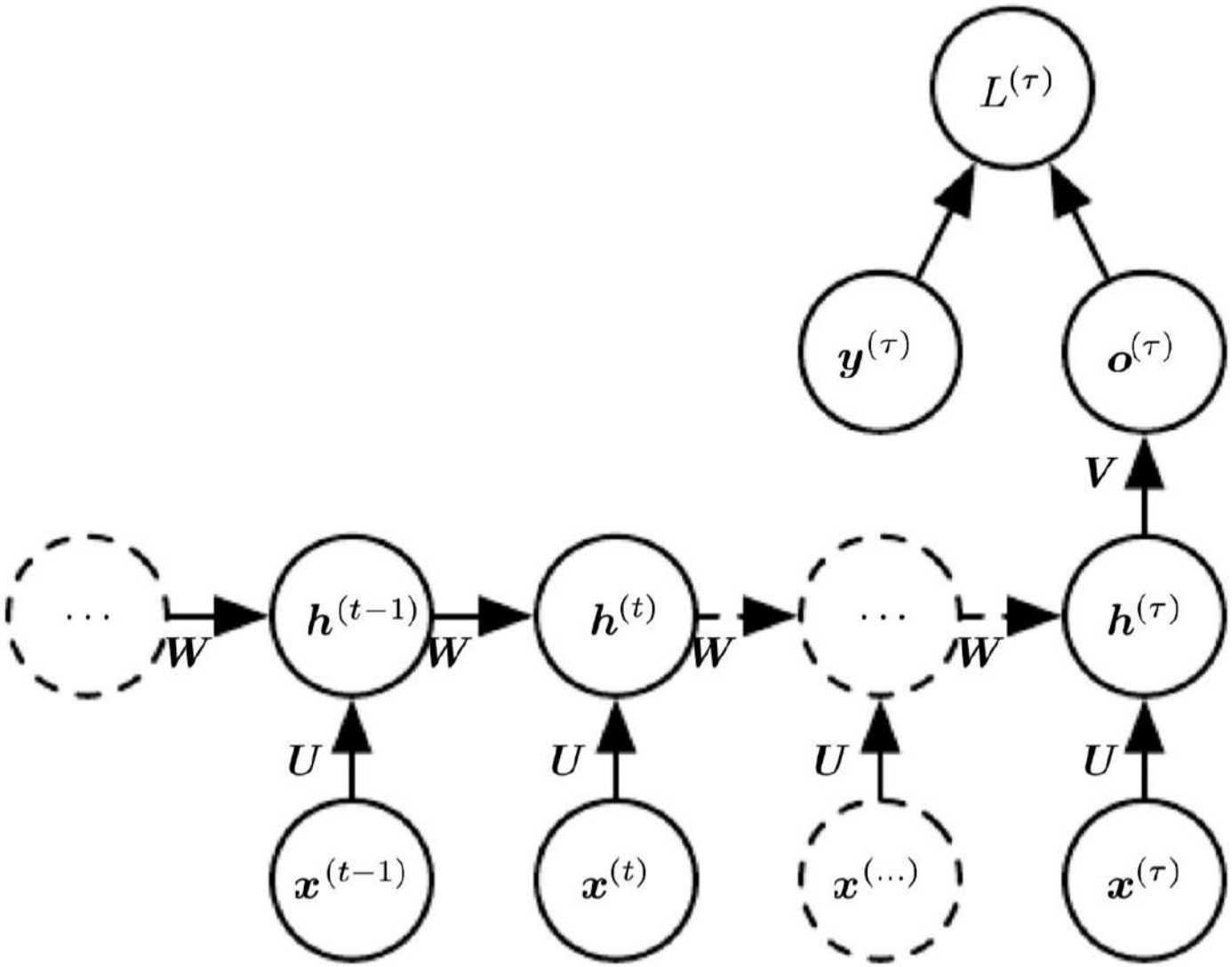
May not generalize well!

Randomly choose y^{t-1} or O^{t-1} at each iteration

Gradually converge to using O^{t-1}

Type 3

Output only at time τ



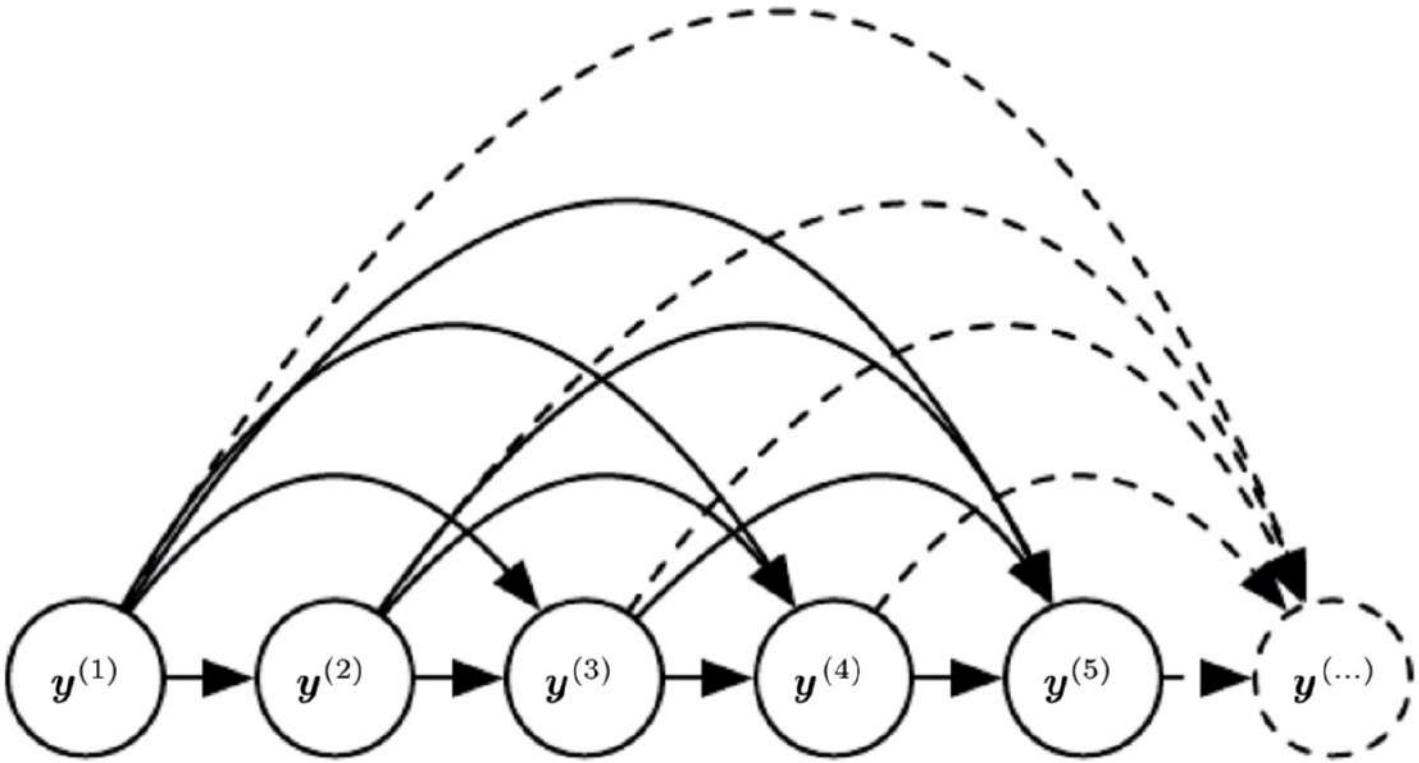
After unrolling, backpropagation works exactly as expected

Loss function

Maximize $\log P(y|x)$

RNN setting $P(y^t | x^t, x^{t-1}, \ldots, x^1)$

$P(y^t | x^t, x^{t-1} \ldots, x^1, y^{t-1}, \ldots, y^1)$



→ indicates influence

reverse indicates dependency

Representing conditional probabilities graphically

Naive Bayes model

Given an item $(a_1, a_2, \ldots, a_k)$

Compute $P(C = c \mid a_1, \ldots, a_k)$

Factorize as $P(c \mid a_1), P(c \mid a_2) \ldots$

Judea Pearl

California is prone to earthquakes

Set up a system where neighbours alert him to break-in at home

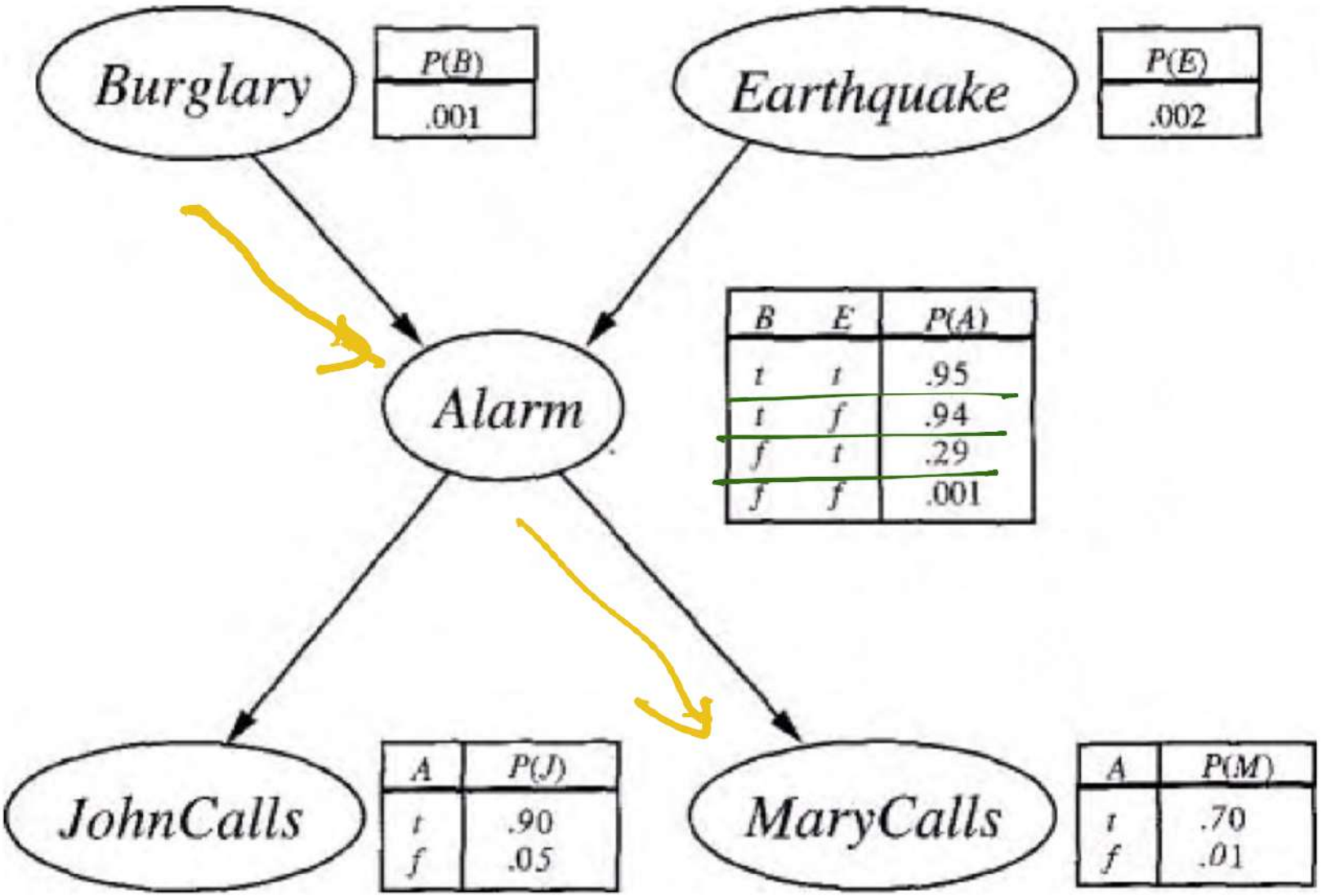
Two neighbours, Mary & John

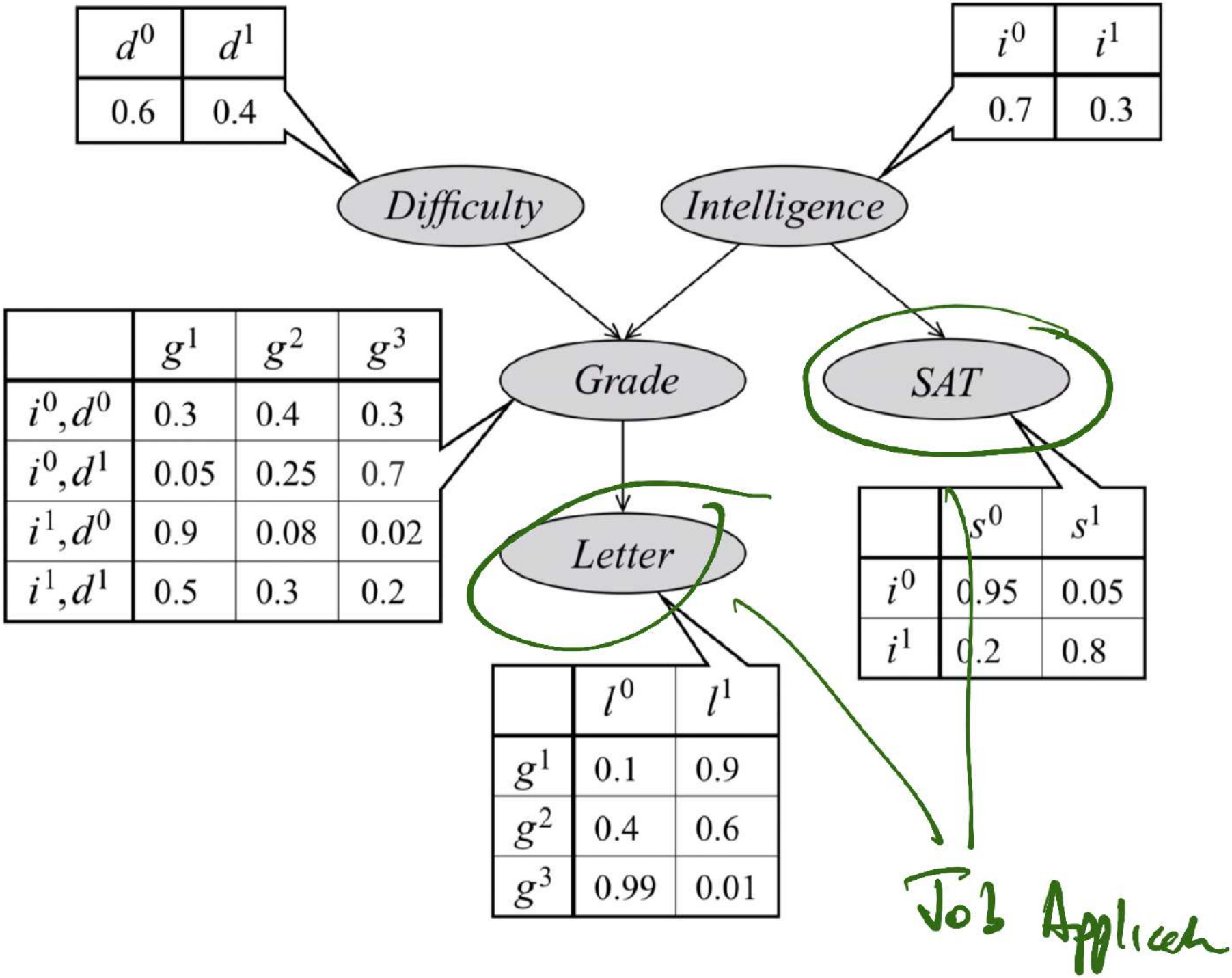
John is trigger happy - may mistake ambulance for alarm

Mary listens to loud music - may miss the alarm

Earthquakes may trigger alarm

John & Mary call & report an alarm.
What is the probability a burglary occurred?

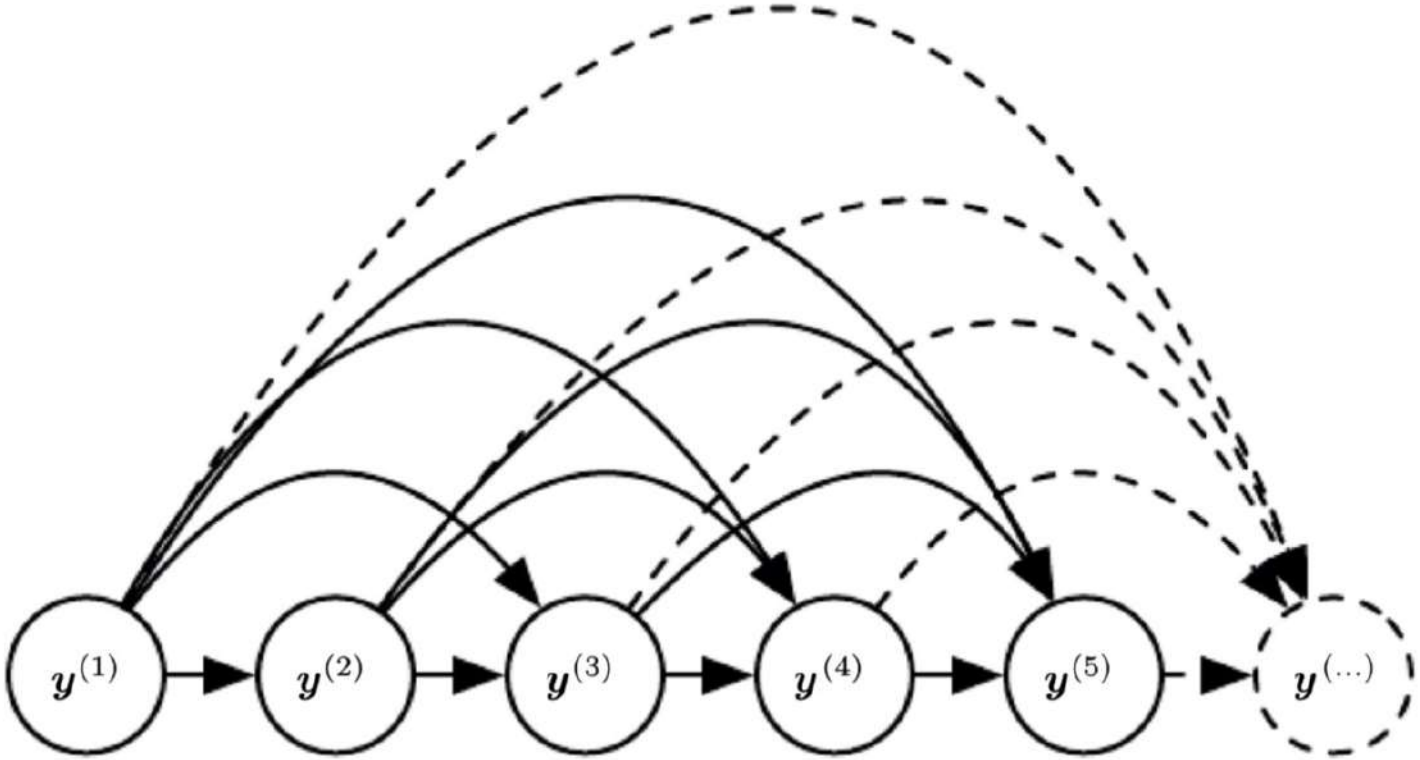




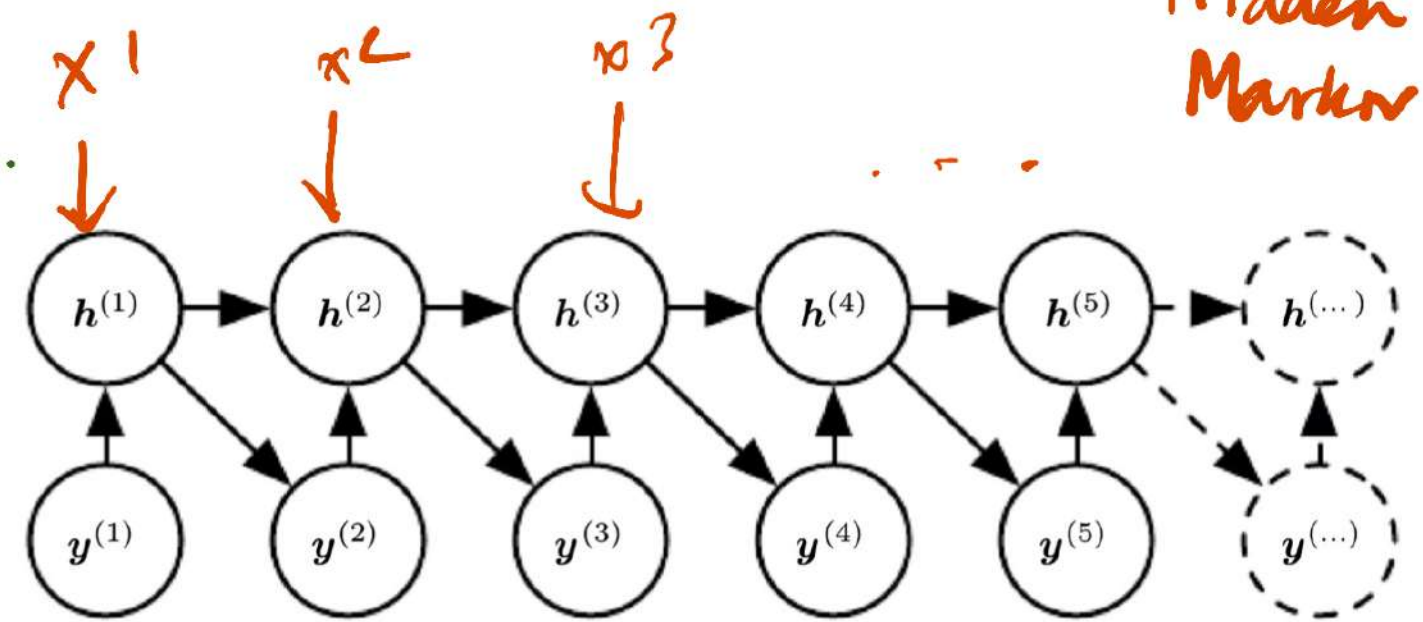
	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01



Hidden Markov Model



Markov

$$P(y^t | x^t x^{t-1} \ldots x^1) = P(y^t | x^t)$$